



**UNIVERZITET U NOVOM SADU  
TEHNIČKI FAKULTET  
„MIHAJLO PUPIN“  
ZRENJANIN**



---

**mr Sanja Maravić Čisar**

**EVALUACIJA ZNANJA  
PROGRAMSKIH JEZIKA C++ I JAVA  
PRIMENOM RAČUNARSKIH  
ADAPTIVNIH TESTOVA**

**DOKTORSKA DISERTACIJA**

**Mentor: dr Branko Markoski**

**Zrenjanin, 2011. godine**



**UNIVERZITET U NOVOM SADU  
TEHNIČKI FAKULTET  
„MIHAJLO PUPIN“  
ZRENJANIN**



---

**EVALUACIJA ZNANJA  
PROGRAMSKIH JEZIKA C++ I  
JAVA PRIMENOM RAČUNARSKIH  
ADAPTIVNIH TESTOVA**

---

**DOKTORSKA DISERTACIJA**

**Mentor**

**dr Branko Markoski**

**Kandidat**

**mr Sanja Maravić Čisar**

**Zrenjanin, 2011. godine**



## KLJUČNA DOKUMENTACIJSKA INFORMACIJA

Redni broj, <b>RBR</b> :	
Identifikacioni broj, <b>IBR</b> :	
Tip dokumentacije, <b>TD</b> :	Monografska dokumentacija
Tip zapisa, <b>TZ</b> :	Tekstualni štampani materijal
Vrsta rada, <b>VR</b> :	Doktorska disertacija
Autor, <b>AU</b> :	Mr Sanja Maravić Čisar
Mentor, <b>MN</b> :	Dr Branko Markoski, docent
Naslov rada, <b>NR</b> :	Evaluacija znanja programskih jezika C++ i Java primenom računarskih adaptivnih testova
Jezik publikacije, <b>JP</b> :	srpski (latinica)
Jezik izvoda, <b>JI</b> :	srpski/engleski
Zemlja publikovanja, <b>ZP</b> :	Srbija
Uže geografsko područje, <b>UGP</b> :	Vojvodina
Godina, <b>GO</b> :	2011.
Izdavač, <b>IZ</b> :	Tehnički fakultet „Mihajlo Pupin“ Zrenjanin
Mesto i adresa, <b>MA</b> :	23000 Zrenjanin, Đure Đakovića bb
Fizički opis rada, <b>FO</b> : (poglavlja/strana/citata/tabela/slika/priloga)	8 poglavlja /170 strana /150 citata /21 tabela /70 slika /2 priloga
Naučna oblast, <b>NO</b> :	Informatika u obrazovanju
Naučna disciplina, <b>ND</b> :	
Predmetna odrednica/Ključne reči, <b>PO</b> :	Računarski adaptivni testovi/ e-učenje/ testovi znanja/ ocenjivanje znanja/
<b>UDK</b>	
Čuva se, <b>ČU</b> :	Biblioteka Tehničkog fakulteta „Mihajlo Pupin“ u Zrenjaninu
Važna napomena, <b>VN</b> :	nema
Izvod, <b>IZ</b> :	Doktorska disertacija razmatra korišćenje računarskog adaptivnog testa u evaluaciji znanja programskih jezika C++ i Java. Za potrebe istraživanja realizovana je aplikacija koja omogućava adaptivno testiranje i njena funkcionalnost proverena je u praksi. Izvršeno je anketiranje studenata radi evaluacije, dobijeni podaci su statistički obrađeni i protumačeni. Na osnovu rezultata istraživanja utvrđeno je da postoji statistički značajna razlika u korist studenata eksperimentalne grupe, naspram studenata koji su test radili na konvencionalan način (papir i olovka).
Datum prihvatanja teme, <b>DP</b> :	17.03.2010.
Datum odbrane, <b>DO</b> :	
Članovi komisije, <b>KO</b> :	
Predsednik:	Dr Miodrag Ivković, redovni profesor, TF „Mihajlo Pupin“ Zrenjanin
Član:	Dr Dragan Ivetić, redovni profesor, FTN Novi Sad
Član:	Dr Dragica Radosav, vanredni profesor, TF „Mihajlo Pupin“ Zrenjanin
Član:	Dr Dragana Glušac, vanredni profesor, TF „Mihajlo Pupin“ Zrenjanin
Član, mentor:	Dr Branko Markoski, docent, TF „Mihajlo Pupin“ Zrenjanin



## KEY WORDS DOCUMENTATION

Accession number, <b>ANO</b> :	
Identification number, <b>INO</b> :	
Document type, <b>DT</b> :	Monographic publication
Type of record, <b>TR</b> :	Textual printed article
Contents code, <b>CC</b> :	Doctoral Thesis
Author, <b>AU</b> :	Sanja Maravić Čisar, M.Sc.
Mentor, <b>MN</b> :	Professor Branko Markoski, Ph.D.
Title, <b>TI</b> :	Computer Adaptive Tests in Evaluation of Knowledge of C++ and Java Programming Languages
Language of text, <b>LT</b> :	Serbian
Language of abstract, <b>LA</b> :	Serbian/English
Country of publication, <b>CP</b> :	Serbia
Locality of publication, <b>LP</b> :	Vojvodina
Publication year, <b>PY</b> :	2011.
Publisher, <b>PB</b> :	
Publication place, <b>PP</b> :	23000 Zrenjanin, Đure Đakovića bb
Physical description, <b>PD</b> : (chapters/pages/ref./tables/pictures/appendixes)	8 chapters /170 pages /150 ref. /21 tables /70 pictures /2 appendixes
Scientific field, <b>SF</b> :	
Scientific discipline, <b>SD</b> :	
Subject/Key words, <b>S/KW</b> :	computer adaptive tests/e-learning/knowledge tests/markings of knowledge/
<b>UC</b>	
Holding data, <b>HD</b> :	Library of the Faculty
Note, <b>N</b> :	none
Abstract, <b>AB</b> :	The doctoral dissertation discusses the application of the computer adaptive test in knowledge evaluation of the programming languages C++ and Java. For the purpose of this research an application was realized which enables adaptive testing. Its functionality was checked in practice. A survey was carried out among students for the purpose of evaluation, the attained data was statistically analyzed and interpreted. Based on the results of the research it was determined that there is a statistically significant difference in favor of the students in the experimental group as opposed to the students who had taken the test in the conventional way (paper-pencil tests).
Accepted by the Scientific Board on, <b>ASB</b> :	17.03.2010.
Defended on, <b>DE</b> :	
Defended Board, <b>DB</b> :	President: Miodrag Ivković, PhD, Full Time Professor , TF Zrenjanin Member: Dragan Ivetić, PhD, Full Time Professor, FTS, Novi Sad Member: Dragica Radosav, PhD, Associate Professor , TF Zrenjanin Member: Dragana Glušac, PhD, Associate Professor, TF Zrenjanin Member, Mentor: Branko Markoski, PhD, Assistant Professor, TF Zrenjanin

## Lista slika

Slika 2.1: Kognitivni domeni prema Blumu .....	23
Slika 4.1: Algoritam računarskog adaptivnog testa .....	46
Slika 4.2: Šema administracije Binetovog testa .....	48
Slika 4.3: Primer zapisa odgovora stradaptivnog testa .....	51
Slika 5.1: Verovatnoća tačnog odgovora $P(\theta)$ za dihotomnu stavku .....	63
Slika 5.2: Stavka funkcija za jednoparametarski (1PL) model .....	64
Slika 5.3: Određivanje težine stavke na osi sposobnost/težina.....	65
Slika 5.4: Funkcija odgovora stavke i funkcija informacije stavke za 1PL model .....	66
Slika 5.5: Funkcija odgovora stavke za pet stavki prema 1PL modelu .....	66
Slika 5.6: Funkcije stavskih odgovora za pet stavki prema 1PL modelu.....	67
Slika 5.7: Funkcija informacije stavke i funkcija informacije testa za pet stavki u 1PL modelu .....	68
Slika 5.8: Funkcija informacije testa i standardna greška merenja 1PL modela za pet stavki .....	69
Slika 5.9: Nalaženje procene sposobnosti .....	69
Slika 5.10: Pronalaženje procene sposobnosti po maksimalnoj verovatnoći .....	71
Slika 5.11: Funkcije verovatnoće za različite šablone odgovora sa ukupnim brojem bodova 1 .....	71
Slika 5.12: Funkcije odgovora stavke za 2PL model .....	72
Slika 5.13: Funkcije odgovora stavke i funkcije informacije stavke 2PL modela za 3 stavke .....	74
Slika 5.14: Funkcije informacija i SEM.....	75
Slika 5.15: Funkcije verovatnoće za tri funkcije odgovora koje imaju jednak posmatrani rezultat 1 .....	76
Slika 5.16: Funkcija odgovora stavke u 3PL modelu .....	77
Slika 5.17: Funkcija odgovora i funkcija informacije stavke 3PL modela .....	78
Slika 6.1: Administrativni panel Concerta .....	82
Slika 6.2: Tab R interakcija .....	82
Slika 6.3: Tab Users .....	82
Slika 6.4: Uvodni ekran adaptivnog testa napravljenog uz pomoć Concerta.....	83
Slika 6.5: HTML šablon za definisanje pitanja .....	83
Slika 6.6: HTML šablon koji prikazuje povratnu informaciju korisniku.....	84
Slika 6.7: Prikaz pitanja .....	86
Slika 6.8: Tab tables za pravljenje baze pitanja .....	87
Slika 6.9: Grafički prikaz standardne greške merenja, procenjene sposobnosti i težine pitanja.....	87
Slika 6.10: Glavni panel TAO-a .....	89
Slika 6.11: Modul Item u kome se definišu pitanja .....	90
Slika 6.12: Izgled modula Tests u kome se definišu pitanja koja čine test .....	90
Slika 6.13: Modul za definisanje korisničkih naloga.....	91
Slika 6.14: Modul Groups .....	91

Slika 6.15: Definisane periode od kad do kad je test aktivan.....	92
Slika 6.16: Definisane načina prikaza rezultata testa .....	92
Slika 6.17: Navigacioni panel .....	93
Slika 6.18: Baza pitanja sa podkategorijama.....	93
Slika 6.19: Editor pitanja .....	94
Slika 6.20: Tab Statistics .....	94
Slika 6.21: Grafički prikaz funkcije informacije pitanja za odabrani IRT model .....	95
Slika 6.22: Pretraga pitanja .....	95
Slika 6.23: Rezultat pretrage za zadati kriterijum.....	96
Slika 6.24: Definisane opsega za $\vartheta$ .....	96
Slika 6.25: Grafički prikaz funkcije informacije testa za zadati opseg vrednosti $\vartheta$ .....	97
Slika 6.26: Statistika pitanja prema parametrima $a$ , $b$ i $c$ .....	97
Slika 6.27: Adaptivna verzija GRE testa napravljena u MATLABu.....	99
Slika 6.28: Arhitektura sistema .....	100
Slika 6.29: Dijagram toka adaptivnog testa.....	102
Slika 6.30: Izgled početnog ekrana aplikacije za unos osnovnih podataka o studentu.....	104
Slika 6.31: Prikaz pitanja .....	105
Slika 6.32: Prikaz rezultata.....	110
Slika 6.33: Izgled testa .....	114
Slika 6.34: Struktura modela sistema .....	115
Slika 6.35: Primer pitanja .....	116
Slika 6.36: Izveštaj o uspehu na testu .....	117
Slika 6.37: Algoritam adaptivnog testa.....	119
Slika 6.38: Grafički korisnički interfejs računarskog adaptivnog testa .....	119
Slika 6.39: Prikaz rezultata testa.....	120
Slika 6.40: Prikaz uspeha na testu prema Blumovoj taksonomiji.....	120
Slika 6.41: Primer za „Preporučene tačke za reviziju”.....	121
Slika 7.1: Pregled završenih srednjih škola učesnika istraživanja .....	125
Slika 7.2: Nedeljni broj časova informatike u srednjoj školi .....	125
Slika 7.3: Odgovor studenata na pitanje koliko dugo poseduju računar .....	126
Slika 7.4: Stav studenata prema korišćenju računara.....	127
Slika 7.5: Rezultati mišljenja studenata eksperimentalne grupe o adaptivnom testiranju.....	137
Slika 7.6: Odgovori studenata na pitanje da procene kako su uradili test.....	138
Slika 7.7: Odgovori studenata na tvrdnje T1-T9.....	140
Slika 7.8: Odgovor studenata na pitanje da li bi adaptivni test preporučili kolegama .....	140

## Lista tabela

<i>Tabela 2.1: Podela testova prema vremenu i načinu korišćenja</i> .....	25
<i>Tabela 2.2: Podela testova prema načinu konstruisanja</i> .....	27
<i>Tabela 2.3: Podela pismenih vrsta pitanja</i> .....	27
<i>Tabela 6.1: Težina pitanja prema kognitivnim sposobnostima</i> .....	101
<i>Tabela 6.2: Broj pitanja prema kategorijama</i> .....	115
<i>Tabela 6.3: Kalibracija pitanja</i> .....	118
<i>Tabela 6.4: Poređenje aplikacija</i> .....	121
<i>Tabela 7.1: Broj studenata koji je učestvovao u istraživanju</i> .....	124
<i>Tabela 7.2: Navike u korišćenju računara</i> .....	126
<i>Tabela 7.3: Statistička analiza I kolokvijuma OOP 2009/2010</i> .....	129
<i>Tabela 7.4: Statistička analiza II kolokvijuma OOP 2009/2010</i> .....	130
<i>Tabela 7.5: Statistička analiza I kolokvijuma Java 2009/2010</i> .....	130
<i>Tabela 7.6: Statistička analiza II kolokvijuma Java 2009/2010</i> .....	131
<i>Tabela 7.7: Statistička analiza I kolokvijuma OOP 2010/2011</i> .....	131
<i>Tabela 7.8: Statistička analiza II kolokvijuma OOP 2010/2011</i> .....	132
<i>Tabela 7.9: Statistička analiza I kolokvijuma Java 2010/2011</i> .....	132
<i>Tabela 7.10: Statistička analiza II kolokvijuma Java 2010/2011</i> .....	133
<i>Tabela 7.11: Statistička analiza I i II kolokvijuma 2009/2010 i 2010/2011</i> .....	134
<i>Tabela 7.12: Značajnost razlika kontrolne i eksperimentalne grupe</i> .....	134
<i>Tabela 7.13: Trajanje testa kontrolne i eksperimentalne grupe</i> .....	135
<i>Tabela 7.14: Odgovori studenata na upitnik prema Likertovoj skali</i> .....	139

## Sadržaj

Lista slika.....	3
Lista tabela.....	5
Sadržaj.....	6
Korištene oznake.....	9
<b>1. Uvod.....</b>	<b>10</b>
1.1.  Obrazloženje o potrebama istraživanja.....	11
1.2.  Problem istraživanja.....	13
1.3.  Predmet i cilj istraživanja.....	15
1.4.  Zadaci istraživanja.....	16
1.5.  Hipoteze istraživanja.....	16
1.6.  Program istraživanja.....	16
1.7.  Metodološki koncept.....	17
1.7.1.  Način izbora, veličina i konstrukcija uzorka.....	17
1.7.2.  Tehnike, postupci i merni instrumenti istraživanja.....	18
1.7.3.  Mesto eksperimentalnog istraživanja.....	18
1.8.  Struktura disertacije.....	18
<b>2. Testovi znanja.....</b>	<b>20</b>
2.1.  Blumova taksonomija.....	22
2.2.  Pisana provera znanja.....	24
2.2.1.  Vrste testova znanja.....	25
2.2.2.  Životni ciklus pitanja.....	30
2.2.3.  Pravila pisanja ispitnih pitanja s više ponuđenih odgovora.....	31
2.2.4.  Analiza nakon sprovedenog testa.....	32
<b>3. Pregled literature.....</b>	<b>35</b>
3.1.  Prednosti i nedostaci računarskih testova.....	36
3.2.  Računarski adaptivni testovi.....	37
3.2.1.  Računarski adaptivni testovi realizovani u praksi.....	41
<b>4. Teorijske osnove računarskih adaptivnih testova.....</b>	<b>45</b>
4.1.  Binetov IQ test.....	47



4.1.1.	Šema administracije Binetovog testa.....	48
4.1.2.	Računarski administriran test – efikasnija varijanta Binetovog testa.....	50
4.2.	Osnovne komponente računarskih adaptivnih testova.....	52
4.2.1.	Baza pitanja.....	52
4.2.2.	Procedura selekcije pitanja.....	53
4.2.3.	Procena sposobnosti.....	54
4.2.4.	Pravilo zaustavljanja.....	56
4.2.5.	Kontrola izloženosti pitanja.....	57
4.2.6.	Projektovanje baze pitanja.....	58
<b>5.</b>	<b>Teorija odgovora na stavke.....</b>	<b>59</b>
5.1.	Klasična teorija testova.....	59
5.1.1.	Pouzdanost.....	60
5.2.	Teorija odgovora na stavke.....	62
5.3.	Jednparametarski model 1PL.....	63
5.3.1.	Funkcija informacije stavke za jednparametarski model.....	65
5.3.2.	Funkcija odgovora testa 1PL modela.....	66
5.3.3.	Funkcija informacije testa 1PL modela.....	67
5.3.4.	Preciznost i greška merenja.....	68
5.3.5.	Procena sposobnosti u 1PL modelu.....	69
5.3.6.	Funkcija verovatnoće.....	70
5.3.7.	Maksimalna verovatnoća procene sposobnosti.....	70
5.4.	Dvoparametarski model 2PL.....	72
5.4.1.	Funkcija odgovora stavke u 2PL modelu.....	72
5.4.2.	Funkcija odgovora testa 2PL modela.....	73
5.4.3.	Funkcija informacije stavke 2PL modela.....	73
5.4.4.	Funkcija informacije testa 2PL modela.....	74
5.4.5.	Standardna greška merenja 2PL modela.....	75
5.4.6.	Procena sposobnosti u 2PL modelu.....	75
5.5.	Troparametarski model 3PL.....	77
5.5.1.	Funkcija odgovora stavke 3PL modela.....	77
5.5.2.	Funkcija informacije stavke 3PL modela.....	78
5.5.3.	Funkcija odgovora testa 3PL modela.....	78
5.5.4.	Funkcija informacije testa 3PL modela.....	78
5.5.5.	Standardna greška merenja 3PL modela.....	79
5.5.6.	Procena sposobnosti u 3PL modelu.....	79
5.5.7.	Pogađanje i 3PL model.....	79
<b>6.</b>	<b>Dizajniranje računarskog adaptivnog testa.....</b>	<b>81</b>
6.1.	Primeri aplikacija za adaptivno testiranje.....	81
6.1.1.	Concerto.....	81

6.1.2.	TAO.....	89
6.1.3.	FastTEST .....	92
6.2.	Odabrano rešenje.....	98
6.3.	Kalibracija pitanja.....	100
6.4.	Početak testa.....	102
6.5.	Izbor sledećeg pitanja za administraciju .....	102
6.6.	Zaustavljanje testa .....	103
6.7.	Pojašnjenje programskog koda .....	104
6.8.	Primer testa.....	111
6.9.	Poređenje aplikacije sa sličnim rešenjima .....	114
<b>7.</b>	<b>Istraživanje.....</b>	<b>123</b>
7.1.	Rezultati eksperimenta .....	123
7.1.1.	Učesnici eksperimenta .....	123
7.1.2.	Pitanje sigurnosti procesa testiranja.....	127
7.2.	Postupak ispitivanja i metodologija .....	128
7.3.	Analiza rezultata.....	128
7.3.1.	Diskusija rezultata .....	133
<b>8.</b>	<b>Zaključak.....</b>	<b>142</b>
<b>Prilog.....</b>		<b>145</b>
A.	Anketa .....	145
B.	Programski kod.....	148
<b>Literatura .....</b>		<b>158</b>

## Korištene oznake

Oznaka	Značenje
1-PL	Jednparametarski logički model
2-PL	Dvoparametarski logički model
3-PL	Troparametarski logički model
CAA	Računarski podržana procena znanja
CAT	<i>Computer Adaptive Testing</i> , računarsko adaptivno testiranje, računarski podržano adaptivno testiranje
CBA	<i>Computer-Based Assessment</i> , računarski podržana procena znanja
CBT	<i>Computer-Based Testing</i> , računarski podržano testiranje
IRT	<i>Item Response Theory</i> , teorija odgovora na stavke
IT	<i>Information technology</i> , informaciona tehnologija
SEM	<i>Standard error of measurement</i> , standardna greška merenja
TOS	Teorija odgovora na stavke

## 1. Uvod

Moderne tehnologije nude brojne mogućnosti za unapređenje procesa obrazovanja i procenjivanja znanja. Istorija upotrebe računara za obavljanje procesa provere znanja počinje 70-tih godina dvadesetog veka. Međutim, tadašnja visoka cena računara i njihove tehničke mogućnosti ograničili su njihovu primenu za testiranje. Napredak tehnologije omogućio je razvoj i primenu računara za testiranje u mnogim oblastima, pa tako i u procesu obrazovanja.

Testiranje je jedan od najčešće primenjivanih načina provere znanja u visokom obrazovanju. Osnovni cilj testiranja je da se odredi nivo znanja studenata iz jedne ili više oblasti predmeta iz kojeg se znanje proverava. Primenuju se različite metode procene znanja u različitim kontekstima, kao na primer prezentacija na času, pisanje eseja, projekti, itd. Međutim, najčešći „alat“ koji se koristi za proveru znanja su usmeni ispit i test. S obzirom da se računar kao nastavno sredstvo poslednjih nekoliko decenija sve više koristi, a njegova upotreba se širi na svim nivoima obrazovanja, tako se sve više mogu sresti i testovi koji se rade na računaru (engl. *computer based tests* – CBT).

Pomak ka testovima koji se rade na računaru kao alternativa klasičnom testu primetan je u velikom broju različitih oblasti. Testiranje primenom računara omogućava fleksibilnost kako u pogledu vremena tako i mesta na kome ispitanici rade test, a olakšava i ubrzava pregledanje testova u slučaju velikog broja ispitanika. Takođe, računar u formi računarskog adaptivnog testa (engl. *computerized adaptive test* – CAT) omogućava proces administracije testa na način koji se razlikuje od tradicionalnog linearnog testa kada svi ispitanici dobijaju isti set pitanja.

Najčešći oblik računarskih testova su linearni testovi fiksne dužine. Test je isti za sve ispitanike. Pitanja se nasumice biraju iz baze pitanja prema unapred definisanom pravilu koliko treba da bude lakših a koliko težih pitanja, a rezultat zavisi od broja pitanja na koje je ispitanik tačno odgovorio. Ova vrsta testova koja se radi na računaru zapravo „imitira“ tradicionalne testove koji se rade na papiru i predstavlja njihovu digitalnu verziju, pri čemu se vrlo malo vodi računa o sposobnostima svakog pojedinog ispitanika.

Tokom rešavanja klasičnog testa, ispitanici se mogu osećati obeshrabreno ako su pitanja suviše teška, ili, s druge strane, možda izgubiti interes, ako su pitanja suviše laka za njihov nivo znanja. Rešenje ovog problema može biti primena računarskih adaptivnih testova koji kao i kod kvalitetnog usmenog ocenjivanja imaju mogućnost da promene nivo težine pitanja prema mogućnostima ispitanika.

Računarski adaptivni test daje mogućnost pravljenja testova prema sposobnostima ispitanika. Pitanja se biraju iz baze pitanja uzimajući u obzir pojedinačne sposobnosti kandidata tokom testa, ili drugim rečima, kako je svaki ispitanik odgovorio na prethodno postavljena pitanja. Ako ispitanik odgovori tačno na postavljeno pitanje, sledeće pitanje bira se iz grupe pitanja koja su za jedan nivo veće težine od prethodno datog pitanja. S druge strane, ako ispitanik odgovori pogrešno, sledeće pitanje bira se iz grupe pitanja koja su za jedan nivo lakše težine od pitanja na koje je ispitanik pogrešno odgovorio. Na ovaj način ispitanicima sa slabijim znanjem biće data lakša pitanja, dok će oni koji su bolje pripremljeni za test dobiti set težih pitanja. Krajnji rezultat ispitanika na testu zavisi od nivoa težine pitanja na koja je tačno odgovorio, stoga ispitanici mogu imati isti procenat tačno datih odgovora, ali oni koji su odgovorili tačno na teža pitanja imaju bolji rezultat.

Na osnovu ovoga može se navesti nekoliko značajnih prednosti računarskih adaptivnih testova u odnosu na tradicionalne linearne testove, kao što su 1) individualizacija testova, 2) test kraće traje, 3) promovisanje pozitivnog odnosa prema testiranju, 4) rezultati testa su odmah poznati, 5) određivanje nivoa znanja sa većom preciznošću i 6) povećanje sigurnosti testiranja.

Pošto se adaptivni test formira prema nivou znanja studenata, to svakako omogućava nastavnicima da bolje procene sposobnost svakog pojedinca. Računar omogućava pravljenje individualnih izveštaja, ne samo u smislu postignutog broja poena nego i slabih i jakih strana svakog pojedinačnog studenta. Istraživanja koja su rađena o uticaju kognitivnih stilova učenja na rezultat postignut na računarskom adaptivnom testu [10], [77] pokazuju da računarski adaptivni testovi pružaju jednake mogućnosti svim ispitanicima bez obzira na kognitivni stil učenja koji im odgovara, da iskažu svoje znanje na način koji je manje stresan za njih.

## **1.1. Obrazloženje o potrebama istraživanja**

Programiranje je sastavni deo edukacije u obrazovanju budućih inženjera informatičkog usmerenja. To je jedna od osnovnih veština koju mora da savlada svako ko je zainteresovan da studira ovu oblast. U učenju programiranja, studenti se prvo upoznaju sa osnovnim konceptima programiranja i strukturama podataka, uče kako da analiziraju problem, primene specifične tehnike za predstavljanje rešenja i testiraju rešenje. Zatim, imaju zadatak da rešenje problema napišu u kodu korišćenjem određenog programskog jezika. Održavanje softverskog rešenja je poslednji korak, ujedno i najskuplja i najduža faza u životnom ciklusu softvera, i zasniva se na zahtevima korisnika.

Programiranje je široka naučna oblast koja zahteva od studenata dobro poznavanje sintakse programskog jezika koji uče, kako bi svoje znanje mogli da primene u praksi. Poznavanje sintakse nije dovoljno za rešenje problema, nego je potrebno posedovati i logiku koja stoji iza ideje kako da se problem reši. Da bi se napisao jednostavan program, student mora da ima bar osnovna znanja o varijablama, ulazu/izlazu

podataka, kontrolnim strukturama itd. Mora da zna da napravi algoritam koji može da rešiti zadati problem. Za studente još veći problem predstavlja ovladavanje složenijim konceptima kao što su pointeri, apstrakcija ili obrada izuzetaka. I kada studenti poseduju neophodna teorijska znanja, problem nastaje kada je potrebno to znanje primeniti za rešavanje konkretnog problema pisanjem programskog koda.

Za nastavnika veliki problem u nastavi predstavlja činjenica da studenti jedne grupe nisu homogeni po prethodnom znanju. Ne mali broj studenata upisuje se na informatički smer nakon završene srednje ekonomske, medicinske ili građevinske škole i prvi put se sreće sa nekim programskim jezikom na I godini studija. Drugu grupu studenata čine oni koji su već u srednjoj školi savladali najmanje jedan programski jezik i imali nekoliko časova nedeljno nastavu programiranja tokom celog školovanja. U takvoj situaciji vrlo je teško osmisliti nastavu koja bi zadovoljila dijametralno različite potrebe studenata. Odabir zadataka za ispit ili kolokvijum, traži od nastavnika vrlo pažljivu selekciju pitanja kako bi se i početnicima u učenju programiranja omogućilo da ispune kriterijume neophodne za polaganje ispita, a da istovremeno ispit ne bude isuviše lak za one studente koji spadaju u napredniju grupu.

Kada je grupa sa kojom se radi velika, što je uobičajeno na prvoj godini studija, vrlo je teško pratiti napredak svakog studenta pojedinačno. Nastavnik nema dovoljno vremena tokom predavanja i vežbi da se posveti svakom studentu pojedinačno, te da prati njegov napredak. Pregledanje kolokvijuma i ispita, koji se rade na tradicionalan način može da traje dugo (više sati, pa čak i dana u zavisnosti od broja studenata i nastavnika koji pregledaju radove), što svakako dovodi do nezadovoljstva studenata zbog brzine kojom se sve odvija. Pregledanje programskog koda vrlo je zamoran i zahtevan posao, pažnja i koncentracija nastavnika opada, te se vrlo lako može desiti da se zbog previda, nehotice ošteti neki student davanjem neadekvatne ocene. Problem s kojim se nastavnici susreću u organizaciji kolokvijuma ili ispita, kada je polaganje organizovano na tradicionalan način da se zadaci daju na papiru, jeste činjenica da se ne zna tačan broj primeraka ispitnih klišea koje je potrebno odštampati. Kako na ispit ne izađu svi studenti koji su ga i prijavili, a nastavnik mora da odštampa onoliko klišea koliko je studenata i prijavilo ispit, neracionalno se troše velike količine papira za štampanje. Kod nastavnika se gomilaju papiri sa ispitnim zadacima koji su bili „višak“, kao i hrpe pregledanih zadataka studenata koji su polagali ispit. Ovo svakako predstavlja neekonomično ponašanje a neracionalno trošenje velike količine papira nije u skladu sa ekološkim principima.

Sigurnost testa je pitanje koje se javlja prilikom svakog ispita. Ukoliko je sigurnost testa narušena, dovodi se u pitanje i validnost rezultata koje su ispitanici ostvarili. Na svakom ispitu postoje pojedinci koji na nedozvoljen način i primenom zabranjenih pomagala (prepisivanje od kolega, iz knjiga, bubice itd.) pokušavaju da ostvare bolji rezultat nego što je njihovo realno znanje. Kada se test polaže na računaru, najčešći je slučaj da studenti pristupaju polaganju testa asinhrono (ukoliko nema dovoljno računara da svi mogu odjednom da polažu). Ukoliko test na računaru predstavlja samo digitalnu verziju papirnog testa, jedan deo studenata može se naći u povoljnijem

položaju i od kolega koji su već polagali test saznati pitanja. Time se narušava objektivnost testa. Kod računarskog adaptivnog testa sigurnost je povećana jer se pitanja dinamički prilagođavaju znanju svakog pojedinog ispitanika. Što je baza pitanja veća, smanjuje se i mogućnost prevelike izloženosti pitanja (ali se ne eliminiše u potpunosti).

Nastava i učenje programiranja stvara znatne teškoće i nastavnicima i studentima [150]. Da bi se te teškoće prevazišle, potrebni su inovativni načini da se poboljšala efikasnost nastave programiranja. Primenom računara i adaptivnih testova za proveru znanja studenata iz programiranja (ili neke druge oblasti) može se postići značajan napredak [137].

Na Visokoj tehničkoj školi strukovnih studija u Subotici, na prvoj godini studija na smeru Informatika, uči se programski jezik C++ u okviru predmeta Objektivno orijentisano programiranje. Jezik C++ je objektivno orijentisan programski jezik koji je vrlo moćan, fleksibilan ali i kompleksan programski jezik. Nastao je kao naslednik jezika C, pa je zadržao i koncepte tradicionalnog programiranja. Stoga je pogodan i za početnike da se upoznaju s osnovnim konceptima programiranja, kao i za sistemsko i aplikativno programiranje gde se primenjuje objektivno orijentisani koncept.

Nastavnim planom i programom, predviđen je fond od 2 časa nedeljno predavanja, 2 časa laboratorijskih i 1 čas auditornih vežbi. Uspešno položen ispit vredi 7 ECTS bodova (engl. *European Credit Transfer System*, Evropski sistem prenosa bodova). Predmet je obavezan za sve studente I godine Informatike i izborni predmet za studente II godine Elektro smeru.

Na drugoj godini studija, smer Informatika, izučava se predmet Java, kao logični nastavak predmeta prve godine Objektivno orijentisano programiranje. Uspešno polaganje predmeta Objektivno orijentisano programiranje predstavlja uslov da student može da sluša predmet Java. Cilj predmeta je da se studenti osposobe za samostalno pisanje jednostavnih konzolnih aplikacija i Java apleta. Plan predviđa fond nastave od 2 časa predavanja i 2 časa laboratorijskih vežbi. Uspešno polaganje ovog predmeta studentu obezbeđuje 6 ECTS kredita.

Objektivno orijentisano programiranje i Java, su predmeti koji su odabrani za realizaciju računarskog adaptivnog testa sa namerom da se nastava inovira i poboljša efikasnost nastave programiranja i poveća motivisanost i zadovoljstvo studenata koji slušaju ove predmete.

## **1.2. Problem istraživanja**

Iskustvo u nastavi programiranja govori da polaganje predmeta Objektivno orijentisano programiranje za studente predstavlja značajan problem. Predispitne obaveze predviđaju polaganje dva kolokvijuma, čiji pozitivan ishod omogućava izlazak na ispit.

Statistički podaci studentske službe Visoke tehničke škole strukovnih studija u Subotici za period od 2004. do 2009. godine, pokazuju da je prolaznost na kolokvijumima tek nešto veća od 20%, a samim tim mali broj studenata stiže pravo i da izađe na ispit. Na predmetu Java, situacija je nešto bolja i prolaznost na kolokvijumima je oko 55%. Tokom rešavanja konvencionalnog testa, ispitanici se mogu osećati obeshrabreno ako su pitanja suviše teška, ili, s druge strane, možda izgube interes, ako su pitanja suviše laka za njihov nivo znanja. Kako je programiranje osnovna veština koju moraju da savladaju budući strukovni inženjeri elektrotehnike i računarstva, negativan uticaj osnovnih uvodnih kurseva može imati štetne posledice u daljnjem obrazovanju studenata u ovoj oblasti.

Da bi se prevazišao problem koji je uočen u nastavi programiranja na I i II godini studija, te da bi se polaganje ispita prilagodilo svakom pojedinačnom studentu i njegovim individualnim sposobnostima, da bi se povećala brzina pregledanja zadataka, pristupačnost studentima, smanjio stres tokom polaganja ispita zbog pitanja koja su neadekvatna sposobnostima studenta, kao i povećala objektivnost ocenjivanja primenjen je računarski adaptivni test. Rezultati dobijeni na testu validirani su i kroz usmeni deo ispita.

Za realizaciju istraživanja dizajniran je adaptivni test u programskom paketu MATLAB verzija 7.10 (R2010a, dostupan od 5.marta 2010. godine). Iako osnovna namena MATLABa nije pravljenje konzolnih aplikacija, odabran je za realizaciju računarskog adaptivnog testa jer ga studenti koriste tokom nastave iz predmeta Osnovi računarstva, koji je obavezan za sve studente u prvom semestru. Autor je smatrao da će radno okruženje koje je poznato svim studentima doprineti pozitivnom stavu i smanjenju napetosti tokom testa. U prvom semestra studenti imaju i predmet Računarska laboratorija I, koji ima za cilj osposobljavanje studenata za ovladavanje osnovnim veštinama rada na računaru, te je time smatrano da je ispunjen uslov da svi studenti poseduju potrebne veštine da na adekvatan način koriste računarski adaptivni test.

Prema rezultatima istraživanja [28], način na koji će se vršiti procena znanja treba da bude pažljivo dizajniran tako da bude u skladu sa pedagoškom teorijom. U radu [80] autori preporučuju nastavnicima da se procena znanja vrši prema kognitivnim nivoima koji su definisani u Blumovoj taksonomiji obrazovnih ciljeva [22]. Prema ovoj taksonomiji postoje tri domena obrazovanja, odnosno načina usvajanja određenog sadržaja: kognitivni domen – usvajanje znanja, afektivni domen – stavovi, vrednosti i interesovanja i psihomotorni domen – veštine. Svaki od domena ima određene nivoe usvajanja informacija i oni su od najnižeg ka najvišem sledeći: znanje, razumevanje, primena, analiza, sinteza i evaluacija. Na prvoj godini studija treba obratiti pažnju na prva tri kognitivna nivoa (znanje, razumevanje i primena), kao što se sugerije u radu [33]. Nakon prve godine, student bi trebalo da bude sposoban da pređe na više kognitivne nivoe (analiza, sinteza i primena). U skladu sa ovom preporukom, pri sastavljanju i odabiru pitanja za istraživanje koje se radilo, vodilo se računa da se



odaberu pitanja koja mogu da daju procenu prve tri kognitivne sposobnosti Blumove taksonomije, što je detaljnije opisano u poglavlju 6.

U cilju procene studentovog znanja, mogu se koristiti testovi koje čine pitanja višestrukog izbora ili rešavanje problema koje zahteva pisanje koda u određenom programskom jeziku. Prema rezultatima istraživanja datih u radu [159], kada su pitanju početnici u programiranju bolje je primeniti pitanja višestrukog izbora, a ako su studenti na naprednijem nivou znanja tada su bolji zadaci koji zahtevaju pisanje koda. Naravno, moguće je primeniti i kombinaciju ova dva načina procene znanja. Pitanja sa višestrukim izborom mogu da daju efikasnu procenu znanja početnika u učenju programiranja u kojoj meri su savladali osnovne veštine [79]. Ukoliko na ovoj proveri znanje osnovnih veština imaju loš rezultat ili prosečan, vrlo je verovatno da neće uspeti da polože finalni ispit koji zahteva više znanja. Naravno, pitanja višestrukog izbora, ukoliko su kvalitetna i dobro dizajnirana, daju mogućnost da se ispitaju i složenije [121]. Složenost testa koje čina pitanja višestrukog izbora može se povećati povećanjem broja ponuđenih odgovora. Računarski adaptivni test s pitanjima višestrukog izbora može se koristiti i za rangiranje studenata prema nivo znanja (početni, srednji, napredni).

Imajući u vidu prethodno rečeno, test koji se koristio u istraživanju sadržao je pitanja višestrukog izbora s pet ponuđenih odgovora. Način izbora pitanja opisan je u poglavlju 6.

Procena sposobnosti ispitanika koju daje računarski adaptivni test napravljen za potrebe testiranja upoređena je sa aplikacijom SIETTE [30], [31], za koju je autor dobio pravo pristupa zahvaljujući ljubaznosti gospodina *Ricarda Conejo Muñoza*. Grupa od 30 ispitanika polagala je adaptivni test primenom obe aplikacije za testiranje i izvršeno je njihovo rangiranje prema nivou znanja u početnu, srednju i naprednu grupu. Baza pitanja koja se koristila bila je ista za obe aplikacije. Validacija rezultata dobijena primenom ova dva računarska adaptivna testa izvršena je davanjem dodatnih zadataka koji su se odnosili na pisanje koda i kroz usmeni razgovor sa ispitanicima. Pokazalo se da adaptivni test realizovan u MATLABu bolje i preciznije vrši rangiranje ispitanike prema nivoima znanja.

### **1.3. Predmet i cilj istraživanja**

Predmet istraživanja disertacije je mogućnost primene računarskih adaptivnih testova u proceni znanja studenata programskih jezika Java i C++, da li se i kakva unapređenja postižu u nastavi, kao i stavovi samih studenata prema ovakvom načinu provere znanja. Primenom računarskog adaptivnog testa, koji ima mogućnost dinamičke promene težine pitanja prema mogućnostima ispitanika, ispitanicima se daju pitanja prilagođena njihovim sposobnostima, a postignuti bodovi zavise od težine pitanja na koje je student tačno odgovorio. Ispitanici sa istim postotkom tačnih odgovora mogu se znatno razlikovati u krajnjem rezultatu u zavisnosti od težine pitanja na koja su tačno odgovorili. Primenom računarskog adaptivnog testa za proveru znanja želi se

postići da ocena bude prava mera znanja studenata uz povećanje zadovoljstva i motivisanosti ispitanika.

Osnovni cilj doktorske disertacije je da se na osnovu teorijskih istraživanja i korišćenja mogućnosti računara u evaluaciji znanja ukaže na statistički značajnu mogućnost podizanja sveobuhvatnog nivoa i kvaliteta nastavnog procesa uvođenjem računarskih adaptivnih testova. Osnovna postavka istraživanja je da se proverom znanja na računaru primenom adaptivnih testova može preciznije proceniti znanje svakog pojedinca i postići veće zadovoljstvo i motivisanost ispitanika. Stoga je cilj naći odgovor na pitanje da li postoje razlike u postignutim rezultatima studenata na testu, a koji su posledica različitog modaliteta isporuke testa.

#### **1.4. Zadaci istraživanja**

Na osnovu predmeta i postavljenog cilja istraživanja, zadatak istraživanja bio je:

1. Izvršiti sistematizaciju znanja iz oblasti primene računara u proveru znanja, posebno primenom računarskih adaptivnih testova;
2. Klasifikovanje naučnih saznanja iz oblasti primene računara u testiranju;
3. Kreiranje modela računarskog adaptivnog testa;
4. Implementiranje modela računarskog adaptivnog testa;
5. Eksperimentalno proveravanje modela računarskog adaptivnog testa;
6. Ispravljavanje eventualno uočenih nedostataka;
7. Ponovna provera modela;
8. Evaluacija modela.

#### **1.5. Hipoteze istraživanja**

Glavna hipoteza disertacije glasi: *Ne postoji statistički značajna razlika u postignutim rezultatima studenata koji polažu test primenom računarski adaptivnog testa i koji polažu test na klasičan način (papir i olovka).*

Definisane su i pomoćne hipoteze:

H1: računarski adaptivni test skraćuje vreme potrebno za proveru usvojenosti gradiva kod studenata;

H2: računarski adaptivni test povećava motivisanost studenata za učenje.

#### **1.6. Program istraživanja**

Istraživanje za ovaj rad obavljeno je tokom akademske 2009/10. i 2010/11. godine. U istraživanju su učestvovali studenti I i II godine Visoke tehničke škole strukovnih studija u Subotici smera Informatika. U istraživanju je učestvovao i jedan broj studenata

Elektro smeru koji ima mogućnost da sluša ove predmete kao izborne. Istraživanje se odvijalo u sledećim fazama:

1. Projektovanje istraživačkog projekta:
  - a. izrada istraživačkog projekta,
  - b. usvajanje istraživačkog projekta,
  - c. generisanje pilot modela adaptivnog testa.
2. Prikupljanje podataka:
  - a. proučavanje relevantne literature,
  - b. analiza postojećih sistema za proveru znanja primenom računara,
  - c. prikupljanje podataka koji će omogućiti proveravanje postavljenih hipoteza.
  - d. korekcije pilot modela.
3. Sređivanje i obrada podataka:
  - a. analiza podataka, tj. statistička obrada podataka,
  - b. prezentovanje rezultata istraživanja,
  - c. rezime i najvažnijih rezultata istraživanja,
  - d. zaključna razmatranja.

Nakon obavljenog eksperimenta studenti su u pismenoj formi mogli da izraze svoje mišljenje o modelu, njihove primedbe i sugestije predstavljaju prvu reviziju modela na osnovu koje je potrebno izvršiti unapređenje postojećeg modela.

## **1.7. Metodološki koncept**

Istraživanje se bazira na pedagoškom eksperimentu i primeni statističkih tehnika obrade i analize empirijskih podataka. Metodološka osnova istraživanja zasniva se na kombinovanom metodološkom pristupu: deskriptivnom metodom dat je pregled relevantnih istraživanja u oblasti primene računara u evaluaciji znanja i računarskih adaptivnih testova, dok se u empirijskom delu rada u eksperimentalnoj metodi koristi tehnika paralelnih grupa. U delu interpretacije rezultata istraživanja primenjena je deskriptivna statistička analiza. Faze rada bile su prikupljanje, predstavljanje, analiziranje i interpretiranje numeričkih podataka. Eksperimentalnom metodom izvršena je implementacija softvera za procenu znanja studenata i proveren učinak u odnosu na tradicionalnu nastavu.

### **1.7.1. Način izbora, veličina i konstrukcija uzorka**

Populaciju istraživanja činili su studenti Visoke tehničke škole strukovnih studija u Subotici koji slušaju predmete Java i Objektno orijentisano programiranje (C++). Predviđeni uzorak pripada kategoriji namernih uzoraka. Ukupan broj studenata koji je učestvovao u istraživanju bio je 352.

### **1.7.2. Tehnike, postupci i merni instrumenti istraživanja**

U istraživanju je primenjena tehnika paralelnih grupa relativno ujednačenih po znanju. Na ovaj način moguće je uporediti rezultate dve grupe jedne sa konvencionalnim načinom provere znanja a druge u eksperimentalnom načinu radu (provera znanja računarskim adaptivnim testom). Grupe su formirane tako da budu ujednačene po broju, računarskoj pismenosti, starosti i polu. Od postupaka koristila se analiza dokumentacije, posmatranje, testiranje i anketiranje.

### **1.7.3. Mesto eksperimentalnog istraživanja**

Istraživanje je sprovedeno na Visokoj tehničkoj školi strukovnih studija u Subotici, akademske 2009/2010. i 2010/2011. godine.

## **1.8. Struktura disertacije**

U uvodnom poglavlju ukazano je na glavne probleme koji su prisutni u nastavi programiranja, opisan je predmet istraživanja i postavljeni cilj disertacije.

Drugo poglavlje daje pregled osnovne teorije vezane za testove, kao najobjektivnijeg sredstva za merenje znanja. Testovi znanja su testovi pomoću kojih se određuje koliko je znanja pojedinac stekao kroz određenu aktivnost ili određeni period učenja. Oni se sastoje iz zadataka (datih u specijalnim oblicima) pomoću kojih se ispituje znanje učenika iz pojedinih delova, ili celokupnog gradiva jednog predmeta. Data je podela testova prema načinu i vremenu zadavanja, kriterijumu u odnosu na koji se procenjuje znanje učenika i načinu konstruisanja. Kako su pitanja sa višestrukim odgovorima najčešća pitanja koja se zadaju na testovima znanja, date su smernice o kojima je potrebno voditi računa prilikom sastavljanja ove vrste pitanja.

Treće poglavlje daje pregled literature koja se bavi testiranjem primenom računara, s posebnim osvrtom na računarske adaptivne testove. Dat je pregled problema vezanih za računarske adaptivne testove koji su razmatrani u literaturi, kao i primeri nekoliko u praksi realizovanih adaptivnih testova.

Četvrto poglavlje čine teorijske osnove računarskih adaptivnih testova. Opisan je način administracije Binetovog IQ testa koji predstavlja prvi adaptivni test, kao i njegova varijacija stradaptivan test. Dalje, daje se pregled četiri glavne komponente koji svaki računarski adaptivni test mora da ima, a to su: baza pitanja, procedura za selekciju pitanja, metod za procenjivanje sposobnosti i pravilo zaustavljanja.

U petom poglavlju govori se o osnovnim razlikama između klasične teorije testova i modernih psihometrijskih koncepcija. Teorija odgovora na stavke, termin koji u novije vreme označava moderne koncepcije, proučava skorovanje testova i pitanja zasnovano na pretpostavkama o matematičkim relacijama sposobnosti ispitanika i odgovora datih

na pitanje. Opisani su osnovni elementi jednoparametarskog (1PL), dvoparametarskog (2PL) i troparametarskog modela (3PL).

Većina CAT sistema koji su realizovani u praksi ne nalaze se u slobodnoj upotrebi, tako da ukoliko neko želi da koristi neki od CAT sistema mora da izvrši plaćanje licence za korišćenje, ili da sam razvija svoj sistem za adaptivno testiranje. Postoji nekoliko platformi za adaptivno testiranje koje su otvorenog koda i kojima može da se pristupi. U poglavlju šest dat je kratak prikaz tri aplikacije Concerto, TAO i FastTEST koje omogućavaju adaptivno testiranje, a koje su otvorenog koda. Za potrebe utvrđivanja efekata primene računarskog adaptivnog testa za proveru znanja realizovan je adaptivni test u programskom paketu MATLAB, što je takođe prikazano u šestom poglavlju disertacije. Opisan je način na koji je izvršena kalibracija pitanja, algoritam izbora sledećeg pitanja za administraciju i zaustavljanja testa. Aplikacija je upoređena sa dve nekomercijalne aplikacije koje se koriste za adaptivno testiranje na visokoškolskim ustanovama u Grčkoj i Velikoj Britaniji.

Sedmo poglavlje prikazuje rezultate istraživanja i daje analizu dobijenih rezultata. Prezentovani su i rezultati ankete o stavovima studenata o računarski adaptivnom testu.

Osmo, završno poglavlje, sumira rezultate rada, daje zaključna razmatranja i ukazuje na moguće dalje pravce istraživanja. Na kraju, u priložima je data anketa, programski kod i spisak literature.

## 2. Testovi znanja

U sistemu vaspitavanja i obrazovanja, proveravanje i vrednovanje znanja ima poseban značaj i položaj kao deo od suštinskog značaja. U prilog tome govori i činjenica da je ocenjivanje formalno regulisano posebnim propisima, da postoji naučna disciplina dokimologija koja se bavi pitanjima procenjivanja znanja, a nastavnici imaju obavezu da izučavaju problem ocenjivanja i da tokom rada proveravaju i usavršavaju sopstvena merila procenjivanja znanja i davanja ocena. Proveravanje i ocenjivanje znanja omogućava nastavnicima da saznaju u kojoj meri su studenti usvojili nastavne sadržaje i stekli određena znanja i da dobiju povratnu informaciju o svom nastavnom radu, o primenjenim nastavnim metodama i oblicima rada kako bi ga mogli poboljšati.

Ocena osim pedagoškog uticaja ima i druge implikacije, u prvom redu psihološke i socijalne. Student putem ocenjivanja dobija povratnu informaciju o svom radu, iskazanom znanju i zalaganju, kao i o vrednosti metoda učenja i svog angažovanja u ostvarivanju školskih zahteva. Ocenjivanje je jedan od bitnih faktora školske sredine koji utiče na formiranje osobina ličnosti, na izgrađivanje odgovarajuće slike o sebi, kao i stvaranje osećaja sigurnosti i samouverenosti. Počevši od osnovne škole, učenikove ocene utiču na njegovu afirmaciju u odeljenju i u školi, među drugovima i u porodici. Uspeh učenika u školi izražen kroz ocene određuje mogućnost izbora određene škole, fakulteta, zanimanja, dobijanje stipendija, a konačno ima ishodište u društvenom i materijalnom položaju pojedinca.

U aktuelnom sistemu obrazovanja u upotrebi je nekoliko metoda proveravanja znanja i ocenjivanja koje pored nesavršenosti i nedostataka imaju upotrebnu vrednost:

- Usmeno proveravanje znanja koje se odvija kroz individualan razgovor sa studentom.
- Pisano proveravanje kroz testove, kolokvijume, pismene zadatke.
- Praktično proveravanje kroz proveru određenog znanja i veština u formi praktičnih ili grafičkih radova.

Ocene se izražavaju kvantitativno, numerički, kvalitativno, odnosno opisno i rangiranjem ili analitički.

Kriterijumi ocenjivanja uspeha studenta su vrsta, obim i nivo usvojenih znanja, umenja i veština u odnosu na propisano nastavnim planom i programom visokoškolske ustanove. Da bi proveravanje znanja i ocenjivanje imalo pravilan efekat, od velike je važnosti da nastavnikova procena znanja studenta bude tačna, objektivna i pouzdana.

Istraživanja koja su rađena ukazuju na veći broj faktora koji se javljaju namerno ili nenamerno u evaluaciji znanja time umanjujući objektivnost. Ovi faktori mogu se grupisati u nekoliko kategorija [44]:

Činioci koji potiču od nastavnika, najbrojniji su i najviše utiču na subjektivnost ocenjivanja:

- Lična praksa ocenjivanja – tendencija neopravdanog podizanja ili spuštanja kriterijuma i davanje pretežno niskih ili pretežno visokih ocena koja je posledica stalnih ili povremenih uzroka. Lično shvatanje nastavnog procesa, takozvana „lična filozofija“ i shvatanje navodne „važnosti“ predmeta, kao i stil ponašanja nastavnika i odnos prema studentima (demokratski, autokratski) utiču na stil ocenjivanja nastavnika.
- Pogreška sredine – nastavnik pretežno ocenjuje srednjim ocenama kako bi izbegao mogućnost velike greške u proceni znanja.
- Halo efekat – tendencija da se ponašanje i osobine osobe smatraju usklađenim i povezanim, pa nastavnici procenjuju znanje studenta prema nekom svom ranijem stavu i ocenama o tom studentu ili prema stavovima (ocenama) drugih nastavnika.
- Shvatanje važnosti gradiva – nastavnici istog predmeta mogu različito da vrednuju važnost pojedinih delova nastavnih sadržaja i to doprinosi subjektivnosti ocenjivanja.
- Kontrast – tendencija da kvalitet prethodno procenjenog znanja drugih studenata utiče na procenjivanje znanja pojedinca.
- Prilagođavanje – u zavisnosti od nivoa znanja cele grupe jedan isti odgovor može biti različito ocenjen u zavisnosti da li je grupa „jača“ ili „slabija“.
- Simpatije ili antipatije prema pojedinim studentima.
- Samovolja – aktivnost ili pasivnost nastavnika u toku odgovaranja studenta, način postavljanja pitanja i davanje posebnih, težih pitanja nedisciplinovanim studentima uzroci su subjektivnosti.

Činioci koji potiču od studenta i koji se odnose na njegove verbalne i druge sposobnosti i način formulisanja odgovora [48]:

- Student koji se lako i tečno izražava ostavlja bolji utisak i najčešće dobija bolju ocenu.
- Inteligentniji studenti su u stanju da prate reakcije nastavnika i brzo se ispravljaju u odgovorima i često su uspešniji u odnosu na trud koji ulažu.
- Kvalitet odgovora zavisi i od trenutnog zdravstvenog i emocionalnog stanja studenta. Trema može da izazove nesigurnost, blokadu i trenutno zaboravljanje.
- Fizički izgled studenta može da utiče na ocenjivanje. Pokazuje se da su lepši, uredniji, simpatičniji studenti u povoljnijem položaju.

Interakcija studenta i nastavnika, odnosno stav studenta prema nastavniku i stav nastavnika prema studentu utiču na subjektivnost ocenjivanja.

U ovim uslovima, vrednost dobijenih rezultata je pod znakom pitanja. Kvalitet ovih informacija zavisi od objektivnosti evaluacionog procesa i kvaliteta primenjenih metoda proveravanja znanja.

Prava snaga procene znanja ogleda se u povratnim podacima studentima. Pобоljšanje kvaliteta procesa učenja ne obuhvata samo određivanje završnog studentskog znanja na kraju kursa, već je još važnije merenje usvojenog znanja tokom kursa. Time studenti bivaju snažnije motivisani uspehom u učenju, preuzimaju više samoodgovornosti za sam tok učenja, sami otkrivaju svoje „jake i slabe tačke“ i tako postaju aktivni učesnici u učenju.

Kao najobjektivnije sredstvo za merenje znanja predloženi su testovi znanja ili postignuća. Oni se ponekad nazivaju i nastavni testovi. Testovi znanja su testovi pomoću kojih se određuje koliko je znanja pojedinac stekao kroz određenu aktivnost ili određeni period učenja. Oni se sastoje iz zadataka (datih u specijalnim oblicima) pomoću kojih se ispituje znanje učenika iz celokupnog gradiva jednog predmeta, ili pojedinih njegovih delova.

## 2.1. Blumova taksonomija

Američki psiholog Bendžamin Blum (*Benjamin Bloom*, 1913-1999.), 1956. godine je sa svojim saradnicima razvio klasifikaciju nivoa učenja. On je primetio da 95% pitanja na testovima zahteva od učenika da misle samo na najnižem nivou, odnosno da se od njih zahteva ponavljanje informacija. Prema Blumu, postoje tri domena obrazovanja, odnosno načina usvajanja određenog sadržaja [22]:

- kognitivna (znanje)
- afektivna (stavovi)
- psihomotorička (veštine).

U okviru kognitivne kategorije, Blum razlikuje šest hijerarhijskih nivoa učenja koji su predstavljeni na slici 2.1.





Slika 2.1: Kognitivni domeni prema Blumu

Hijerarhijski nivoi, počevši od najjednostavnijeg nivoa prema najzloženijem su sledeći [131]:

**1. Činjenično znanje:** Usvajanje činjeničnog znanja je najniži obrazovni cilj. Znanje se definiše kao sećanje na ranije naučene sadržaje. Odnosi se na temeljna znanja koja student mora steći da bi shvatio smisao predmeta koji uči. To se prisećanje može odnositi na široki raspon sadržaja: od usvajanja terminologije, preko prisećanja na specifične činjenice, pa sve do sećanja na složene teorije. Sve što treba postići na tom nivou znanja jeste priseliti se određene informacije, što ne mora nužno značiti i razumevanje. Na primer, student treba memorisati, definisati, opisati, označiti, nabrojati, prepoznati.

**2. Razumevanje:** Razumevanje se definiše kao sposobnost promišljanja o značenju usvojenih činjenica. Ta se kognitivna kategorija znanja može pokazati interpretiranjem naučenih činjenica, sažimanjem, objašnjavanjem ili predviđanjem rezultata ili posledica. Ovaj obrazovni cilj je viši od prethodnog jednostavnog prisećanja na informacije i predstavlja najniži stepen razumevanja. Na primer, za taj nivo znanja student treba da zna interpretirati slike, karte, tabele i grafikone, verbalne zadatke prevesti u formule, na temelju činjenica predvideti posledice, dati primer, interpretirati, parafrazirati.

**3. Primena:** Primena se odnosi na sposobnost primene naučenih pravila, zakona, metoda ili teorija u novim, konkretnim situacijama. Na primer, na tom spoznajnom nivou student treba da zna da reši matematički problem, konstruiše grafikon ili krivu, demonstrira ispravnu upotrebu neke metode ili postupka.

**4. Analiza:** Na analitičkom nivou znanja student mora biti sposoban da naučene sadržaje razdvoji na sastavne delove i razume organizacijsku strukturu. Pri tome student mora znati da odredi sastavne delove i odnose među njima kao i organizacijske principe. Ovaj obrazovni cilj je viši od nivoa razumevanja i nivoa primene jer je za taj nivo znanja potrebno združeno razumevanje sadržaja i organizacijske strukture

materijala. Na primer, na tom nivou student mora upoređivati, suprotstavljati, prepoznati neizrečene pretpostavke, razlikovati činjenice od zaključaka, razlikovati uzrok od posledice, odrediti relevantnost podataka, analizirati organizacijsku strukturu dela (umetničkog, muzičkog, literarnog).

**5. Sinteza:** Sintetizovati znači iz pojedinačnih delova stvoriti novu celinu. Obrazovni cilj u ovom slučaju ističe kreativno ponašanje s naglaskom na formulisanje novih obrazaca ili struktura. Primeri obrazovnih ciljeva sintetičkog nivoa znanja su: sposobnost kombinacije, postavljanja hipoteze, planiranja, reorganizacije, pisanja dobro organizovanog rada, držanje dobro organizovanog govora (predavanja), kreativno napisati priču (pesmu, muziku), predložiti plan eksperimenta.

**6. Procena:** Procena znači sposobnost svrsihodne procene vrednosti materijala (pesme, romana, govora, istraživačkog izveštaja, projekta). Procene se moraju bazirati na tačno definisanim kriterijumima. Obrazovni ciljevi ovog područja su najviši u spoznajnoj hijerarhiji jer sadrže elemente svih prethodnih nivoa uz dodatak sposobnosti procene vrednosti zasnovane na tačno definisanim kriterijumima. Primeri obrazovnih ciljeva ovog nivoa znanja jesu: proceniti primerenost zaključaka iz prikazanih podataka, proceniti vrednost nekog dela (umetničkog, muzičkog, literarnog), proceniti logičnu postojanost pisanog materijala ili predavanja.

## 2.2. Pisana provera znanja

U sistemu visokog obrazovanja najčešće se procenjuje kognitivno znanje, a za to su najpogodnije pisane procene znanja (u papirnatom ili elektronskom obliku), te se one najviše i primenjuju.

U zavisnosti od svrhe, posledice pisane provere znanja mogu biti teže ili lakše. Na primer, od rezultata završnog ili prijemnog ispita zavisi budućnost studenta, dok pisane samoprocene imaju edukacijski zadatak da poboljšaju pamćenje i da daju studentu povratnu informaciju. S obzirom na posledice, pisane procene znanja se mogu podeliti u sledeće kategorije [131]:

- **ispit** (engl. *exam*) = završna procena znanja (engl. *final summative assessment*) radi dobijanja dokaza o aktuelnom nivou i kvalitetu znanja, veština i stavova.
- **kviz** (engl. *quiz*) = samoprocena znanja (engl. *selfassessment, formative assessment*) kojoj je cilj da student dobije povratnu informaciju o aktuelnom nivou vlastitog znanja.
- **anketa** (engl. *survey*) = dijagnostička procena znanja, veština i stavova cele grupe studenata kojoj je cilj utvrditi stepen usaglašenosti s propisanim nastavnim ciljevima.
- **test** (engl. *test*) = dijagnostička procena znanja radi informisanja studenta i nastavnika o dostignutom nivou i kvalitetu znanja koja pomaže studentu da ostvari nastavne ciljeve (testom se procenjuje znanje iz manjeg opseg gradiva nego ispitom, kolokvijumom).

U zavisnosti o kojoj kategoriji se radi (kakve posledice nosi sa sobom), razlikuje se i stepen motivisanosti studenata za polaganja i/ili varanje, što su vrlo važni elementi pri izboru sistema kojim će se vršiti pisano procenjivanje znanja [131].

Za pisanu proveru znanja mogu se koristiti različite vrste pitanja. Izbor vrste pitanja u najvećoj meri zavisi od obrazovnih ciljeva, svrsi provere i broju studenata. Iako nema oštih granica, neke su vrste pitanja pogodnije za procenu nižih nivoa znanja, a neke za vrednovanje viših kognitivnih domena.

### 2.2.1. Vrste testova znanja

Test mora biti valjan, što znači da mora ispunjavati uslove objektivnosti, osetljivosti i primerenosti po težini [70].

Pod objektivnošću se podrazumeva nezavisnost rezultata testa od nastavnika koji ga provodi; pod osetljivošću testa podrazumeva se finoća mernih skala, odnosno adekvatan raspored bodova za zadatke različite težine, tako da lakši zadatak mora biti vrednovan sa manjim brojem bodova od težeg zadatka, a raspored lakih i teških zadataka mora biti u izvesnoj realnoj harmoniji; primerenost testa po težini podrazumeva ukupnu zahtevnost testa u odnosu na učeničku populaciju. Test u celini ne sme biti ni prelagan, ni pretežak. Njegovi rezultati trebalo bi da odgovaraju grafikonu idealne Gausove krive, što znači da broj laganih, srednje teških i teških zadataka mora biti uravnotežen tako da najveći procenat učenika može rešiti zadatke srednje težine. Težina zadataka, odnosno zahtevnost testa, ne sme izlaziti iz ishoda učenja propisanih nastavnim programom [70].

Testovi znanja mogu se klasifikovati prema:

1. načinu i vremenu zadavanja
2. kriterijumu u odnosu na koji se procenjuje znanje učenika
3. načinu konstruisanja

U tabeli 2.1 prikazana je podela testova prema vremenu zadavanja i načinu korišćenja.

Tabela 2.1: Podela testova prema vremenu i načinu korišćenja [19]

	Formativni	Sumativni
<b>vreme zadavanja</b>	pre ili u toku podučavanja	na kraju podučavanja ili učenja
<b>svrha/cilj</b>	oblikovanje nastavnog procesa i procesa učenja – potpora učenju	procena znanja stečenog u toku procesa učenja – validacija, evaluacija, akreditacija procesa
<b>ocenjivanje</b>	ne ocenjuje se	ocenjuje se

Prema kriterijumu u odnosu na koji se procenjuje znanje, testovi se mogu podeliti na [177]:

- **Kriterijumsko ocenjivanje** – zasnovano na unapred utvrđenim standardima i ciljevima koji treba da se ostvare u toku procesa podučavanja, pri čemu su kriterijumi za svaku ocenu utvrđeni unapred.
- **Normativno ocenjivanje** – postignuće učenika se procenjuje u odnosu na postignuće ostalih učenika.

Prema načinu konstruisanja, testovi se mogu podeliti na:

- baždarene (pravi testovi) i
- nebaždarene testove znanja (nizovi zadataka objektivnog tipa).

Testovi imaju isti „spoljašnji“ izgled, a razlikuju se po tome ko je konstruktor testa, načinu konstrukcije, obimu gradiva koje test pokriva i oblasti primene. Tabela 2.2 prikazuje razlike u testovima prema načinu konstruisanja.

Pravi testovi znanja, imaju sve metrijske karakteristike testova uopšte, baždareni su i obično su izrađeni od strane ekipe stručnjaka [98]. Oni obuhvataju ceo program za jedan predmet određenog kursa (razreda) i upotrebljavaju se u većem broju različitih obrazovnih institucija. Nizovi zadataka objektivnog tipa sastavljeni su od zadataka istog oblika kao testovi znanja, ali njih obično izrađuje nastavnik za svoju upotrebu. Oni nisu baždareni i nemaju određene metrijske karakteristike, tako da se ne mogu upotrebiti u većem broju škola i na osnovu njih se mogu vršiti poređenja znanja učenika samo iz jednog odeljenja ili iz nekoliko odeljenja u kojima isti nastavnik predaje jedan predmet. Mada nizovi zadataka objektivnog tipa nemaju vrednost testova znanja, oni mogu nastavniku veoma korisno poslužiti: pomoću njih on može objektivno pratiti napredovanje pojedinih učenika i celog razreda, proveravati da li su učenici razumeli neki deo gradiva ili proveriti upotrebu neke nove metode u obradi određenog dela gradiva, ispitati predznanje učenika koje im je potrebno za savladavanje novog gradiva itd. Oni takođe, mogu da obuhvate celokupno gradivo jednog predmeta, mada obično obuhvataju samo jedan određeni deo gradiva. Nastavnik prilikom ocenjivanja nikad nije u nedoumici da li je neki odgovor tačan ili nije, pošto ima utvrđeno merilo za ocenjivanje svakog zadatka. U jednom pogledu zadaci objektivnog tipa, koje prave sami nastavnici, imaju izvesne prednosti nad pravim testovima znanja. Pošto ih pravi sam nastavnik, oni mogu da obuhvate sve pojedinosti i finese koje je nastavnik obradio u tom predmetu. Testovi znanja se, međutim, prave prema opštem programu za jedan razred i namenjeni su upotrebi u većem broju škola, te mogu da obuhvate samo one osnovne stvari koje se moraju obraditi u svim školama, a iz njih moraju da se izostave svi detalji za koje se može pretpostaviti da mogu biti izostavljeni ili da su obrađeni na nešto drugačiji način u različitim školama [97].

Tabela 2.2: Podela testova prema načinu konstruisanja [19]

	baždareni	nebaždareni
<b>konstruktor</b>	tim stručnjaka	predmetni nastavnik
<b>način konstruisanja</b>	veliki broj pitanja iz svih oblasti izbor oblika testa znanja probno zadavanje (metrijske karakteristike) konačna verzija sa zadacima različite težine	
<b>obim gradiva</b>	celokupno gradivo predmeta	deo gradiva
<b>oblast primene</b>	sve škole sa istim nastavnim planom i programom	učenici kojima predaje nastavnik koji sastavlja test

Prema obliku u kome se zadaju, zadaci se u testovima znanja dele na testove reprodukcije i testove rekognicije ili prepoznavanja [97]. U testovima reprodukcije od učenika se traži da sami nađu, reprodukuju odgovor na postavljeno pitanje. U testovima rekognicije, učenik treba od ponuđenih odgovora da odabere, prepozna onaj koji je tačan. U svakoj od ovih grupa postoji nekoliko oblika zadataka koji se upotrebljavaju u testovima znanja.

U testove reprodukcije spadaju [97]:

1. esejski testovi i
2. testovi dopunjavanja ili prisećanja.

U testove rekognicije ili prepoznavanja spadaju:

1. alternativni testovi ili testovi dvočlanog izbora
2. testovi višestrukog izbora i
3. testovi upoređivanja i sređivanja.

U tabeli 2.3 data je podela pismenih vrsta pitanja [131].

Tabela 2.3: Podela pismenih vrsta pitanja [19]

Vrste pismenih pitanja			
Pitanja s odabirom odgovora (treba izabrati ponuđeni odgovor)		Pitanja bez odabira odgovora (treba upisati odgovor)	
Vrsta	Kratak opis	Vrsta	Kratak opis
<b>Pitanja višestrukog izbora odgovora</b>	samo jedan od ponuđenih odgovora je tačan	<b>Esej</b>	
<b>Pitanja višestrukog izbora odgovora</b>	više ponuđenih odgovora može biti tačno	<b>Upisivanje tekstualnog odgovora</b>	uz pitanje nema ponuđenih odgovora već postoji prazan prostor u koji student upisuje odgovor
<b>Tačno / Netačno</b>	pitanje je tvrdnja koju treba proceniti da li je tačna ili nije (postoje samo 2 izbora)	<b>Unos brojčanog odgovora</b>	student upisuje brojnu vrednost u prazan prostor
<b>Sparivanje pojmova</b>	međusobno treba tačno spariti pojmove složene u 2 kolone	<b>Upisivanje izbrisanih reči</b>	student upisuje reči koje nedostaju u tekstu

Karakteristike ovih pitanja su sledeće [131]:

**Pitanja sa odabirom odgovora** (engl. *selected – response questions*). Ova vrsta pitanja može biti vrlo korisna i delotvorna za određivanje znanja studenata i načina rasuđivanja. Mnogi od standardizovanih testova se baziraju na ovoj vrsti pitanja, pa je neophodno da nastavnici budu vični u sastavljanju pitanja. Pored toga, nastavnici bi trebalo da budu u mogućnosti da dizajniraju kvizove koji ciljaju ka višim nivoima znanja shodno Blumovoj taksonomiji (primena, analiza, sinteza). Pitanja sa odabirom odgovora uključuju sledeće vrste pitanja: pitanja višestrukog izbora, sparivanje pojmova i tačno/netačno.

**Pitanja višestrukog izbora odgovora sa samo jednim tačnim odgovorom** (engl. *multiple choice question – MCQ*). Ova vrsta pitanja naročito je pogodna ukoliko se računar koristi za sprovođenje testiranja. Omogućavaju jednostavno, precizno i objektivno bodovanje i znatno pojednostavljaju rad nastavnika pri ocenjivanju. Prema Blumovoj taksonomiji moguće je ispitivati različite kognitivne nivoe znanja i primenjivati ih kako za samotestiranje tako i za završne (sumativne) ispite. Statistička obrada celokupnog ispita je jednostavna i brza, kao i utvrđivanje težine pojedinih pitanja i diskriminacijske sposobnosti pitanja. Testovi sačinjeni od MCQ pitanja omogućavaju sveobuhvatnije ispitivanje sadržaja predmeta nego pitanja esejskog tipa. Student nema mogućnost zaobilaženja sadržaja i pojednostavljenja teme.

Problem kod ove vrste testova je što je sastavljanje kvalitetnih MCQ pitanja složeno i vremenski zahtevno. Nastavnici su skloni sastavljanju pitanja kojima se ispituje samo memorisanje činjeničnog znanja jer su takva pitanja najlakša za osmišljavanje. Određeni postotak odgovora može se slučajno pogoditi, jer se do tačnog odgovora može doći metodom isključivanja netačnih opcija. Kod MCQ sa pet ponuđenih odgovora prag za prolaz je oko 60% tačno odgovorenih pitanja. Ova vrsta pitanja nije pogodna za testiranje kreativnosti koja se najbolje ispituje esejskim tipom pitanja.

**Pitanja višestrukog izbora odgovora s više tačnih odgovora** (engl. *multiple response question*). Prednost ove vrste pitanja je vrlo laka automatska obrada rezultata, a nedostatak je veća mogućnost pogađanja.

**Tačno/Netačno** (engl. *True/False question*). Predstavlja podvrstu MCQ pitanja sa samo dve opcije – „tačno“ i „netačno“, pogodnih za ispitivanje činjeničnog znanja. Pitanja se pišu u obliku tvrdnje čiju je istinitost potrebno proceniti. Tvrdnju je potrebno napisati drugim rečima u odnosu na ono kako je napisana u udžbeniku ili nastavnom materijalu, kako jednostavno memorisanje ne bi bilo dovoljno za odabir tačnog odgovora. Pri pisanju tvrdnje potrebno je izbegavati reči koje bi mogle sugerisati tačnost/netačnost odgovora, kao na primer „nikada“, „ništa“, „uvek“, „sve“ koje imaju tendenciju ka netačnosti, a kvalifikacije poput „obično“, „uopšteno“, „ponekad“ ili „često“ su verovatno tačne. Poželjno je izbegavati negacijske tvrdnje, a preporuka je da u testu bude otprilike jednak broj tačnih i netačnih tvrdnji. Testovi ovog tipa su pogodni za kratko formativno samotestiranje kojim student i nastavnik brzo dobijaju povratnu

informaciju o stepenu usvojenog činjeničnog znanja. Verovatnoća da se bez stvarnog znanja pogodi tačan odgovor je 50%, pa prag za prolaz na celokupnom testu mora biti najmanje 75% [131].

**Sparivanje pojmova** (engl. *matching list*). Predstavlja podvrstu MCQ pitanja u kojoj se pojmovi složeni u dve kolone moraju međusobno tačno povezati (spariti). Ova vrsta pitanja pogodna je i za formativno samotestiranje kojim student i nastavnik dobijaju brzu povratnu informaciju o stepenu usvojenog znanja, ali i za završne sumativne ispite. Omogućavaju proveru činjeničnog znanja, ali dobrom konstrukcijom pitanja može se postići provera i viših kognitivnih nivoa znanja, kao na primer uzročno-posledično povezivanje. Važno je da informacije u obe kolone budu što je moguće više homogene i da bude veći broj ponuđenih odgovora za sparivanje od pojmova koji predstavljaju pitanje.

**Pitanja bez odabira odgovora** (engl. *constructed – response*). Kod pitanja sa odabirom odgovora, odgovor je vidljiv, student samo treba da ga prepozna među ponuđenim odgovorima. Pitanja bez odabira odgovora nemaju ponuđene odgovore i obično se koriste za procenu viših nivoa znanja prema Blumovoj taksonomiji. U ovu vrstu spadaju sledeće vrste pitanja: upisivanje odgovora, upisivanje izbrisanih reči i esej.

**Upisivanje odgovora** (engl. *text match*). Uz pitanje se ne nudi odgovor, već postoji prazan prostor u koji student upisuje odgovor. Za razliku od MCQ pitanja gde student bira između ponuđenih odgovora, te postoji mogućnost da i bez znanja tačno odgovori (eliminacijom ili srećom), u ovoj vrsti pitanja student mora znati tačan odgovor. Ukoliko se radi o pitanju koje traži unos brojnog podatka, moraju se predvideti sve varijante koje se smatraju tačne, na primer  $\pi$  i 3.14.

**Upisivanje izbrisanih reči** (engl. *fill in the blank*). Omogućava ispitivanje i viših nivoa znanja, ali je nedostatak teža automatska obrada jer se moraju predvideti svi oblici tačnog odgovora.

**Esej.** Predstavlja jedinu ispitnu metodu kojom se može proceniti sposobnost studenta da samostalno pismeno obrazloži odgovor. Ispituje više kognitivne nivoe znanja (analiza, sinteza, evaluacija). Postoje dve vrste eseja: dugi, prošireni kojima se ispituje jedan uzorak gradiva na višem kognitivnom nivou znanja (kreativnost) i kratki, ograničeni. Sa nekoliko kratkih eseja može se površnije obraditi veći broj uzoraka ispitnog gradiva. Nedostatak ovog tipa pitanja je što zahteva veliki utrošak vremena za ocenjivanje, a moguća je i nepouzdanost i neobjektivnost u ocenjivanju. Preporuka je da se ova vrsta pitanja primenjuje za vrednovanje onih vrsta sposobnosti koje se ne mogu uspešno proceniti nekom drugom metodom. Potrebno je odabrati zadatak koji je primeren nastavnim ciljevima i predstavlja reprezentativni uzorak gradiva.

### 2.2.2. Životni ciklus pitanja

Životni ciklus pitanja može se podeliti na [131]:

- priprema za test,
- sprovođenje (isporuka) testa,
- procena rezultata,
- vrednovanje kvaliteta testa,
- unos povratnih podataka u bazu.

**Priprema za test** započinje autorovim stvaranjem pitanja. Informacione tehnologije pružaju autorima alate za lako stvaranje pitanja, ispravljanje, prikupljanja informacija, administriranje i pohranjivanje u bazu podataka. Pojedinačna pitanja treba pregledati i oceniti ih od nekoliko nastavnika (najmanje 3). Aktivan život pitanja započinje odabirom pitanja za test. Izbor može obaviti nastavnik pre testa ili sistem koji automatski generiše test slučajnim ili nekim drugim zadanim načinom izbora. Nasuprot tome, informacione tehnologije (IT) omogućavaju da sistem dinamički generiše test tokom ispitivanja, prema nekim unapred zadanim zakonitostima, što omogućuje jedinstvene testove za svakog ispitanika ili grupu ispitanika. Postojeće IT nude vrlo velik broj različitih alata koji podržavaju sve faze pripreme i održavanje baze pitanja.

**Sprovođenje (isporuka) testa.** Drugo razdoblje je isporuka testova. Tokom testa se sakupljaju odgovori ispitanika za vrednovanje (to može biti klasičan papirnati obrazac). Pored toga, ispit se može vršiti i na računaru, pa se odgovori beleže u datoteku, e-mail upućen nastavniku ili u ispitnu bazu. Ispit se može izvršavati i *online* sa servera. Pre isticanja rezultata treba obaviti vrednovanje kvaliteta pojedinih pitanja i eventualne ispravke testa, pa naravno i rezultata.

**Procena rezultata testa** je faza u kojoj sistem vrši klasifikaciju datih odgovora da li su tačni ili netačni, sabira ih, ocenjuje, objavljuje rezultate i skladišti ih. Ukoliko se ispit radi na računaru, obrada rezultata je vrlo brza i jednostavna, te se rezultati znaju odmah nakon testa.

**Vrednovanje kvaliteta testa.** Pre objave rezultata preporučuje se da se izvrši vrednovanje kvaliteta testa (*item* analiza) i zatim, ukaže li se potreba, isprave rezultati.

**Unos povratnih podataka u bazu.** Nakon završetka testa treba sve sakupljene povratne informacije uneti u bazu pitanja, a podatke o kvalitetu pitanja dostaviti autorima.



### 2.2.3. Pravila pisanja ispitnih pitanja s više ponuđenih odgovora

U računarskim adaptivnim testovima najčešće se koriste pitanja s više ponuđenih odgovora, te slede neke preporuke o kojima bi trebalo voditi računa pri sastavljanju pitanja.

Pitanja s više ponuđenih odgovora imaju tri obavezna dela [131]:

1. formulisanje pitanja, „telo pitanja“
2. tačan odgovor, „ključ pitanja“
3. nekoliko netačnih odgovora, alternativa tačnom odgovoru, tzv. „distraktori pitanja“ i opcionalno (posebno vredno pri samoprocenjivanju znanja)
4. povratna informacija kojom se komentariše studentov odgovor.

Sledi primer pitanja s više ponuđenih odgovora:

*Koja je vrednost varijable min, nakon izvršenja sledećih izraza:*

```
int m = 1, n = 2;  
int min = (m < n ? m++ : n++);
```

- a. 0 (*distraktor*)
- b. 0.5 (*distraktor*)
- c. 1 (*ključ*)
- d. 2 (*distraktor*)
- e. 3 (*distraktor*)

Napisati dobro ispitno pitanje s više ponuđenih odgovora je veština koja se stiče iskustvom. Vrlo je važna povratna informacija koju autori pitanja dobijaju nakon testa analizom studentskih odgovora (*item analiza*). Postoji nekoliko pravila kojih bi se trebalo pridržavati kako bi se poboljšao kvalitet pisanja pitanja [131]:

**1. Ispitivati samo bitno i važno.** Trebalo bi da se vodi računa da se svakim pitanjem ispituje samo bitno i važno znanje. Treba izbegavati pitanja koja se odnose na sitne detalje. Svako pitanje treba biti relevantno unapred postavljenim ciljevima predmeta.

**2. Pisati jednostavnim jezikom.** Pitanja bi trebalo da se pišu jednostavnim jezikom, pazeći na pravopis i gramatiku. Pravopisnim i gramatičkim nepravilnostima (osim ako se ne ispituje gramatika i pravopis) zbunjuju se studenti.

**3. Pisati kratka i jasna pitanja.** Telo pitanja je potrebno očistiti od svih suvišnih reči i nerelevantnih sadržaja. Poželjno je pitanja formulirati tako da se najveći deo teksta nalazi u telu pitanja bez ponavljanja delova teksta u ponuđenim odgovorima.

**4. Ispravno formulirati pitanje.** Trebalo bi da se pazi da se u formulaciji pitanja indirektno ne skriva ključ za nalaženje tačnog odgovora. Student (vičan rešavanju

testova) to lako prepoznaje, pa tačan odgovor pronalazi zahvaljujući igri reči, gramatici i sl., a ne zahvaljujući stvarnom znanju.

**5. Voditi računa o nezavisnosti pitanja.** Trebalo bi da se vodi računa da se unutar istog tematskog područja u pitanjima ne ponavlja isti sadržaj i termini, jer tada odgovor na jedno pitanje može postati ključ za pronalaženje odgovora drugog pitanja.

**6. Ponuditi ujednačene odgovore.** Svi ponuđeni odgovori trebalo bi da su ujednačeni, jasni i realni. Na primer, neverovatnost nekih od ponuđenih distraktora ili neujednačena dužina teksta ponuđenih odgovora mogu upućivati na tačan odgovor, pa takvim pitanjem se ne ispituje stvarno znanje. Položaj ključa u nizu ponuđenih odgovora treba biti nasumičan. Ponuđene brojčane odgovore treba poredati po rastućim vrednostima.

**7. Izbegavati postavljanje negacijskih pitanja.** Ako se u telu pitanja koristi negacija, trebalo bi da se obavezno naglasi velikim slovima, na primer „Šta od navedenog NIJE tačno za ...“ ili „Od navedenih tvrdnji ... sve su tačne, OSIM:“.

**8. Izbegavati distraktor tipa „svi ponuđeni odgovori su tačni“ ili „ništa od navedenog nije tačno“.** Nastavnici najčešće koriste navedene tvrdnje kada im ponestane ideja za poslednji distraktor. Studenti, naslućujući „filozofiju“ takvih pitanja, retko se daju zavesti tim distraktorom. Stoga, ako se koriste, obvezno ih je ponekad potrebno koristiti kao tačan odgovor (ključ). Takođe, kada student prepozna da postoje dva tačna odgovora (od npr. 5 mogućnosti) moći će zaključiti bez znanja (ne)tačnosti ostalih distraktora da se ključ tačnog odgovora nalazi u tvrdnji „svi su ponuđeni odgovori tačni“.

**9. Distraktori se moraju bitno razlikovati od tačnog odgovora (ključa).** Loši distraktori su oni koji se od tačnog odgovora (ključa) razlikuju u blagim nijansama. Dobri (jaki) distraktori su tvrdnje koje su same po sebi tačne, ali nisu odgovor na postavljeni problem u telu pitanja.

**10. Ponuditi odgovarajući broj distraktora.** Što je veći broj distraktora, to je manja verovatnoća da će student slučajno pogoditi tačan odgovor (ključ). U visokoškolskim testovima se najčešće koriste pitanja s 5 ponuđenih odgovora (1 ključ + 4 distraktora). Znači da je verovatnoća slučajnog pogađanja 20%.

#### 2.2.4. Analiza nakon sprovedenog testa

Statistička analiza testa (*item analiza*), koja se sprovodi nakon ispravljanja obrazaca ili sakupljanja odgovora *online*, ima dvostruku ulogu [131]:

- korekcija konačnih rezultata testa, ako se nađu pitanja koja valja izbaciti ili korigovati tačan odgovor
- ispravljanje pitanja u bazi i dobijanje smernica za pisanje novih pitanja.

Statističku analizu testa čini [131]:

1. Osnovna deskriptivna statistika – izračunava se broj studenata na testu, srednja vrednost, medijan, standardna devijacija, najviši i najniži rezultat.
2. Indeks teškoće pojedinog pitanja
  - a. Teškoća (odnosno lakoća) svakog pitanja (P) je udeo tačnih odgovora za svako pitanje. Određuje se nakon testa izračunatim procentom tačnih odgovora.

**Lakoća pitanja** = ukupan broj tačnih odgovora/broj izašlih na test  
**Težina pitanja** = 1 – lakoća pitanja

Pitanja na koja odgovaraju gotovo svi ili niko su „loša“, tj. preteška ili prelagana i valja ih izbaciti iz testa, test treba revalorizovati, a pitanja ispraviti u bazi za narednu upotrebu. Preporučuje se iz psiholoških razloga da par prvih pitanja u testu bude lakši (oko 0.9). Takođe, prelagana pitanja su ipak prihvatljiva, ako se radi o dijagnostičkom testu ili prethodnom testu za proveru znanja ne samo pojedinca već posebno grupe polaznika.

- b. Optimalna teškoća MCQ pitanja s  $n$  predloženih odgovora može se izračunati iz formule:  **$P = 0.5 + 0.5 (1/n)$**
    - c. Za pitanja tipa tačno/netačno optimalna je teškoća 0.75, a prihvatljiv raspon od 0.65 do 0.85. Pitanja ispod, ili iznad preporučenog raspona treba ispraviti.
    - d. Najbolja MCQ pitanja s 3 distraktora imaju teškoću 0.67
    - e. Najbolja MCQ pitanja s 4 distraktora imaju teškoću 0.63
    - f. Najbolja MCQ pitanja s 5 distraktora imaju teškoću 0.60
    - g. Uopšteno govoreći prihvatljiva težina pitanja je od 0.3 do 0.7. Pitanja ispod 0.3 su preteška, a iznad 0.7 prelagana.
  3. Indeks diskriminativnosti pojedinog pitanja
    - a. Nakon testa se indeks diskriminacijske vrednosti pojedinačnih pitanja računa tako da se svi studenti ( $x$ ) podele u, uslovno rečeno, „lošije“ tj. donja trećina i u „bolje“, tj. gornja trećina na rang listi.
    - b. Za svako pitanje prebroji se broj tačnih odgovora u lošijoj (L) i u boljoj (B) grupi, pa se izračuna indeks pomoću formule:

**Diskriminativnost =  $2(B-L)/x$**

- c. Što je veći broj, pitanje je „bolje“. Iznad 0.35 su pitanja izvrsna. Između 0.35 do 0.25 su dobra, između 0.25 i 0.15 još prihvatljiva, ali ih treba za drugi test ispraviti, a pitanja ispod 0.1 treba izbaciti iz testa i test ponovo vrednovati bez njih. Vrlo laka, kao i vrlo teška pitanja slabo diskriminiraju studente po znanju. Međutim ipak se preporučuje, iz psiholoških razloga, staviti prva dva – tri pitanja testa, lakša od ostalih.
      - d. Drugi način merenja diskriminativnosti je izračunavanje koeficijenta biserijalne korelacije (eng. *point biserial correlation coefficient*, PBCC). Time se izračunava zapravo korelacija između učestalosti tačnog odgovora na određeno pitanje i ukupnog rezultata studenata. Zbog toga što uzima u obzir sveukupne rezultate svih studenata, ovaj je pokazatelj bolji od jednostavnog odnosa

diskriminativnosti. Poželjni koeficijent je između 0.2 i 0.8. Međutim, imaju li sva pitanja visok koeficijent (iznad 0.6) test će verovatno biti pretežak.

4. Učestalost pojedinih odgovora u svakom pitanju
  - a. Ukoliko postoje distraktori koji nikoga nisu „zaveli“, treba ih do sledećeg ispita promeniti. „Prejake“ distraktore treba proveriti da možda nisu tačni.
5. Ponovljivost testa
  - a. Treba izračunati Kuder Ričardson 20 (engl. *Kuder-Richardson formula 20*). KR20 izračunava koeficijent pouzdanosti na osnovu broja pitanja testa ( $k$ ), udela tačnih odgovora na pitanja ( $p$ ), deo odgovora koji su netačni ( $q$ ), i varijanse ( $\sigma^2$ ).

$$r = \frac{k}{k-1} \left( 1 - \frac{\sum pq}{\sigma^2} \right)$$

Dobar test ima 0.70 KR20, ili veći. Radi se o testiranju ujednačenosti testova u različitim rokovima. Kada bi isti student pristupio testu s visokom ponovljivošću (KR20) on bi postigao isti rezultat. KR20 zavisi i od dužine testa, tako da raste što je test obimniji.

### 3. Pregled literature

Zahvaljujući napretku i razvoju informaciono-komunikacione tehnologije, došlo je i do porasta primene računarskih testova. Isporuca testova putem računara omogućila je dobijanje licence, sertifikata i polaganje prijemnih ispita. Osim toga, računari se mogu koristiti za povećanje statističke tačnosti rezultata testa korišćenjem računarskih adaptivnih testova. Za razliku od testova gde svaki ispitanik dobija isti test sa fiksiranim brojem pitanja, kod računarskih adaptivnih testova izbor pitanja se prilagođava nivou sposobnosti svakog pojedinca i nakon svakog datog odgovora procenjena sposobnost se ažurira a sledeće pitanje se bira tako da ima optimalna svojstva za novoodređenu procenu [142]. Računar kontinualno preračunava sposobnost ispitanika dok tačnost procene ne postigne statistički zadovoljavajući nivo, ili dok se ne dostigne neka granica, kao na primer, maksimalan broj pitanja koji može da bude dat ispitaniku. Rezultat testa određuje se na osnovu nivoa težine pitanja na koja je ispitanik odgovarao. Kao rezultat toga, iako su svi ispitanici možda odgovorili tačno na isti procenat pitanja, oni koji se na lestvici sposobnosti nalaze pri vrhu, imaju bolji rezultat testa jer su tačno odgovorili na teža pitanja.

Istorija primene računara za proveru znanja (engl. *computer based assessment* CBA) počinje 70-tih godina XX veka. Međutim, tadašnja visoka cena računara i njihove tehničke mogućnosti ograničili su njihovu primenu u testiranju. Napredak tehnologije omogućio je razvoj i primenu računara za testiranje u mnogim oblastima, uključujući i obrazovanje [37].

Jedan od najranijih pokušaja primene računara kao pomoćnog sredstva u nastavnom procesu i za proveru znanja bio je 60-tih godina XX veka, kada je projekat PLATO (engl. *Programmed Logic for Automatic Teaching Operations*) započet na Univerzitetu Illinois (engl. *University of Illinois*). Projekat TICCIT (engl. *Time-Shared, Interactive, Computer-Controlled, Information Television*), započet 1967. godine, predstavlja još jedan primer primene računara u obrazovanju.

Istorija e-evaluacije uključuje i primenu računara za ocenjivanje zadataka studenata iz programiranja. Jedan od najranijih pokušaja primene računara za automatizaciju procesa provere zadataka iz programiranja bio je „automatski ocenjivač“ (engl. *automatic grader*) [59]. Automatski ocenjivač nije korišćen kao kompajler, nego je njegov zadatak bio da pomogne studentima da nauče da programiraju, a takođe je olakšavao nastavniku praćenje velikog broja studenata koji su pohađali kurs. Automatski ocenjivač, svoju primenu našao je i u učenju na daljinu. Ovaj sistem koristili su studenti sa Univerziteta Stanford (engl. *University of Stanford*) koji su pohađali kurs numeričke analize za ocenjivanje njihovih vežbi iz programiranja.

Tokom 80-tih godina i kasnije, počelo je sve veće interesovanje za primenu računara u nastavi, a koristili su se i za automatizaciju nekih zadataka nastavnog modela. Ubrzo nakon toga javljaju se sistemi za procenu znanja u oblastima kao što su matematika, hemija, učenje stranih jezika [6].

Najveći uticaj na primenu računara u obrazovanju tokom 90-tih godina imao je razvoj WWW, svetske mreže svih mreža (engl. *World Wide Web*). Od tada, sistemi provere znanja počinju da bivaju bazirani na *webu*, kao što su na primer WebCT, Blackboard, Moodle, ATutor, Claroline, Desire2Learn, da pomenemo samo neke od njih [170].

### 3.1. Prednosti i nedostaci računarskih testova

U sistemu vaspitavanja i obrazovanja, proveravanje i vrednovanje znanja ima poseban značaj i položaj kao deo od suštinskog značaja. U prilog tome govori i činjenica da je ocenjivanje formalno regulisano posebnim propisima, da postoji naučna disciplina dokimologija koja se bavi pitanjima procenjivanja znanja, a da nastavnici imaju obavezu da izučavaju problem ocenjivanja i da tokom rada proveravaju i usavršavaju sopstvena merila procenjivanja znanja i davanja ocena. Proveravanje i ocenjivanje znanja omogućava nastavnicima da saznaju u kojoj meri su studenti usvojili nastavne sadržaje i stekli određena znanja i da dobiju povratnu informaciju o svom nastavnom radu, o primenjenim nastavnim metodama i oblicima rada kako bi ga mogli poboljšati. Primenjuju se različite metode procene znanja u različitim kontekstima, kao na primer prezentacija na času, pisanje eseja, projekti, itd. Međutim, najčešći „alat“ koji se koristi za proveru znanja je usmeni ispit i test. Kako se računar kao nastavno sredstvo sve više koristio, tako se njegova upotreba širila na svim nivoima obrazovanja, pa se počeo primenjivati i za proveru znanja, te se javljaju i testovi koji se rade na računaru (engl. *computer based tests* CBT).

Pored dostupnosti računara, izuzetan napredak CBT prouzrokovan je brojnim prednostima koje ima u odnosu na konvencionalne testove, koji se često u literaturi nazivaju papir i olovka testovi (engl. *paper-and-pencil test*). Najčešće pominjana prednost je brža administracija testa i ocenjivanje, kao i mogućnost dobijanja više informacija o ispitaniku.

Testiranje primenom računara omogućava bolju kontrolu uslova testiranja. Test se zadaje na isti način za sve ispitanike i svi dobijaju iste instrukcije. Ukoliko vreme predviđeno za test istekne, računar zaustavlja proces testiranja. Identični uslovi isporuke testa i aplikacije za testiranje svih ispitanika eliminiše greške merenja koje mogu nastati usled neispunjenosti ovih uslova, te na taj način povećava pouzdanost i validnost testova.

Kao jedna od prednosti CBT može se navesti i povećana fleksibilnost (mogućnost dodavanja poruka koje pružaju dodatna objašnjenja, uputstva i podsetnika, mogućnost da se isprave greške), veća motivacija i zadovoljstvo ispitanika u poređenju sa onim ispitanicima koji rešavaju klasičan test [20]. Testiranje primenom računara je

ekonomičnije, omogućava bolje iskorišćenje ljudskih resursa, a multimedijalne mogućnosti računara nude brojne prednosti u načinu predstavljanja pitanja (audio, video...).

Naravno, CBT imaju i neke nedostatke, u prvom redu potrebna je određena računarska pismenost ispitanika kako bi se izbegao uticaj načina isporuke testova na postignute rezultate [5]. Sledeća bitna osobina ispitanika koja može da utiče na njegov rezultat primenom CBT je stav koji ispitanik ima prema računaru.

Neki ljudi nisu upoznati sa tehnologijom i ne mogu da drže korak sa njenim brzim razvojem, te stoga ne vole da se bave ili koriste bilo koji oblik tehnologije, niti da ga primenjuju u svom akademskom obrazovanju, ili društvenom životu [3]. Značajan nedostatak testiranja putem računara je što se ispitanik ne može vratiti na prethodno pitanje (ne može se preći na sledeće pitanje dok se ne odgovori na tekuće pitanje), lakše se pogreši pri čitanju sa monitora računara nego prilikom čitanja teksta napisanog na papiru, a stalno skretanje pogleda sa monitora računara na papir s beleškama može da ometa ispitanika u radu [169].

### **3.2. Računarski adaptivni testovi**

Tokom rešavanja klasičnog testa, ispitanici se mogu osećati obeshrabreno ako su pitanja suviše teška, ili, s druge strane, možda izgubiti interes, ako su pitanja suviše laka za njihov nivo znanja. Rešenje ovog problema može biti primena računarskih adaptivnih testova (engl. *computerized adaptive test*, CAT) koji kao i kod kvalitetnog usmenog ocenjivanja imaju mogućnost da promene nivo težine pitanja prema mogućnostima ispitanika.

Računarski adaptivni test daje mogućnost pravljenja testova prema sposobnostima ispitanika. Pitanja se biraju iz baze pitanja uzimajući u obzir pojedinačne sposobnosti kandidata tokom testa, ili drugim rečima, kako je svaki ispitanik odgovorio na prethodna pitanja. Ako ispitanik odgovori tačno na postavljeno pitanje, sledeće pitanje koje dobija je za jedan nivo veće težine od prethodnog. S druge strane, ako ispitanik pogrešno odgovori, pitanje koje mu se daje kao sledeće je lakše. Na ovaj način, ispitanicima se daju pitanja prilagođena njihovim sposobnostima. Postignuti bodovi zavise od nivoa težine pitanja, te se ispitanici sa istim postotkom tačnih odgovora mogu znatno razlikovati u krajnjem rezultatu testa u zavisnosti od težine pitanja na koja su tačno odgovorili.

Ideja prilagođavanja testa ispitanicima javila se sa samim počecima psiholoških merenja, a potekla je od Alfreda Bineta koji je razvio Binetov IQ test [17] koji je kasnije objavljen kao Stanford-Binetov IQ test (engl. *Stanford-Binet IQ Test*). Binetov adaptivni pristup merenja sposobnosti ostao je jedini operativni adaptivni test više od pola veka. Tokom 50-tih godina dvadesetog veka u američkoj vojsci vršena su istraživanja o mogućnostima primene adaptivnog testiranja na klasičan način (papir i olovka) i

primenom mašina za testiranje [14], [15]. Međutim, oba ova pristupa bila su neuspešna.

Uvođenjem računara u proces testiranja povećala se fleksibilnost upotrebe testova. Računarski adaptivni testovi su redizajnirani psihološki i edukacioni testovi za efikasno administriranje pomoću računara [141], [146]. Zadatak računara je da odabere za svakog ispitanika grupu pitanja koja najefikasnije meri njegove sposobnosti. Na početku razvoja CAT bili su zasnovani na varijacijama Binetovog pristupa [157] ili pristupima baziranim na različitim strukturiranjima baze pitanja (engl. *item bank*) [72], [73]. Međutim, ubrzo se pokazalo da svaki od ovih pristupa ima određena ograničenja i da nastali problemi mogu da se reše primenom teorije odgovora na stavke (engl. *Item Response Theory* – IRT) što je i primenjeno tokom 70-tih godina.

Računarsko adaptivno testiranje nudi niz prednosti: povećana sigurnost testiranja, efikasnost i preciznost merenja. Sigurnost testa postiže se na osnovu toga što se za svakog ispitanika test prilagođava prema njegovim mogućnostima što obezbeđuje da ispitanici ne dobiju isti set pitanja, čime se mogućnost prepisivanja svodi na minimum [164]. Pošto je manje odgovora potrebno za postizanje statistički prihvatljivog nivoa tačnosti, znatno manje vremena je potrebno za izvođenje računarskih adaptivnih testova u odnosu na računarske testove fiksne dužine. Adaptivni testovi vreme potrebno za proveru znanja smanjuju od 30% do 50% u odnosu na konvencionalne testove iste preciznosti [78], [122]. Takođe, adaptivni testovi daju bolji kvalitet merenja u odnosu na klasične testove za željeni nivo sposobnosti [165]. U radu [13] takođe se analiziraju prednosti računarskih adaptivnih testova.

Kod računarski adaptivnih testova ispitanicima najčešće nije dozvoljeno da se vrate na neko od prethodnih pitanja i da promene već jednom dat odgovor. U radu [75] prikazuju se rezultati istraživanja, gde su studenti jedne grupe bili u mogućnosti da izmene prethodno dat odgovor u računarski adaptivnom testu, a drugi nisu. Rezultati pokazuju da ova mogućnost nije dovela do značajnije razlike u postignućima dve grupe studenata. Upkos malog efekta modifikacije rezultata, smatra se da ova opcija može dovesti do smanjenja anksioznosti studenata kao i povećanja njihovog poverenja u ovakav vid provere znanja.

U radu [2] opisan je adaptivni sistem za testiranje koji se bazira na IRT i eksperimentisanje sa IRT u cilju procene primenjivosti i prednosti koje pruža. Procena parametara pitanja rađena je metodom najmanjeg kvadrata i pokazano je da je moguće smanjiti broj pitanja u adaptivnom testu kako bi se postigao željeni procenat tačnosti procene znanja studenata modifikacijom jednačine na osnovu koje se izračunava stepen sposobnosti. Kako bi se IRT učinio realističniji i primenljiv, uvedeni su novi faktori u IRT: početni nivo sposobnosti studenata, težina predmeta, broj vežbi i časova predavanja koje je predavač pokrio tokom nastave. Rezultati konvencionalnog ispita upoređeni su sa adaptivnim testom i primenjeni su za određivanje IRT parametara. Stavovi studenata prema adaptivnom testiranju i njihovi rezultati dobijeni su na osnovu upitnika. Rezultati su bili pozitivni po pitanju stavova studenta o



adaptivnom testiranju, kao i minimizacije broja potrebnih pitanja za postizanje određene tačnosti procene znanja.

U radu [149] autor sugeriše da CAT imitira način usmenog intervjua u kojem nastavnik prilagođava odabir pitanja na odgovarajući nivo znanja pojedinih učenika. CAT se može opisati i kao poseban slučaj inteligentnog korisničkog interfejsa, u kojem se nenametljivo prati dostignuće korisnika i nivo težine pitanja prilagođava shodno tome [65]. Računarski adaptivni testovi navode se kao jedan od elemenata paradigme nastale u okviru razvoja obrazovnog softvera od oblika gde jedan sistem učenja odgovara svima („one-size-fits-all“) u oblik koji nudi viši nivo personalizacije i interakcije [25].

Međutim, iako su ovo nesumnjive prednosti računarskih adaptivnih testova, i u opšte testova koji se rade na računaru, u radu [27] autori su se osvrnuli i na nedostatke ovakve provere znanja. U prvom redu pominje se cena tehnologije koja je neophodna za realizaciju testiranja, neodgovarajuće funkcionisanje tehnologije, potreba osposobljavanja administratora kao i veština studenata u radu s računarom.

Klasični testovi provere znanja imaju određene nedostatke, kao što su na primer bodovanje i brzina dobijanja povratne informacije o postignutom rezultatu. U zavisnosti od broja ispitanika, predavačima je potrebno znatno vremena da bi pregledali sve radove, što znači da ispitanici ne mogu odmah nakon završetka testa da dobiju informaciju o svom rezultatu. Iz psiholoških razloga veoma je važno da se informacija o uspehu na testu dobije odmah nakon završetka testa, jer predstavlja motivacioni faktor, pomaže studentima da se fokusiraju na učenje i ukazuje im na to da li je potrebno da ulože više truda u proces učenja. Jedan od načina da se prevaziđe ovaj problem jeste primena računarskih adaptivnih testova. Prednosti računarskih adaptivnih testova uključuju testove koji vremenski kraće traju, fleksibilan raspored testiranja, povećanu sigurnost testiranja, bolju kontrolu izlaganja pitanja (engl. *item exposure*), bolje balansiranje sadržaja oblasti čije se znanje proverava za sve nivoe sposobnosti, brže izveštavanje o rezultatima i bolje iskustvo ispitanika o samom procesu testiranja.

Kada se konvencionalan test isporučuje na računaru, mogu se primeniti dva načina digitalizacije. Jedan je linearan računarski test (CBT) koji zapravo predstavlja samo računarsku verziju papirnog testa. U linearnom CBT, pitanja, način skorovanja i procedure su iste kao i kod konvencionalnog testa. Razlika je jedino u načinu isporuke testa – papir i olovka, ili računar. Drugi tip je računarski adaptivni test gde ne samo da je promenjen medijum administracije već i algoritam isporuke testa koji nije više linearan nego adaptivan. Stoga, u razmatranju uporedivosti rezultata testova rađenih na klasičan način i primenom računarskog adaptivnog testa, mora se uzeti u obzir način administracije kao i efekat paradigme kako bi se obezbedila komparabilnost ova dva testa.

U [144] poredi se CBT i CAT verzija testa koji proverava znanje reči (engl. *vocabulary test*) u smislu određivanja preciznosti i tačnosti. Rezultati su pokazali da je CAT precizniji u proceni sposobnosti, naročito kada je u pitanju CAT fiksne dužine a zahteva i manji broj pitanja kako bi se postigla preciznost jednaka onoj kod CBT. Do sličnih rezultata došli su i autori u [145] kada su poredili dve verzije testa za određivanje muzičkih sposobnosti. U meta analizi studije veština čitanja i znanja matematike [153], [152] gde su ispitanici bili učenici osnovnih i srednjih škola, pokazalo se da CBT ili CAT imaju značajnu prednost u procenivanju sposobnosti u odnosu na konvencionalan test. Rezultati ovih istraživanja sugerišu da je paradigma testa jedan od faktora koji utiče na nekomparabilnosti CAT i klasičnog testa.

Tri glavna kriterijuma za određivanje komparabilnosti između CAT i konvencionalnog testa sumirana su u [154]: (1) valjanost, (2) psihometrijske i (3) statističke pretpostavke/način administracije testa. Ovi kriterijumi primenjivi su i za utvrđivanje komparabilnosti lineranog računarskog testa i konvencionalnog. Za CAT procedura evaluacije se dodatno komplikuje zbog načina administracije CAT, kao što je na primer procena parametara stavke, selekcija stavke, skorovanje testa i pravilo zaustavljanja.

U kontekstu valjanosti, najčešće se u istraživanjima pominju pol i etnička pripadnost [126]. Psihometrijski kriterijum odnosi se na poređenje oblika raspodele rezultata, pouzdanosti i standardne greške merenja [154]. U literaturi koja se bavi poređenjem koji efekat ima način administracije testa, najčešće je to CBT i konvencionalan test, psihometrijske karakteristike ispituju se na nivou testa i na nivou stavke. Na nivou testa, najvažniji kriterijum je razlika srednjih vrednosti a na nivou stavke (itema) analiza sadržaja, razlika srednjih vrednosti, razlika IRT-a i distribucija odgovora. Komparabilnost alternativnih verzija testa ne može se uzeti zdravo za gotovo; ispitivanja koja se rade u vezi s tim moraju se uraditi tako da se obezbedi da ispitanici ne budu oštećeni zbog načina administracije testa.

Razvoj računarskog adaptivnog testa zahteva psihometrijsku ekspertizu, koja u nekim slučajevima nije moguća. U radu [134] razmatraju se pitanja relevantna za svaku fazu razvoja i daju se smernice o kojim je potrebno voditi računa. CAT razvijen bez adekvatnih istraživanja i dokumentacije u svakoj od faza razvoja dolazi u opasnost da bude neefikasan. Na primer, proizvoljno postavljanje uslova za zaustavljanje testa ili definisanje maksimalnog broja stavki, bez empirijskog dokaza zašto su ti kriterijumi postavljeni, može dovesti do pogrešnog skorovanja rezultata testa. Autori daju širok okvir za razvoj CAT primenjiv u najvećem broju slučajeva. Međutim, kako se naglašava, ova diskusija ni u kom slučaju nije sveobuhvatna jer je svaki slučaj razvoja aplikacije drugačiji i jedinstven.

U radu [63] razmatra se proces koji prati razvoj računarskog adaptivnog testa rađenog za Singapursku agenciju za razvoj radne snage (engl. *Singapore Workforce Development Agency*) koji proverava znanje engleskog jezika i elementarne matematike prema potrebama radnog mesta za koje radnik konkuriše. Razmatra se

stvaranje i selekcija pitanja, kalibracija baze pitanja, analiza i testiranje psihometrijskih karakteristika, prikazivanje rezultata i njihovo tumačenje.

### 3.2.1. Računarski adaptivni testovi realizovani u praksi

Računarski adaptivni testovi imaju široki spektar primene ne samo u obrazovanju, nego na primer i u medicini [32], [40], [50], [136], [155], ili za određivanje profila kupca u cilju definisanja optimalnog konfiguratora proizvoda prema karakteristikama kupca [46]. Takođe, primenu su našli i za merenje kognitivnih sposobnosti, znanja i osobina ličnosti prilikom testiranja za zaposlenje [43], [49], [66]. U obrazovanju, adaptivni testovi primenjuju se u mnogim programima za sticanje sertifikata za različite profesije, kao na primer dobijanje sertifikata za računovođe u SAD (engl. *American Institute of Certified Public Account*) [167], medicinske tehničare (engl. *American Society of Clinical Pathologists Board of Registry Certification Examinations*) [166] i test za određivanje kvalifikacija za vršenje službe u oružanim snagama SAD (engl. *Armed Services Vocational Aptitude Test Battery*) [172].

Služba za obrazovno ispitivanje (engl. *Educational Testing Service*), najveća svetska privatna neprofitna organizacija za testiranje u oblasti obrazovanja, od 1993. godine uvela je GRE (engl. *Graduate Record Exam*) test kao adaptivni test. GRE je standardizovan test čije je uspešno polaganje neophodno za mnoge visoko obrazovne institucije u SAD, kao i u drugim zemljama s engleskog govornog područja. Ista organizacija razvija i sprovodi i TOEFL (engl. *Test Of English as a Foreign Language*, „test engleskog kao stranog jezika”), za ispitivanje znanja stranih studenata koji žele da slušaju nastavu na koledžima i univerzitetima SAD. Većina institucija visokog školstva s engleskog govornog područja priznaje rezultat na TOEFL testu kao meru studentovih govornih veština, i u slučaju dovoljno visokog skora, zadovoljavanje kriterijuma za pohađanje nastave. TOEFL postoji i u verziji računarskog adaptivnog testa [171].

Grupa za istraživanje računarskog adaptivnog testiranja sa madridskog univerziteta (*Autonoma University of Madrid*), razvila je adaptivni test engleskog jezika za one ispitanike kojima je španski maternji jezik [108]. Univerzitet Midlseks u Velikoj Britaniji (engl. *Middlesex University Kingdom*) razvio je adaptivni test engleskog jezika za studente koji žele sa studiraju u Velikoj Britaniji, a engleski im nije maternji jezik [168]. Cito, Nacionalni institut za pedagoška merenja u Holandiji (engl. *National Institute for Educational Measurement in The Netherlands*) primenjuje računarske adaptivne testove za proveravanje znanja matematike u obrazovanju odraslih i u srednje stručnom obrazovanju, zatim testove čitanja za holandski kao prvi jezik, testove čitanja i razumevanja holandskog kao drugog jezika, proveravanje znanja iz geografije [39]. Razvijeni su adaptivni testovi i za uzrast od 4 do 6 godina, koji uključuju poznavanje osnovnih aritmetičkih veština, osnovne govorne veštine kao i orijentacija u prostoru i vremenu.

J-CAT (engl. *Japanese Computerized Adaptive Test*) je računarski adaptivni test koji može da se rešava *on line*, a koristi se za određivanja nivoa znanja japanskog jezika kao

drugog/stranog jezika. Čine ga četiri dela: razumevanje, gramatika, čitanje i fond reči [180].

Majkrosoft je razvio adaptivne testove za polaganje nekih od ispita iz njihovog sertifikacijskog programa i time sticanja odgovarajućeg zvanja IT profesionalca [183].

Iskustva u primeni CAT u Izraelu u svrhu provere znanja na prijemnom ispitu opisana su u radu [47]. Opisana su dva testa. Prvi test je AMIRAM, računarski adaptivni test engleskog jezika kao stranog jezika koji primenjuje više visoko obrazovnih institucije u Izraelu već 22 godine. Drugi je MIFAM, računarski podržana adaptivna verzija psihometrijskog testa koji je u upotrebi 9 godina, kao prijemni ispit visoko obrazovnih institucija za osobe sa invaliditetom. Obe aplikacije rade paralelno sa konvencionalnom verzijom testova. Rad se fokusira na specifične tehnike koje se koriste kako bi se obezbedilo pravično ocenjivanje u dva medija, kao i na ispitivanje pogodnosti primene računarskih adaptivnih testova za osobe sa invaliditetom. Do sada je 5100 kandidata testirano pomoći MIFAMA. U upitniku koji popunjavaju nakon testa, ispitanici sa invaliditetom kažu da su zadovoljni sa CAT, da je jasan i jednostavan za upotrebu (engl. *user friendly*). Takođe, ocenjuju da je test „darežljiv“ u vremenu koji stoji na raspolaganju za rešavanje pitanja, s obzirom da neki od njih imaju teškoće pri koncentraciji i sporo čitaju. Test ocenju kao „fer“. Primena MIFAMA uklonila je problem posebnog obeležavanja rezultata pojedinih studenata (sa nekim vidom invaliditeta) jer ne postoji više grupa studenata koja je rešavala isti test kao i drugi studenti, ali pod nestandardnim uslovima.

SIETTE sistem je jedna komponenta projekta TREE (engl. *TRaining of European Environmental Trainers and Technicians*) finansiranog od strane programa Leonardo da Vinči (engl. *Leonardo da Vinci Program*) Evropske unije. SIETTE sistem omogućava nastavnicima da *online* edituju testove, a mogućnost uključivanja multimedijalnih elemenata i apleta u pitanja (npr. prepoznavanje objekata sa fotografije, interakcija) obezbeđuje definisanje pitanja koja se teško mogu realizovati klasičnim testom [30]. Neformalna evaluacija sistema rađena je na Departmanu za računarske nauke Univerziteta Malaga, Španija (engl. *University of Málaga, Spain*) sa studentima dobrovoljcima koji su slušali kurs Veštačka inteligencija. Test je sadržao pitanja koja se tiču poznavanja jezika LISP, kao i upitnik o mišljenju studenata o samom sistemu SIETTE. U istraživanju je učestvovalo 24 studenta, većina njih (87.5%) ovakav način polaganja preporučio bi i kolegama, a 8.33% ne bi. Stavovi studenata većinom su pozitivni; kao pozitivne elemente ističu jednostavnost i dizajn interfejsa, inteligentno (adaptivno) generisanje pitanja, pristup preko weba i grafički prikaz rezultata testa.

U radu[29] opisan je web baziran adaptivni sistem za procenu znanja programiranja (PAT, *Programming Adaptive Testing*). Primenjen je u dve grčke srednje škole za proveru znanja osnovnih elemenata programiranja i pseudo-jezika Glossa. Rezultati na testu korišćeni su i za predviđanje uspeha učenika na nacionalnom ispitu Panhelenik (*Panhellenic*). U planu je razvoj sistema kako bi se podržalo testiranje znanja i drugih programskih jezika (Java, Visual Basic).

Autori u radu [76] opisuju dizajn, razvoj i primenu računarskog adaptivnog testa za proveru znanja *Visual Basica* na Univerzitetu Hertfordšajar (engl. *the University of Hertfordshire*). Prema rezultatima istraživanja u kojem je učestvovalo 133 studenta, studenti koji su polagali test na ovaj način nisu bili u nepovoljnijem položaju od onih koji su polagali na klasičan način. Rezultati koje su studenti postigli na testu verifikovani su i dodatnim proverama (usmeni deo ispita).

Primenom aplikacije KAPI [8] koja omogućava adaptivno testiranje, sproveden je eksperiment na Fakultetu organizacionih nauka u Beogradu, kojim je obuhvaćeno polaganje usmenog dela ispita Sistemi za podršku odlučivanju. U eksperimentu je učestvovalo 36 ispitanika. Obrada rezultata ankete pokazala je da sami studenti, usled svog svakodnevnog rada na računaru, pokazuju visok nivo spremnosti ka ocenjivanju znanja preko računara. Samo učešće u eksperimentu pokazuje njihovu želju za praćenjem novih, savremenih trendova, kako su se i u anketi izjasnili. Kao osnovne prednosti ovakvog načina ispitivanja studenti ističu: smanjenje vremena (nema pisanja), povećanu preglednost, nezavisnost u odnosu na okruženje, smanjenje mogućnosti greške pri obradi rezultata, savremeni način polaganja ispita, ujednačen kriterijum ocenjivanja, smanjenje treme, zanimljivije je, lakši način, mogućnost studenata koji su bolesni da polažu od kuće, savremeniji način, mogućnost trenutnog uvida u rezultate, povećanje motivacije, isključivanje subjektivnosti ispitivača, očekivani izgled pitanja, više vremena za razmišljanje, smanjenje zabune jer je prezentovano samo jedno pitanje i sva koncentracija je usmerena na njega kao i polaganje u malim grupama.

Na fakultetu za industrijski menadžment u Kruševcu [7], na uzorku od 100 studenata primenjen je računarski adaptivni test. Web aplikacija realizovana je u Visual Studio.NET okruženju u programskom jeziku C#. Studenti su testirani na konvencionalan način i primenom računarskog adaptivnog testa. Na osnovu rezultata ankete, 82% ispitanika je izjavilo da im konvencionalan način provere znanja ne odgovara. Kao jedan od razloga studenti su naveli da pri proveru znanja primenom klasičnog testa, često se dešava da za isti odgovor studenti budu različito ocenjeni (25% ispitanika) i da test nije prilagodljiv prema studentu (35%). Primetno je da i slabiji i bolji studenti gube koncentraciju i motivaciju za daljnim rešavanjem testa, ukoliko su pitanja neodgovarajuća za njihov nivo znanja (suviše laka, ili suviše teška). Pri rešavanju adaptivnog testa, studentima je bilo ponuđeno da odaberu početni nivo težine pitanja od tri moguća. Studenti su imali osećaj da je test „pravljen baš za njih“, jer se težina pitanja prilagođavala njihovom nivou znanja, pa su pitanja koja su se generisala bila „blizu“ njihovog nivoa znanja, ni suviše ispod, ni suviše iznad. Od ukupnog broja studenata koji su rešavali adaptivni test, 74% smatra da primenjeni sistem adaptivnog testiranja realno oslikava njihovo znanje. Studenti koji su izjavili da im se ne sviđa adaptivno testiranje, uglavnom su pripadali grupi onih ispitanika koji retko koriste računar (stariji studenti, od 30 do 50 godina).

U radu [95] opisana je primena jednog veb – baziranog sistema za testiranje studenata. Prikazan sistem ne obuhvata sve vidove procene znanja i veština uz upotrebu IT-a.

Pored klasičnog, računarski podržanog testiranja, uvedena je i mogućnost adaptivnog testiranja studenata koja se ograničava na jednostavnu procenu znanja testom, koji se prilagođava trenutnom znanju ispitanika. U svrhu istraživanja definisana je jedna varijanta eksperimentalnog istraživanja sa paralelnim grupama. U eksperimentu je učestvovalo 30 studenata podeljenih u kontrolnu i eksperimentalnu grupu, koji su u akademskoj godini 2010/11 bili na trećoj godini studija na Visokoj poslovnoj školi strukovnih studija iz Blaca. Testovi provere znanja pokazali su da su izabrane grupe ekvivalentne i da je eksperimentalna grupa (adaptivni test) pokazala bolje rezultate na prvom i završnom testu od kontrolne grupe, ali se na drugom testu kontrolna grupa pokazala boljom. Autor rada smatra da je razlog toga nedovoljna stručna definisanost pitanja (bez konsultovanja eksperata) kao i definisanje kompleksnosti pitanja. Rezultati pokazuju da adaptivno testiranje nije uticalo na promenu vrednosti merene veličine (skor na testu znanja). Zaključak rada je da računarski podržan adaptivni test jednako dobro meri ovu veličinu kao i konvencionalan test.

## 4. Teorijske osnove računarskih adaptivnih testova

Računarsko adaptivno testiranje (engl. *Computerised Adaptive Testing*) je vid testiranja razvijen sa ciljem da se poveća efikasnost procene ispitanikovog znanja. To se postiže tako što se pitanja, tokom celog testa, prilagođavaju ispitaniku na osnovu njegovih prethodnih odgovora, zbog čega se često naziva i „krojeni test“ (engl. *tailored testing*). Težina svakog sledećeg pitanja se bira tako da novo pitanje ne bude ni previše teško, ali ni previše lako za ispitanika. Tačnije, bira se pitanje za koje se procenjuje, sa verovatnoćom od 50%, da će ispitanik tačno odgovoriti. Naravno, prvo pitanje se ne može izabrati na taj način pošto se u početnom trenutku još ništa ne zna o ispitanikovim sposobnostima (nego se postavlja pitanje srednje težine), ali se već izbor drugog pitanja može bolje prilagoditi ispitaniku i sa svakim narednim odgovorenim pitanjem, računar je u mogućnosti da preciznije proceni njegovo znanje.

Prednosti ovakvog načina testiranja su [158]:

- Ispitanik ne može znati koja pitanja će dobiti na testu, zato što izbor sledećeg pitanja zavisi od njegovog odgovora.
- Smanjen je broj lakih pitanja koja se postavljaju ispitanicima sa veoma dobrim znanjem (pa je onda moguće tačnije odrediti u kojoj meri je kvalitetno znanje takvih ispitanika).
- Takođe je i smanjen broj teških pitanja koja se postavljaju ispitanicima sa malim znanjem (jer postoji verovatnoća da su pogađali moguć odgovor, što iskrivljuje sliku o stvarnom znanju).
- Primenom adaptivnog testiranja za procenu nečijeg znanja/sposobnosti potreban je mnogo manji broj pitanja nego u slučaju konvencionalnih testova, a to za sobom povlači i manje vremena koje je potrebno za utvrđivanje znanja ispitanika.
- Rezultati su dostupni odmah po završetku testa.
- Ispitanik se na svakom pitanju zadržava onoliko vremena koliko mu je potrebno.

Naravno, adaptivni testovi imaju i nedostatke:

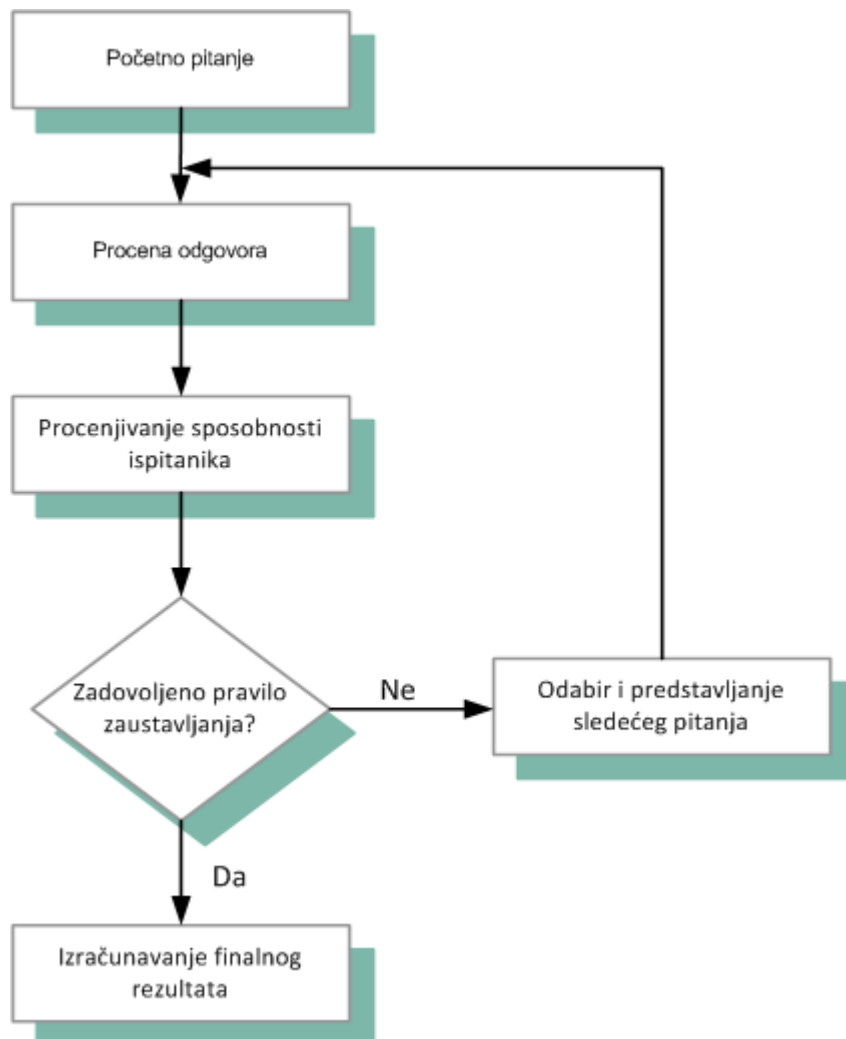
- Računarski adaptivni test nije moguće primeniti za svako testiranje, npr. kada treba prikazati veliku količinu teksta koja bi se trebalo pročitati da bi se odgovorilo na pitanje, ili kada treba proveriti znanje u oblastima koje zahtevaju posedovanje određenog talenta (npr. prijemni ispit za muzičku školu, likovnu akademiju i slično).
- Pri sastavljanju testa, potrebno je veoma pažljivo kalibrirati pitanja.
- Za testiranje velikog broja ispitanika, potrebno je imati i dovoljan broj računara.

- Nije moguće zameniti pitanje ili se vratiti jedno pitanje unazad, jer računarski adaptivni testovi uglavnom ne dozvoljavaju izmenu prethodno datog odgovora.

Koraci koji se sprovode pri svakom postavljenom pitanju pri računarskom adaptivnom testiranju su:

1. Od svih pitanja, bira se ono koje je najpodobnije za procenu sposobnosti ispitanika.
2. Izabrano pitanje se prikazuje i ispitanik na njega daje odgovor.
3. Nakon datog odgovora, sposobnost ispitanika se nanovo procenjuje (imajući u vidu sve prethodne odgovore).
4. Sledi ponavljanje koraka 1-3 sve do kraja testa (koji je određen time što je ispitanik prešao sva pitanja, ili je vreme za izradu testa isteklo ili je kraj testa određen nekim drugim kriterijumom).

Slika 4.1 služi kao ilustracija algoritma računarskog adaptivnog testa.



Slika 4.1: Algoritam računarskog adaptivnog testa



#### 4.1. Binetov IQ test

Osnovni princip prilagođavanja testa svakom pojedincu prepoznao je još u ranim danima razvoja psiholoških merenja, čak i pre razvoja standardizovanih konvencionalnih papir i olovka testiranja, Alfred Binet koji je razvio Binetov IQ test [17]. Ovaj test je kasnije nazvan Stanford-Binetov IQ test. Binetov test se sastojao od više grupa testova normiranih po hronološkom starosnom nivou.

Glavna karakteristika Binetovog testa je kompletno prilagodljiva procedura, kao što se može videti iz sledećeg [158]:

1. Koriste se prekalibrisane baze pitanja. Binet selektuje pitanje za svaku starosnu grupu, ako je otprilike 50% dece istog uzrasta odgovorilo tačno na zadato pitanje. Stoga, u originalnoj verziji testa postoje setovi pitanja za uzraste dece od treće do jedanaeste godine života. Sve ove stavke sačinjavaju delove grupe Binetovog adaptivnog testa.
2. To je individualno administriran test od strane stručnog psihologa i dizajniran da „ispita“ nivoe težine (npr. po godini uzrasta) testa, kako bi se ustanovilo šta je najprikladnije za svakog pojedinca, isto kao što je za atletičara koji preskače prepone, visina i razdaljina istih, zavisna od uzrasta.
3. Test ima promenljive početne opcije. Administrator testa započinje Binetov test na osnovu najbolje procene o nivou sposobnosti ispitanika kojom raspolaže. Obično je to uzrast ispitanika, ukoliko ne postoje neke dodatne informacije koje mogu da promene početni nivo težine pitanja na višu ili nižu vrednost.
4. Koristi se definisani metod bodovanja – skup pitanja u datom uzrastu se odmah evidentira i ocenjuje od strane administratora testa.
5. Postoji „grananje“, ili pravilo odabira pitanja, koje određuje koje će sledeće pitanje administrator uputiti ispitaniku. U Binetovom testu naredni set pitanja zavisi od odgovora koje je ispitanik dao u predhodnim setovima pitanja. Ako je ispitanik odgovorio tačno na neka, ili na većinu pitanja za njegov uzrast, onda se u sledećem setu pitanja podižu za jedan nivo (viša starosna grupa). Ako je većina odgovora netačna, sledeći set pitanja će biti za niži nivo (niža starosna grupa).
6. Postoji unapred definisano pravilo završetka testa. Binetov test se završava kada se odredi „donja“ i „gornja“ granica ispitanika. Gornja granica (po godinama starosti) bi bila ona na kojoj je ispitanik netačno odgovorio na sva pitanja a osnovna ona na kojoj je ispitanik dao sve tačne odgovore. Efektivni opseg merenja za svakog ispitanika je između ova dva nivoa.

Konačni rezultat ispitanika na Binetovom testu zavisi od podskupa pitanja na koja je odgovorio tačno. U stvari, ova pitanja su ponderisana prema starosnom uzrastu ispitanika i IQ bodovima dobijenim iz testa, s obzirom da će svaki ispitanik odgovoriti na različiti broj pitanja i različit podskup pitanja.

#### 4.1.1. Šema administracije Binetovog testa

Slika 4.2 ilustruje administraciju Binetovog testa. Pitanja testa su raspodeljena po nivoima koji su na slici obeleženi kao „psihološki uzrast“ (engl. *mental age*), koji uključuju pitanja na koje je tačno odgovorilo približno 50% ispitanika za datu starosnu grupu.

Mental Age	Items	Adaptive Branching	Number Administered	Proportion Correct
10.5			—	—
<b>Ceiling Level</b> → 10	51- 52- 53- 54- 55- 56- 57- 58- 59- 60-		10	0.00
9.5	41+ 42+ 43+ 44- 45- 46+ 47- 48- 49- 50-		10	.40
<b>Starting Level</b> → 9	1+ 2+ 3- 4+ 5+ 6+ 7- 8- 9- 10+		10	.60
8.5	11+ 12- 13+ 14+ 15+ 16- 17+ 18+ 19+ 20+		10	.80
8	21+ 22+ 23+ 24+ 25+ 26+ 27+ 28- 29+ 30+		10	.90
<b>Basal Level</b> → 7.5	31+ 32+ 33+ 34+ 35+ 36+ 37+ 38+ 39+ 40+		10	1.00
7			—	—
6.5			—	—
<b>Total</b>			<b>60</b>	<b>.617</b>

Slika 4.2: Šema administracije Binetovog testa [38]

U ovom primeru test počinje pitanjima za uzrast od 9 godina. Ispitanik je odgovorio tačno na pitanja 1, 2, 4, 5, 6, 8 i 10 a na pitanja 3, 7, 8 i 9 netačno. Od postavljenih 10 pitanja na 60% je odgovoreno tačno. Pošto je na neka pitanja ispitanik odgovorio tačno, a na neka ne, uzrast od 9 godina nije ni donja granica (100% tačno), ni gornja granica (0% tačno), pa se test nastavlja.

Ispitivač može da ide na niži, ili na viši starosni nivo da bi utvrdio gornju ili donju granicu. U primeru koji se opisuje, ispitivač je odlučio da pronađe prvo donju granicu, verovatno da bi ispitaniku pružio malo samopouzdanja s obzirom da je na lakša pitanje lakše odgovoriti. Zato je test „razgranat“ na pitanja za nivo uzrasta od 8.5 godina. Na 80% odabranih pitanja je tačno odgovoreno. Ispitivač sada nastavlja da traži donju granicu tako što počinje da postavlja pitanja za uzrast od 8 godina koja su lakša i daju rezultat od 90% tačno odgovorenih. Konačno grananje na nivou prema uzrastu 7.5 godina daje 100% tačnih odgovora i utvrđuje donju granicu ispitanika.

Nakon indentifikacije donje granice ispitivač nastavlja test u potrazi za gornjom granicom. S obzirom da su sva pitanja od uzrasta 7.5 do uzrasta 9 godina obrađena, test se grana ka pitanjima za uzrast 9.5 godina koji je sledeći nivo težine pitanja koja još nisu administrirana. Za ovu grupu pitanja 40% odgovora na postavljena pitanja bilo je tačno. Pošto ovo nije gornja granica (0% tačnih odgovora) test se nastavlja ka sledećem nivou težine pitanja za uzrast 10 godina. Rezultat pokazuje da je na ovom nivou 0% tačnih odgovora te tako predstavlja i gornju granicu ispitanika.

Primer koji je objašnjen ilustruje nekoliko karakteristika Binetovog adaptivnog testa, koje su karakteristične za većinu adaptivnih testova:

1. Startna, ili polazna tačka testa može da varira za svakog ispitanika ponaosob. U ovom slučaju test je mogao započeti u gotovo bilo kom životnom uzrastu ispitanika, na osnovu bilo koje informacije koju je ispitivač mogao da ima o ispitaniku. Da je test započeo na bilo kom nivou uzrasta između 7.5 do 10 godina, ista pitanja bi bila obrađena i rezultat testa bi bio nepromenjen. Da je test startovao van ovog opsega, bila bi postavljena dodatna pitanja (i zbog toga bi test duže trajao) ali rezultat testa bi ostao nepromenjen (jer je u Binetovom testu rezultat zasnovan na broju tačno odgovorenih pitanja za dati starosni uzrast). Na primer, da je test počeo za uzrast od 7 godina, verovatno bi na sva pitanja bio dat tačan odgovor (kako su to vrlo laka pitanja) uspostavljajući tako donju granicu. Slično tome, pitanja za uzrast od 10.5 godina bi dala gornju granicu s obzirom da su ova pitanja teža nego za uzrast od 10 godina.
2. Test se završava kada se dobije dovoljno informacija o ispitaniku da se može odrediti njegov nivo sposobnosti. Kod Binetovog testa, test se prekida u slučaju kada pitanja ne pružaju više informacija o nivou ispitanikovih sposobnosti. Pitanja ispod donje granice bila bi previše laka a pitanja iznad gornje granice bila bi suviše teška za ispitanika. Nijedna od ove dve grupe pitanja ne bi dala nikakve dodatne informacije o ispitanikovim sposobnostima koje su predmet ispitivanja.
3. Broj pitanja koji se postavlja nije isti za svakog ispitanika. Dobro dizajniran adaptivni test nastavlja da postavlja pitanja dok se ne dobije dovoljno informacija o svakom ispitaniku kako bi se mogao odrediti njegov nivo sposobnosti za unapred određen nivo preciznosti. U Binetovom testu, ovaj nivo preciznosti se određuje postavljanjem donje i gornje granice, bez obzira na broj pitanja potrebnih za određivanje granica za svakog ispitanika. Drugi adaptivni testovi koriste neke druge pokazatelje preciznosti.
4. Svaki adaptivni test potencijalno koristi različitu podgrupu pitanja iz baze pitanja. Adaptivni test je dizajniran da upravlja, iz unapred definisane baze pitanja, podgrupom pitanja koja će najbolje odrediti nivo svakog ispitanika. U primeru prikazanom na slici 4.2, set pitanja je iz grupe pitanja za uzrast od 7.5 do 10 godina. Neki drugi ispitanik možda će dobiti pitanja za uzrast od 5 do 7.5 godina, a neki bi mogao dobiti pitanja za uzrast od 8 do 13 godina.
5. Udeo tačno odgovorenih pitanja za svakog ispitanika za pitanja postavljena u adaptivnom testu biće otprilike 0.50, što predstavlja nivo težine pitanja koji

pruža maksimalnu informaciju za svakog ispitanika. U hipotetičkom primeru sa slike 4.2 taj udeo za sva postavljena pitanja je 0.60. Ova osobina adaptivnog testa će težiti da izjednači psihološki „uticaj sredine“ na poimanje težine testa, do mere u kojoj ispitanici vide očigledne teškoće u postavljenim pitanjima. Kao rezultat toga, ispitanici sa slabijim sposobnostima mogu smatrati adaptivni test lakšim u odnosu na konvencionalan test koji su već polagali, zato što u konvencionalnom testu mogu da vide da su na većinu pitanja odgovorili netačno (nizak procenat tačnih odgovora, „težak“ test). Obrnuto, ispitanici većih sposobnosti mogu imati utisak da je adaptivni test teži jer su navikli da na konvencionalnom testu na većinu pitanja odgovore tačno (visok procenat tačnosti, ili „lak“ test).

#### 4.1.2. Računarski administriran test – efikasnija varijanta Binetovog testa

Godine 1973. predložena je varijacija Binetovog testa [157] koja je zahtevala primenu računara za administriranje testa, s namerom da se test učini efikasnijim. Ovaj test nazvan je „slojno prilagodljiv“ (engl. *stratified adaptive*) ili „stradaptivan“ test (engl. *stradaptive test*).

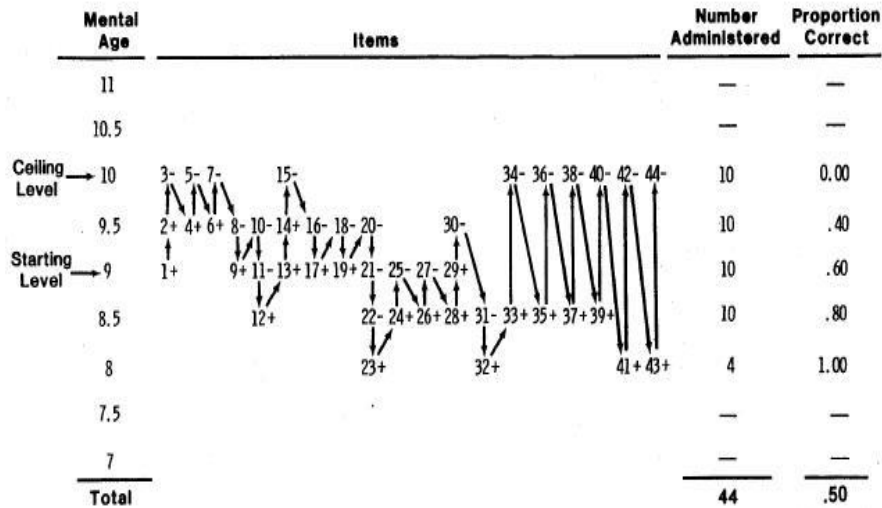
Stradaptivni test koristi istu strukturu baze pitanja kao i Binetov test. To znači da su test pitanja slojevita, ili organizovana po nivoima težine, koji se nazivaju „slojevi“ (engl. *strata*). Slično Binetovom testu, koristi se promenljiv početni nivo koji omogućava da test počne od bilo kog nivoa težine koji se čini prikladnim za svakog ispitanika ponaosob. Stradaptivan test se razlikuje od Binetovog testa u pravilu grananja i pravilu o prekidu testa.

U Binetovom testu, set pitanja za dati nivo težine (*stratum*) je administriran i bodovan. U zavisnosti od rezultata ispitanika za dati set pitanja, ispitivač određuje teži ili lakši nivo sledećeg seta pitanja. U stradaptivnom testu, svako pojedinačno pitanje je administrirano i vrednovano. Odluka o daljnjem grananju donosi se nakon svakog pitanja. Ako je na pitanje odgovoreno tačno, prvo neadministrirano pitanje iz sledećeg težeg sloja – *stratuma* se postavlja. Ako je na pitanje odgovoreno netačno, prvo neadministrirano pitanje sa sledećeg lakšeg sloja – *stratuma* se postavlja. Ovaj proces ocenjivanja pitanja i grananja na jedan nivo težine iznad ili ispod, u zavisnosti od toga da li je na pitanje odgovoreno tačno ili netačno, nastavlja se pitanje po pitanje dok se ne dostigne kriterijum prekida testa.

Binetov test se prekida kada se odrede obe granice, gornja i donja, za svakog ispitanika. Gornja granica je onaj nivo na kome su na sva pitanja dati netačni odgovori; donja granica je nivo na kome je na sva pitanja odgovoreno tačno. Kriterijum prekidanja testa u stradaptivnom načinu testiranja primenjuje varijaciju gornjeg nivoa. Stradaptivan test se može prekinuti kada se na sva pitanja na bilo kom nivou daju netačni odgovori (kao kod Binetovog testa), ili kada se odgovori netačno na pet uzastopno postavljenih pitanja na datom nivou.

#### 4.1.2.1. Primer stradaptivnog testa

Slika 4.3 pokazuje primer zapisa odgovora stradaptivnog testa. Na ovoj slici „+“ predstavlja pitanje na koje je dat tačan odgovor, a „-“ netačno odgovoreno pitanje. Pitanja su grupisana po nivoima (godinama starosti) kao i u Binetovom testu, sa po 10 pitanja na svakom nivou. Dve kolone desno prikazuju broj pitanja postavljenih na svakom nivou starosnog uzrasta i udeo tačnih odgovora za dati nivo.



Slika 4.3: Primer zapisa odgovora stradaptivnog testa [38]

Stradaptivan test dozvoljava promenljiv početni nivo. U primeru testa sa slike 4.3, početni nivo je nivo uzrasta od 9 godina. Prvo pitanje (1) je postavljeno i dat je tačan odgovor (+), pa je kao sledeće pitanje postavljeno prvo neadministrirano pitanje sa nivoa za 9.5 godina. Na ovo pitanje odgovoreno je tačno, pa je kao sledeće pitanje zadato prvo pitanje sa nivoa za 10 godina (pitanje 3). Na ovo pitanje odgovoreno je netačno (3-), stoga je dato prvo sledeće, do tada nepostavljeno pitanje, sa nižeg nivoa (uzrast 9.5 godina) i dat je tačan odgovor (4+).

Proces obrade pitanja, njihovo ocenjivanje i grananje na niži ili viši nivo zavisi od toga da li je na pitanje odgovoreno tačno ili ne, sve do 30-tog pitanja. Na pitanje broj 30 odgovoreno je netačno, ali svih 10 pitanja za uzrast od 9 godina prethodno je već postavljeno, pa je sledilo pitanje broj 6 za uzrast od 8.5 godina. Jedan sloj grananja je zatim nastavio ka pitanjima od 31 do 33. Međutim, kako je svih 10 pitanja na nivou 9 godina postavljeno, treći sloj grananja počinje nakon što je na pitanje 33 odgovoreno tačno. Konačno, nakon što su sva pitanja na nivou 8.5 godina iskorišćena, četvrti sloj grananja je upotrebljen za pitanja od 40 do 44.

Stradaptivni test je prekinut kada je indentifikovana gornja granica. Gornja granica se ustanovljava kada se na sva pitanja na datom nivou da netačan odgovor. Ovo se dogodilo kada se na pitanje 44 dao netačan odgovor (44-), dajući time sve netačne odgovore na nivou starosnog uzrasta od 10 godina.

Kolona koja pokazuje udeo tačnih odgovora prikazuje tipičan rezultat adaptivnog testa. Kao što se i očekuje, udeo tačnih odgovora se povećava (od 0.0 do 1.0) kako nivo težine (godine starosti) opada. Pored toga, ukupan udeo tačnih odgovora je na optimalnom nivou 0.50.

## 4.2. Osnovne komponente računarskih adaptivnih testova

Svaki računarski adaptivni test sastoji se od četiri glavne komponente: baze pitanja, procedure za selekciju pitanja, metode za procenjivanje sposobnosti i pravila zaustavljanja [119]. U nastavku će biti opisane ove četiri komponente.

### 4.2.1. Baza pitanja

Baza pitanja predstavlja skup velikog broja pitanja koja mogu biti postavljena ispitanicima tokom računarskog adaptivnog testa. Konvencionalni testovi (papir i olovka) takođe se dizajniraju na osnovu prethodno stvorene baze pitanja. Razlika je u tome da pitanja odabrana iz baze pitanja su ista za sve ispitanike koji polažu test, dok su kod računarskih adaptivnih testova pitanja individualizovana za svakog pojedinca. Stoga kvalitet baze pitanja ima značajan uticaj na izvršenje adaptivnog algoritma kod računarskih adaptivnih testova [44]. Kvalitet baze pitanja ne zavisi samo od njene veličine, nego i od širine i pokrivenosti sadržaja, dubine pitanja za svaku oblast sadržaja, kao i psihometrijskih osobina pitanja [110].

Neki elementi koji se uzimaju u razmatranje prilikom koncipiranja baze pitanja za računarske adaptivne testove slični su onima kod klasičnih testova. U oba slučaja, pitanja moraju da budu napisana u skladu sa specifikacijom sadržaja, pregledana što se tiče kvaliteta sadržaja i osetljivosti testa, pre-testirana kako bi se njihove psihometrijske karakteristike mogle vrednovati pre nego se uvrste u bazu pitanja [40]. Metode koje se koriste za procenu kvaliteta pitanja kombinuju klasičnu statistiku pitanja kao što je procenat tačnih odgovora i biserijska korelacija sa kriterijumom teorije odgovora na stavke (engl. *Item Response Theory*, IRT) ili teorije odgovora na testlete (engl. *Testlet Response Theory*, TRT) kao što su parametri pitanja ili informacija o pitanju [148].

Sledeća važna karakteristika pitanja koja se ispituje jeste proveravanje da li se svako pitanje uklapa u model merenja koji se primenjuje, tj. da li odgovara IRT ili TRT modelu. Prema [68], jedan od metoda za ocenu da li pitanje odgovara primenjenom metodu jeste analiza regresije pitanje – sposobnost (engl. *analysis of item-ability regressions*). Ovaj grafički metod poredi grafik empirijske raspodele frekvencije pitanja u funkciji sposobnosti ispitanika ( $\theta$ ) sa krivom karakteristike pitanja (engl. *Item characteristic curve*) nastale na osnovu procene parametara pitanja. Ako su dva grafika slična, tada se pitanje dobro uklapa u model i odgovarajuće je za uključivanje u bazu pitanja. Kako se u računarskom adaptivnom testu pitanja biraju tako da odgovaraju procenjenim sposobnostima svakog pojedinca koji polaže test, pri izboru pitanja koja će biti uključena o bazu mora se voditi računa da budu odabrana kvalitetna pitanja za različite nivoe znanja. Nasuprot tome, kod konvencionalnog testa koji se radi na papiru, pitanja

se biraju tako da najbolje odgovaraju ispitaniku prosečnih sposobnosti, tj. onima čiji je nivo znanja blizu centra raspodele sposobnosti [44]. Iz tog razloga, baza pitanja za računarski adaptivni test je veća od baze pitanja kod konvencionalnog testa, kako bi se različite kombinacije pitanja mogle generisati za širok raspon sposobnosti ispitanika [34].

#### 4.2.2. Procedura selekcije pitanja

Kod konvencionalnog testa, pitanja koja će biti postavljena na testu bira konstruktor testa (nastavnik) pre početka samog testa. Prepoznatljivo obeležje računarskih adaptivnih testova jeste da biraju pitanja koja su najprikladnija za svakog ispitanika ponaosob. Kao rezultat toga, pitanja koja će biti postavljena biraju se tokom samog testa kada se na osnovu ispitanikovih odgovora procenjuje nivo njegovih sposobnosti. Da bi se to postiglo, potreban je algoritam koji adaptivno vrši odabir pitanja. Jedan od prvih adaptivnih algoritama za testiranje bio je fleksinivo test (engl. *flexilevel test*) koji je opisao Lord [83]. Test fleksinivo bio je adaptivan test koji nije koristio računar ili neki složen model kao što je IRT ili TRT. Međutim, tokom godina nastali su sofisticirani algoritmi za testiranje koji su razvijeni primenom statističkih procedura i savremenih modela merenja zahvaljujući napretku računarske tehnologije. Dva najčešća algoritma koja se primenjuju [67], [133] su selekcija na osnovu maksimalne informacije (engl. *maximum information selection*) i Bajesova selekcija (engl. *Bayesian selection*).

##### 4.2.2.1. Selekcija na osnovu maksimuma informacije

Jedna od najčešćih procedura za selekciju pitanja je odabir na osnovu maksimuma informacija [18], [84]. Pitanje se selektuje ukoliko obezbeđuje najviše informacija na osnovu trenutno procenjene sposobnosti ispitanika. Proces uključuje primenu procenjene sposobnosti ispitanika na osnovu njihovih odgovora na prethodno postavljena pitanja kako bi se odredila količina informacija koju će obezbediti svako od preostalih pitanja u bazi pitanja. Pitanje koje obezbeđuje najviše informacija bira se kao sledeće pitanje. Obično, ovaj proces se ponavlja nakon administracije svakog pitanja. Početno pitanje za administraciju se obično bira za pretpostavljeni nivo sposobnosti ispitanika na sredini distribucije.

##### 4.2.2.2. Bajesova selekcija

Bajesova selekcija procenjuje svako pitanje u bazi pitanja kako bi se odredila očekivana varijansa kasnije raspodele. Pitanje koje minimizuje očekivanu varijansu bira se za administraciju. Nova raspodela se računa uključujući i prethodno postavljeno pitanje i verovatnoću. Ova prethodna raspodela se računa za svako preostalo pitanje u bazi pitanja kako bi se odredilo pitanje koje minimizuje očekivanu varijansu nove raspodele. Nedostatak Bajesove selekcije je što je računski zahtevnija nego maksimum informacije, kao i činjenica da redosled kojim se pitanja odabiraju za administriranje utiču na procenu sposobnosti  $\hat{\theta}$  [133].

### 4.2.3. Procena sposobnosti

U većini sistema za računarsko adaptivno testiranje, smatra se da su vrednosti parametara u bazi pitanja pre-testirana i kalibrisana pre operative upotrebe. Stoga, jedini parametar koji zahteva procenu tokom izvođenja računarski adaptivnog testa jeste znanje ispitanika ili nivo sposobnosti  $\hat{\theta}$ .

Prvi korak u procesu procene sposobnosti uključuje određivanje početne procene sposobnosti. Početna procena sposobnosti je potrebna kako bi algoritam za testiranje odabrao prvo pitanje ili testlet testa. Jedan od načina da se odredi procena početne sposobnosti jeste da se iskoristi neka prethodno poznata informacija o ispitaniku, kao na primer njegov rezultat na prethodnom testu iz iste oblasti. Ili, jednostavno može da se postavi na srednju vrednost pretpostavljene raspodele, koja bi bila nula ako se pretpostavi da  $\theta$  ima standardnu normalnu raspodelu. Nakon svakog pitanja ili testleta, neophodna je privremena procena za  $\theta$  za CAT algoritam kako bi se odredilo sledeće pitanje ili testlet. Finalna ili krajnja procena sposobnosti određuje se na kraju testa na osnovu skupa svih odgovora ispitanika. Privremena i konačna procena sposobnosti ne moraju se odrediti primenom iste metode [27]. Konačna procena sposobnosti može se transformisati u različite metrike sposobnosti [110]. Dva uobičajena načina za procenu sposobnosti u računarskom adaptivnom testiranju jesu procena maksimalne verovatnoće i Bajesova procena.

#### 4.2.3.1. Procena maksimalne verovatnoće

Najčešće korišćenja procedura za procenu sposobnosti je procena maksimalne verovatnoće (engl. *maximum likelihood estimation*), [85]. Na osnovu odgovora ispitanika na pitanja, procena maksimalne verovatnoće određuje vrednost sposobnosti  $\theta$ , koja maksimizuje verovatnoću šeme odgovora na pitanja  $(x_1, x_2, \dots, x_n)$ . Verovatnoća vrednosti sposobnosti  $\theta$  za datu šemu odgovora  $(x_1, x_2, \dots, x_n)$  data je jednačinom (1):

$$L(x_1, x_2, \dots, x_n | \theta) = \prod_{i=1}^n P_{i_{x_i}}(\theta) \quad (1)$$

gde  $P_i(\theta)$  predstavlja verovatnoću tačnog odgovora na pitanje  $i$ , a  $n$  je broj pitanja na koje je odgovoreno tokom testa.

Zbog diminutivne prirode množenja brojeva između nula i jedan [41], u daljnim izračunavanjima koristi se prirodni logaritam funkcije verovatnoće,  $\ln L(\theta)$ . Uzimanjem logaritma funkcije verovatnoće, sabiraju se logaritmi verovatnoće tačnog odgovora za svako pitanje. Ista vrednost sposobnosti  $\theta$ , daje maksimum funkcije verovatnoće  $L(\theta)$  i logaritam funkcije verovatnoće  $\ln L(\theta)$ . Kako bi se odredila koja je to vrednost  $\theta$  koja maksimizuje logaritam verovatnoće [41], potrebno je odrediti prvi izvod funkcije po  $\theta$ , izjednačiti ga sa nula a zatim rešiti jednačinu po  $\theta$ , kao u jednačini (2).



$$\frac{\partial \ln L(\theta)}{\partial \theta} = \sum_{i=1}^I [x_{si} - P_i(\theta)] \frac{P'_i(\theta)}{P_i(\theta)Q_i(\theta)} = 0 \quad (2)$$

Za pronalaženja nula jednačine primenjuje se dalje Njutn-Rapsonova metoda (engl. *Newton-Raphson*).

Procena maksimalne verovatnoće je nepouzdana za kratke testove i čak može biti beskonačna. To se dešava, na primer, kada ispitanik odgovori na sva pitanja tačno ili na sva pitanja odgovori netačno. U tom slučaju, procena maksimalne verovatnoće biće  $\pm\infty$  [110]. Ovo delimično može da predstavlja problem na početku adaptivnog testa kada je moguće da odgovori ispitanika budu u jednoj od kategorija (svi tačni, ili svi netačni). Stoga se najčešće na početku testa, koristi drugi metod, na primer Bajesov, dok se ne dobije stabilna procena sposobnosti s maksimalnom verovatnoćom. Dodatni problem je kada procena maksimalne verovatnoće ima veći broj lokalnih ekstrema. U tom slučaju traženjem nula jednačine (2) nije sigurno da će se dobiti globalni maksimum, nego samo lokalni maksimum ili čak minimum funkcije verovatnoće. Ovaj problem se uobičajeno rešava pažljivim odabirom početne vrednosti za Njutn-Rapsonovu metodu koja će omogućiti iteracije za lociranje globalnog maksimuma, a ne da se „zaglavi“ na lokalnom minimumu ili maksimumu [146].

#### 4.2.3.2. Bajesova procena

Značajna karakteristika Bajesove procedure za procenu sposobnosti je primena prethodne (*a priori*) raspodele  $p(\theta)$ , zajedno sa funkcijom verovatnoće  $L(\theta)$ . Prethodna distribucija predstavlja ono što je poznato o raspodeli  $\theta$  pre nego test započne. Stoga, Bajesova procedura omogućava ugradnju prethodnog znanja o populaciji koja se testira u proceni sposobnosti, omogućavajući tako efikasniju procenu [41]. Bajesova procena sposobnosti bazira se na Bajesovoj teoremi koja povezuje uslovne verovatnoće dva događaja preko njihovih verovatnoća (3).

$$P(\theta|X) \propto P(X|\theta) \cdot p(\theta) = L(\theta) \cdot p(\theta) \quad (3)$$

gde je  $X=(x_1, x_2, \dots, x_n)$ , šema odgovora na  $n$  postavljenih pitanja tokom testa. Izraz  $P(\theta|X)$  u jednačini (3) predstavlja *a posteriori* distribuciju  $\theta$ . Bajesova procedura procenjuje sposobnosti ispitanika merenjem centralne tendencije za *a posteriori* raspodelu.

Za razliku od procene maksimalne verovatnoće, Bajesova procena omogućava procenu sposobnosti čak i kada su svi odgovori ispitanika ili tačni, ili netačni [24], [41]. Pored toga, Bajesova procena nije pod uticajem redosleda kojim se pitanja odabiraju, a takođe daje manju srednju kvadratnu grešku nego procena maksimalne verovatnoće kada raspodela sposobnosti populacije odgovara *a priori* raspodeli [41].

Bajesova procena sposobnosti ima određene greške prilikom procene sposobnosti – ima tendenciju da potceni pojedince sa visokom sposobnošću kao i da preceni one sa slabim sposobnostima [110]. Ovo je poznato kao regresija ka srednjoj vrednosti i problem je, osim ako je broj pitanja velik. Osim toga, Bajesova procedura može da bude dodatno pristrasna ukoliko prethodna raspodela nije ispravno određena [147].

#### 4.2.4. Pravilo zaustavljanja

Svaki računarski adaptivni test mora da ima pravilo zaustavljanja koje određuje kada se prestaje sa administracijom testa. Pravila zaustavljanja mogu se svrstati u dve kategorije koje time određuju i dva tipa adaptivnih testova: test fiksne dužine i test promenljive dužine.

##### 4.2.4.1. Računarski adaptivni test sa fiksnim brojem pitanja

Kod računarskih adaptivnih testova fiksne dužine, test se sprovodi dok se ne postavi unapred predviđen broj pitanja. To znači da svaki ispitanik odgovara na isti broj pitanja. Ova vrsta testova ima prednost u pogledu jednostavnije implementacije i imaju bolje predviđanje stope korišćenja baze pitanja [133]. Testovi fiksne dužine daju percepciju pravednosti i lakši su za objašnjenje korisnicima. U slučaju kada ispitanici dobijaju različit broj pitanja, tada ispitanik koji je postigao loš rezultat na testu koji je imao relativno mali broj pitanja, može da tvrdi da nije dobio istu priliku kao drugi ispitanici da pokaže svoje znanje [16]. Iz tog razloga, računarski adaptivni testovi fiksne dužine su veoma popularni za različite vrste procene znanja, kao na primer GRE test (engl. *Graduate Record Examinations*); [103], sertifikacioni ispit za Američko udruženje kliničke patologije (engl. *American Society of Clinical Pathology ASCP*) [16] kao i test koji se koristi za određivanje kvalifikacija za vršenje službe u oružanim snagama SAD (engl. *Armed Service Vocational Aptitude Battery CATASVAB*) [128]. Međutim, administriranje računarski adaptivnog testa fiksne dužine negira jedno od najvažnijih prednosti adaptivnog testa u odnosu na neadaptivni, a to je da adaptivni test može da odredi sposobnost svakog ispitanika do istog nivoa preciznosti. Međutim, to se može postići jedino ako test nije unapred određene dužine [133].

##### 4.2.4.2. Računarski adaptivni test sa varijabilnim brojem pitanja

Računarski adaptivni testovi sa varijabilnim brojem pitanja, izvršavaju se sve dok se ne dostigne unapred definisana preciznost merenja. Kriterijum zaustavljanja može da bude zadana standardna greška (engl. *standard error SE*) ako je u pitanju procena maksimalne verovatnoće, odnosno zadana *posteriori* preciznost u slučaju Bajesove procene [133]. Glavna prednost adaptivnih testova sa promenljivim brojem pitanja je što se znanje svakog ispitanika proverava do istog nivoa preciznosti. Ispitanici koji imaju sposobnost koja se nalazi negde oko sredine raspodele sposobnosti, uglavnom rešavaju test sa manjim brojem pitanja nego ispitanici čije su sposobnosti na granici, bilo gornjoj ili donjoj [110]. Generalno, ovo znači optimalnije korišćenje pitanja iz baze jer mnogi ispitanici rešavaju test minimalne dužine, te efikasnije korišćenje vremena i napora ispitanika [16], [128]. Međutim, može biti teško da se objasni ispitanicima kako može da se izvrši uporediva procena znanja bazirana na testovima različite dužine. Pored toga, može se javiti potreba da se testovi produže i učine jednakim kako bi se

obezbedilo ekvivalentno pokrivanje nastavnog sadržaja [16]. Primer testova sa promenljivim brojem pitanja su testovi za medicinske sestre koji se primenjuju u SAD (engl. *National Certification Examination (NCE) for registered nurse anesthetists and the National Council Licensure Examination of Registered Nurses*), [16]. Moguće je primeniti i kombinovanu metodu pravila zaustavljanja testa, tako na primer autori u [133] savetuju da bi bilo poželjno uvek koristiti mešavinu „ciljne preciznosti“ i „maksimalnog broja pitanja“ kako bi se mogla postići neka tačno određena preciznost merenja jedino ako se ne desi da se sva pitanja iz baze pitanja upotrebe.

#### **4.2.5. Kontrola izloženosti pitanja**

Algoritam za testiranje koji spontano bira najbolju stavku (bez određenih pravila koja bi sprečila izbor određenog pitanja), bilo da se bazira na proceni maksimalne informacije ili na Bajesovoj proceni, često bira sličan set stavki za ispitanike na startu testa, naročito ako nisu poznate prethodne informacije o svakom ispitaniku na osnovu kojih bi se moglo odabrati početno pitanje [133]. Ispitanici sa sličnim sposobnostima mogu imati značajno preklapanje u setu pitanja koja su im postavljena tokom testa. Nekoliko studija je pokazalo da su u slučaju spontanog CAT algoritma neka pitanja iz baze bila data gotovo za svakog ispitanika [61], [104], [110]. Ovakva neravnoteža u korišćenju baze podataka je ekonomski nepoželjna jer predstavlja beskorisno ulaganje napora u stvaranje velike baze pitanja u kojoj se značajan deo stavki retko ili gotovo nikada ne primenjuje [120]. Pored neekonomičnosti, česta upotreba određenog broja pitanja utiče i na sigurnost samog testa. Ispitanici mogu da zapamte pitanja koja su bila i prenesu ih drugim, budućim ispitanicima. Ukoliko u algoritam za izbor pitanja ne postoji ugrađena kontrola izloženosti pitanja, velika je verovatnoća da će se ta pitanja vrlo uskoro ponoviti na testu.

Procedura za kontrolu izloženosti pitanja ima dva osnovna cilja: da spreči prekomerno izlaganje određenih stavki i da se poveća stepen upotrebe stavki koje se retko ili gotovo nikada ne koriste [120]. U literaturi se mogu naći tri kategorije strategija za kontrolu izloženosti stavki: procedure randomizacije, uslovne procedure i slojevite procedure [36], [35], [156].

##### **4.2.5.1. Procedure randomizacije**

Procedure randomizacije kontrolišu izbor sledećeg pitanja tako što iz skupa gotovo optimalnih pitanja nasumice biraju sledeće, umesto da se uvek bira pitanje koje je najinformativnije. Različite strategije randomizacije razlikuju se po tome na koji način se formira skup gotovo optimalnih pitanja i veličina ovog skupa. Primeri procedura randomizacije za kontrolu izloženosti su 5-4-3-2-1 tehnika, metod random red, „izaberi jedan od tri“ random procedura, metod unutar 0.10 logita i progresivna metoda. Prednost ovih procedura je što su jednostavne za razumevanje i relativno jednostavne za primenu. Međutim, one ne daju nikakvu garanciju da će stepen izloženosti biti u okviru datog nivoa.

#### 4.2.5.2. Uslovne procedure

Uslovne procedure kontrolišu verovatnoću da se svako pitanje pojavi u svakom pojedinom koraku testa, uslovno za dati kriterijum. Kriterijum je uobičajeno vezan za parametre kontrole izloženosti, koji ograničavaju maksimalni stepen izloženosti svake stavke za unapred utvrđeni nivo. Stoga, uslovne procedure imaju prednost jer obezbeđuju maksimalan stepen izloženosti. Međutim, one zahtevaju složene, vremenski zahtevne simulacije kako bi se odredili parametri kontrole izloženosti pitanja. Ove simulacije moraju se odraditi pre „puštanja u upotrebu“ CAT i mogu povećati složenost primene. Primeri uslovnih procedura su Simpson-Heterova procedura (engl. *Sympson-Hetter procedure*), Stoking i Luisova multinomijalna procedura (engl. *Stocking and Lewis multinomial procedure*), i tri-uslovna procedura (engl. *tri-conditional procedure*).

#### 4.2.5.3. Slojevite procedure

Kod slojevitih procedura, baza pitanja je podeljena na slojeve (engl. *strata*) na osnovu parametra diskriminacije svake stavke ( $a$ ). Rezultujući slojevi su složeni od najnižeg do najvišeg stepena diskriminacije. Test se zatim deli na faze koje se podudaraju sa brojem slojeva i unapred određen broj pitanja iz svakog sloja se obrađuje iz svakog sloja u odgovarajućoj fazi testa. Razlog za ovu slojevitu strategiju je da se na početku testa administrira najmanje diskriminativna stavka kada je tačnost procene sposobnosti mala, a da se visoko informativne stavke ostave za kasnije faze testa kada je neophodno da se sposobnost ispitanika istakne.

#### 4.2.6. Projektovanje baze pitanja

Baza pitanja ima vrlo važnu ulogu u računarskim adaptivnim testovima. Pitanja u bazi su indeksirana, strukturirana ili na neki drugi način im je dodeljena informacija koja može da se iskoristi kako bi se olakšao njihov izbor za test. Razlikuju se dva tipa baze podataka za tipičan CAT program. Jedna baza se uobičajeno naziva glavna (*master*) baza pitanja, koja uključuje što je moguće više pitanja za testiranje aplikacije. Druga je operativna baza pitanja, koja predstavlja manji podskup glavne baze pitanja, dizajnirana tako da bude dovoljno mala da računar može lako da pristupi pitanju i kada je neophodno da minimizuje izloženost pitanja. S druge strane, mora da bude dovoljno velika da obezbedi pitanja sa traženim karakteristikama. Zbog načina na koji funkcioniše CAT, vek trajanja pitanja u operativnoj bazi pitanja je ograničen. Nakon određenog broja puta što se pitanje našlo u upotrebi, pitanje je potrebno vratiti u glavnu bazu pitanja. Neka pitanja mogu se naći ponovo u upotrebi nakon isteka razumno dugog vremenskog perioda. Nedoumica koje uvek postoji pri projektovanju baze pitanja je koliko pitanja je potrebno da baza sadrži. U idealnom slučaju, što baza ima više pitanja to je bolje jer pruža više mogućnosti što se tiče izbora pitanja za test i smanjuje se mogućnost da se jedno te isto pitanje često ponavlja. Što je baza pitanja veća, manja je verovatnoća da će ispitanici moći da zapamte pitanja u slučaju kada im je baza pitanja dostupna. Takođe, veća baza pitanja znači da se može naći veći broj pitanja koja odgovaraju po sadržaju i formatu.

## 5. Teorija odgovora na stavke

Računarski adaptivni testovi zasnovani su na teoriji odgovora na stavke (engl. *Item Response Theory*, IRT). IRT je familija matematičkih funkcija koja pokušava da predvidi verovatnoću da ispitanik uspešno izvrši zadatak, ili tačnije, da tačno odgovori na pitanje. Osnovna premisa IRT modela je da je verovatnoća ispravnog odgovora funkcija sposobnosti, označava se grčkim slovom  $\theta$ , sa skalom koja se obično prikazuje u rasponu od -3 do +3, gde 0.0 predstavlja prosečnu sposobnost [132].

U nastavku sledi kratak prikaz klasične teorije testova i teorije odgovora na stavke.

Klasična teorija testova bavi se pouzdanošću testa i pretpostavlja da su stavke u testu nasumice (engl. *random*) odabrane iz grupe relevantnih stavki. Pouzdanost se posmatra kao karakteristika testa i varijanse obeležja koje meri. Stavke se tretiraju kao njihovo međusobno slučajno ponavljanje i njihovih karakteristika, ako su uopšte pregledane, iskazuju se u korelaciji sa ukupnim rezultatom na testu ili kao skriveni faktor opterećenja za promenljivu(e) od interesa. Karakteristike njihovih osobina ne analiziraju se detaljno. Ovo je navelo holandskog psihologa Mellenberga (G.J. Mellenbergh) [100] da napravi razliku između teorije testova [81] i teorija stavki [86], [118].

### 5.1. Klasična teorija testova

Klasična teorija testova je u svojoj osnovi povezana sa psihometrijskom teorijom koja predviđa ishode psihološkog testiranja kao što je težina pitanja i sposobnost ispitanika. Uopšteno govoreći, cilj klasične teorije testova je da razume i unapredi pouzdanost psiholoških testova.

Klasična teorija testova može grubo biti smatrana za sinonim teorije pravih rezultata (engl. *true score theory*). Termin „klasična” ne odnosi se samo na hronologiju ovog modela, nego označava i razliku ove teorije u odnosu na novije psihometrijske teorije, koje se uopšteno nazivaju teorija odgovora na stavke (engl. *Item Response Theory*). Srodni termini za označavanje ovih novih psihometrijskih teorija su moderne teorije latentnih osobina (engl. *modern latent trait theory*) ili moderne teorije.

Prema definiciji, klasična teorija testova pretpostavlja da svaka osoba ima pravi rezultat  $T$  (engl. *true score*), koji će se dobiti ako nema grešaka u merenju. Pravi rezultat svakog pojedinca definiše se kao očekivani broj tačnih odgovora naspram beskonačno mnogo nezavisnih ponavljanja testa. Na žalost, ispitanici nikada ne dobiju pravi rezultat, nego posmatrani rezultat  $X$  (engl. *observed score*), koji predstavlja linearnu kombinaciju pravog rezultata i komponente greške:

$$\begin{array}{ccccc} X & = & T & + & E \\ \text{posmatrani rezultat} & & \text{pravi rezultat} & & \text{greška} \end{array}$$

Klasična teorija testova bavi se odnosom tri promenljive X, T i E u populaciji. Ovi odnosi koriste se da bi se reklo nešto o kvalitetu rezultata testa. U tom smislu, najvažniji je koncept *pouzdanosti* (engl. *reliability*). Koeficijent pouzdanosti  $\rho_{XT}^2$ , definiše se kao odnos varijanse pravog rezultata  $\sigma_T^2$  i varijanse posmatranog rezultata  $\sigma_X^2$ , kao što je prikazano jednačinom (4):

$$\rho_{XT}^2 = \frac{\sigma_T^2}{\sigma_X^2} \quad (4)$$

Kako se varijansa posmatranog rezultata može prikazati kao zbir varijanse pravog rezultata i varijanse greške, jednačina se može napisati na sledeći način:

$$\rho_{XT}^2 = \frac{\sigma_T^2}{\sigma_X^2} = \frac{\sigma_T^2}{\sigma_T^2 + \sigma_E^2} \quad (5)$$

Jednačina (5), koja predstavlja odnos signal/šum, ima intuitivno značenje: pouzdanost rezultata testa raste kako se odnos varijanse greške i varijanse posmatranog rezultata smanjuje. Kvadratni koren pouzdanosti predstavlja odnos pravog i posmatranog rezultata.

### 5.1.1. Pouzdanost

Pouzdanost se ne može direktno odrediti jer bi to značilo poznavanje pravog rezultata, što je prema klasičnoj teoriji testova nemoguće. Međutim, procena pouzdanosti se može dobiti na različite načine. Jedan od načina za procenu pouzdanosti je konstrukcija tzv. paralelnog testa (engl. *parallel test*). Osnovna karakteristika paralelnog testa je da daje isti pravi rezultat i istu varijansu posmatranog rezultata za svakog pojedinca kao i originalni test. Ako imamo paralelne testove X i X', tada znači da je:

$$\varepsilon(X_i) = \varepsilon(X'_i) \quad (6)$$

i

$$\sigma_{E_i}^2 = \sigma_{E'_i}^2 \quad (7)$$

Pod ovim pretpostavkama, sledi da je korelacija između rezultata paralelnog testa jednaka pouzdanosti (dokaz se može naći u [81], poglavlje 2).

$$\rho_{XX'} = \frac{\sigma_{XX'}}{\sigma_X \sigma_{X'}} = \frac{\sigma_T^2}{\sigma_X^2} = \rho_{XT}^2 \quad (8)$$

Primena paralelnih testova za određivanje pouzdanosti je mučan posao jer je teško doći do paralelnih testova. U praksi, ovaj način se retko primenjuje. Umesto toga, istraživači koriste meru unutrašnje konzistentnosti poznatu kao Kronbahov  $\alpha$ , koeficijent pouzdanosti. Posmatrajmo test koji čini  $k$  stavki  $u_j, j=1, \dots, k$ . Ukupan rezultat testa definiše se kao suma pojedinačnih rezultata stavki, tako da je za pojedinačno  $i$ :

$$X_i = \sum_{j=1}^k U_{ij} \quad (9)$$

Tada je Kronbahov koeficijent  $\alpha$  jednak:

$$\alpha = \frac{k}{k-1} \left( 1 - \frac{\sum_{j=1}^k \sigma_{U_j}^2}{\sigma_X^2} \right) \quad (10)$$

Kronbahov koeficijent  $\alpha$  može se koristiti kao donja granica pouzdanosti. Stoga je pouzdanost rezultata testa u populaciji uvek veća od vrednosti Kronbahovog  $\alpha$  u toj populaciji. Dakle, ovaj metod je izvodljiv i empirijski, a kao rezultat toga veoma je popularan među istraživačima.

Kao što je već pomenuto, kod klasične teorije ispitivanje je dizajnirano tako da se postigne odgovarajuća definicija pouzdanosti. Pouzdanost ima zadatak da kaže nešto o opštem kvalitetu pitanja u testu. Opšta ideja je da je tim bolje što je pouzdanost veća. Klasična teorija testa ne govori o tome koliko treba da bude velika pouzdanost. Prevelika vrednost za  $\alpha$ , preko 0.9 ukazuje na redundantnost stavki, oko 0.8 se preporučuje za istraživanje ličnosti, dok preko 0.9 za ciljano testiranje pojedinaca na završnim (npr. maturalnim) ispitima. Mora se napomenuti da su ove vrednosti usvojene konvencijom i nisu zasnovane na opravdanim argumentima.

Kao nedostaci klasične teorije testova navodi se sledeće [177]:

- Fundamentalna pretpostavka u osnovi koncepta pouzdanosti o paralelnim merenjima teško je ostvariva u praksi.
- Zavisnost indeksa težine i indikatora diskriminativnosti zadataka od uzorka ispitanika.
- Procena pouzdanosti zavisi od sposobnosti uzorka ispitanika.

- Komparacija ispitanika prema sposobnostima merenoj skupom zadataka sadržanih u testu ograničena je na situaciju kada ispitanici rešavaju iste ili paralelne zadatke.
- Varijansa greške jednaka je za sve ispitanike.
- Ne daje osnovu za određivanje procene uratka ispitanika u zadatku.

Mnogi testovi postignuća i sposobnosti prilagođeni su za ispitanike prosečnih sposobnosti, pri čemu ne nude preciznu procenu sposobnosti za vrlo uspešne i vrlo neuspešne ispitanike. Značajna poteškoća jeste poređenje ispitanika koji rešavaju testove različite težine, budući da broj tačnih odgovora zavisi od težinske strukture zadataka [177].

Nedostaci klasične teorije prevaziđeni su razvojem modernih psihometrijskih koncepata, koje u osnovi predstavljaju sistem modela povezanih skupom zajedničkih pretpostavki organizovanih oko teorije latentnih osobina (engl. *latent trait theory*). Pored ovog, srodni termini koji označavaju moderne koncepcije su teorija karakteristične krive zadatka (engl. *Item Characteristic Curve*), ili u novije vreme teorija odgovora na stavke (engl. *Item Response Theory*). U literaturi na srpskom jeziku može se naći i termin teorija ajtemskih odgovora i teorija stavskih odgovora [42].

## 5.2. Teorija odgovora na stavke

Teorija odgovora na stavke (TOS) proučava skorovanje testova i pitanja zasnovano na pretpostavkama o matematičkim relacijama sposobnosti ispitanika (ili drugih karakteristika koje su predmet ispitivanja) i odgovora datih na pitanja. Teorija je usmerena na pojedinačni odgovor, matematički egzaktno modelira verovatnoću odgovora pojedinačnog ispitanika na datu stavku – ajtem [42]. Parametar osobe zove se latentna osobina ili sposobnost; ona može, na primer, da predstavlja inteligenciju osobe ili jačinu nekog stava. Parametri stavke su težina (lokacija), diskriminacija (nagib ili korelacija) i pseudopogađanje (donja asimptota).

Pioniri u oblasti teorije odgovora na stavke koja se pojavila 50-tih godina dvadesetog veka bili su psihometričar *Frederic M. Lord*, danski matematičar *Georg Rasch* i austrijski sociolog *Paul Lazarsfeld*, koji su nezavisno radili paralelna istraživanja u ovoj oblasti. Teorija odgovora na stavke postala je široko prihvaćena krajem 1970-ih i 80-tih godina kada su personalni računari dali istraživačima neophodnu snagu da primenjuju ovu teoriju. Najčešća primena je u obrazovanju gde psihometričari koriste ovu teoriju za razvoj i preradu ispita, održavanje baze pitanja i ujednačavanje nivoa teškoća koje nastaju usled uzastopnog ponavljanja ispita (na primer, da se omogući poređenje rezultata tokom vremena) [53].

Teorija odgovora na stavke podrazumeva tri pretpostavke [111]:

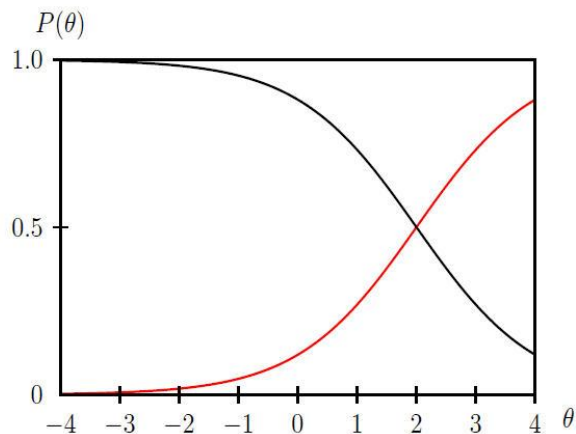
1. Bezdimezonu osobinu koja se obeležava sa  $\theta$ .



2. Lokalnu nezavisnost stavki.
3. Odgovor pojedinca na stavku može se modelovati matematičkom funkcijom koja se zove funkcija stavskog odgovora (engl. *Item Response Function – IRF*).

Sposobnost svakog pojedinca, predstavljena na horizontalnoj osi, obeležava se sa  $\theta$ , predstavlja bezdimenzionu veličinu i odnosi se na nivo latentne osobine. Vertikalna osa predstavlja verovatnoću  $P(\theta)$  tačnog odgovora kod zadatog nivoa  $\theta$ .

Verovatnoća odgovora  $P(\theta)$  za dihotomnu stavku sa dve kategorije odgovora, pozitivnom i negativnom, prikazana je na slici 5.1. Verovatnoća tačnog odgovora za dati nivo sposobnosti ( $\theta$ ) obeležena je na slici crvenom bojom, a netačnog odgovora crnom. Za bilo koju vrednost  $\theta$ , zbir ove dve verovatnoće je jednak 1. Kako se vrednost za  $\theta$  povećava, tako se i verovatnoća tačnog odgovora povećava a netačnog smanjuje. Pošto je verovatnoća netačnog odgovora  $Q(\theta)=1-P(\theta)$ , možemo pažnju usmeriti samo na verovatnoću tačnog odgovora.



Slika 5.1: Verovatnoća tačnog odgovora  $P(\theta)$  za dihotomnu stavku [111]

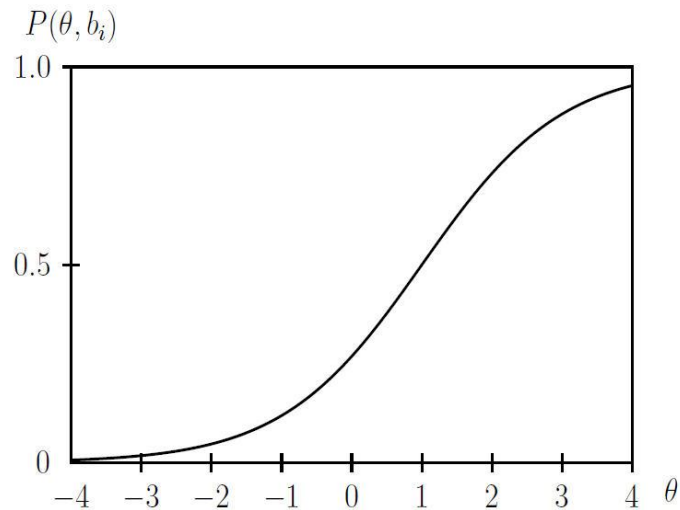
### 5.3. Jednparametarski model 1PL

Svaki TOS model predviđa verovatnoću da će određena osoba dati određeni odgovor na postavljenu stavku. Svaki pojedinac ima različit nivo sposobnosti, a i stavke se međusobno mogu razlikovati u mnogim aspektima, od kojih je najvažniji težina stavke – neke su teže, neke su lakše. Stoga verovatnoću sada obeležavamo sa  $P_{ij}$  umesto  $P$ , gde se indeks  $i$  odnosi na stavku, a indeks  $j$  na osobu. Kada su za stavku moguće više od dve opcije, neophodno je uvesti indeks koji će to predstavljati [111].

Verovatnoća zavisi od osobina stavke, što se označava parametrima stavke. Za dihotomne stavke postoje modeli TOS sa jednim, dva i tri parametra, a verovatnoće koje modeli predviđaju obeležavaju se kao  $P_{ij}(\theta_j, b_j)$ ,  $P_{ij}(\theta_j, b_i, a_i)$  ili  $P_{ij}(\theta_j, b_i, a_i, c_i)$ , gde su  $a_i$ ,

$b_i$  i  $c_i$  parametri stavke. Modeli za stavke sa više od dve opcije takođe će imati određene parametre pridružene opcijama odgovora.

Najjednostavniji TOS model za dihotomnu stavku ima samo jedan parametar. Stavka funkcija (tj. verovatnoća tačnog odgovora za jedan parametar  $b_i$  i individualni nivo sposobnosti  $\theta$ ), prikazana je na slici 5.2 i naziva se jednoparametarska logička funkcija (engl. *one-parameter logistic function* 1PL).



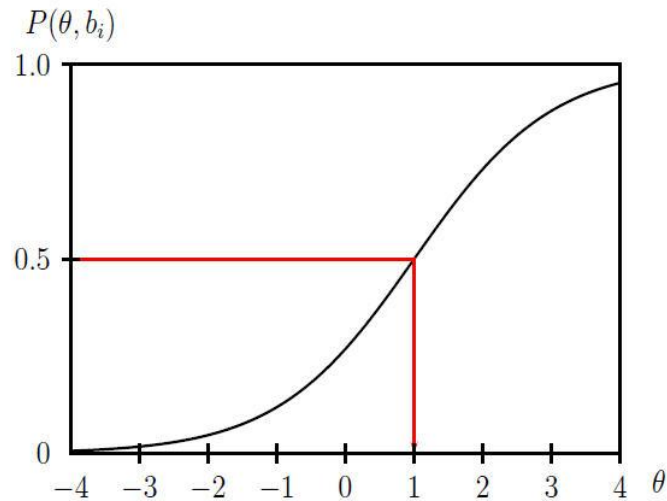
Slika 5.2: Stavka funkcija za jednoparametarski (1PL) model [111]

1PL funkcija data je jednačinom:

$$P_{ij}(\theta_j, b_i) = \frac{\exp(\theta_j - b_i)}{1 + \exp(\theta_j - b_i)} \quad (11)$$

1PL model predviđa verovatnoću tačnog odgovora na osnovu interakcije između individualne sposobnosti  $\theta$  i parametra stavke  $b_i$ . Parametar  $b_i$  naziva se parametar težine i predstavlja lokaciju stavke na kontinuumu osobine.

Može se naći položaj parametra  $b_i$  na osi sposobnost/težina stavke u tački u kojoj je predviđena verovatnoća  $P_{ij}(\theta_j, b_i) = 0.5$ , što je pokazano na slici 5.3. Parametar težina stavke prema teoriji odgovora na stavke, definiše se kao sredina funkcije. Stavka za koju je prikazana funkcija ima težinu 1.



Slika 5.3: Određivanje težine stavke na osi sposobnost/težina [111]

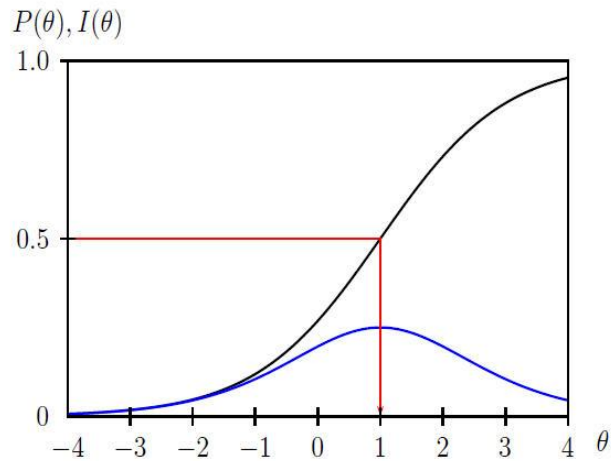
### 5.3.1. Funkcija informacije stavke za jednoparametarski model

Funkcija informacije stavke (engl. *Item information function*) ima važnu ulogu u teoriji odgovora na stavke. Svaka stavka u testu daje neku informaciju o sposobnostima ispitanika, ali količina ove informacije zavisi od toga koliko dobro se težina stavke poklapa sa sposobnostima ispitanika. U slučaju 1PL modela to je jedini faktor koji utiče na informaciju stavke, dok se u drugim modelima on kombinuje sa drugim faktorima.

Funkcija informacije stavke za 1PL model je data jednačinom:

$$I_i(\theta_i, b_i) = P_i(\theta, b_i)Q_i(\theta, b_i) \quad (12)$$

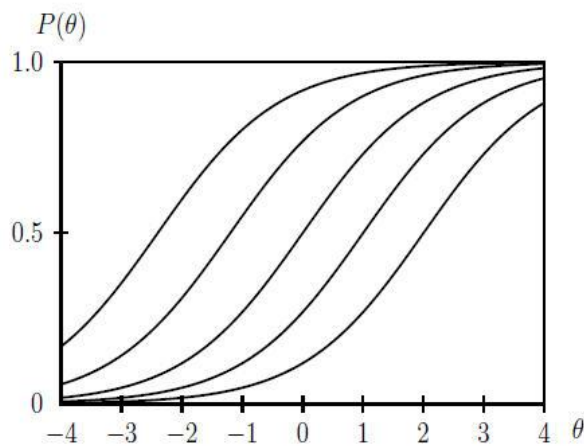
Maksimalna vrednost funkcije informacije stavke je 0.25, a nalazi se u tački gde su verovatnoća tačnog i netačnog odgovora jednake i iznose 0.5. Drugim rečima, bilo koja stavka u 1PL modelu je najinformativnija za ispitanika čija je sposobnost jednaka težini stavke. Kako sposobnost postaje manja ili veća od težine stavke, informacija stavke se smanjuje. Ovo je prikazano na slici 5.4. Najvažnija praktična posledica ovoga je da su nam potrebne stavke različite težine ako želimo da izvršimo kvalitetno merenje za ljude različitih sposobnosti.



Slika 5.4: Funkcija odgovora stavke i funkcija informacije stavke za 1PL model [111]

### 5.3.2. Funkcija odgovora testa 1PL modela

U teoriji, sposobnost može da ima bilo koju vrednost od  $-\infty$  do  $+\infty$ , ali u praksi vrednosti koje se posmatraju nalaze se na skali od -3 do +3 [55]. Na slici 5.5 prikazana je stavka funkcija odgovora za pet stavki čije su težine (-2.0, -1.5, 0.0, 1.0, 2.0) ravnomerno raspoređene duž najvažnijeg dela opsega sposobnosti. Krive su paralelne i nikada se ne presecaju.

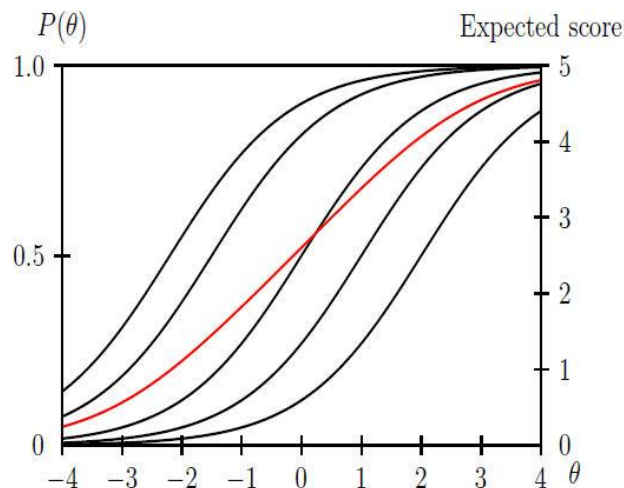


Slika 5.5: Funkcija odgovora stavke za pet stavki prema 1PL modelu [111]

Pet stavki nikako nije dovoljno za praktičnu primenu, ukoliko nisu pažljivo odabrane da odgovaraju sposobnostima ispitanika, kao što je slučaj u adaptivnom testiranju. Međutim, pomoću ovog primera uvodimo sledeći koncept u teoriji odgovora na stavke, a to je funkcija odgovora testa (engl. *test response function*). Svaka od pet funkcija odgovora na stavke, prikazanih na slici 5.5 predviđa verovatnoću da će osoba datih sposobnosti  $\theta_j$ , dati tačan odgovor na odgovarajuću stavku. Funkcija odgovora testa ima sličan zadatak, ali za test u celini, za bilo koju sposobnost  $\theta_j$  funkcija predviđa

očekivani rezultat testa. Razmotrimo prvo rezultat posmatranog testa. Bilo koja osoba koja pristupi rešavanju ovog testa sa pet stavki može da ima jedan od šest mogućih rezultata, u zavisnosti od toga da li ima 0, 1, 2, 3, 4 ili 5 tačnih odgovora. Za svaku od pet stavki i datu sposobnost  $\theta_j$  stavka funkcija odgovora predvideće verovatnoću tačnog odgovora, recimo  $P_{ij}(\theta_j, b_i)$ . Suma ovih verovatnoća daje očekivani rezultat za sposobnost  $\theta_j$ . Kao i posmatrani rezultat, očekivani rezultat može da ima vrednost između 0 i 5, ali za razliku od posmatranog rezultata to može biti bilo koji broj u datom opsegu i ne mora da bude ceo broj.

Slika 5.6 prikazuje funkciju odgovora testa (crvena linija) i pet stavskih funkcija odgovora. Skale za ove dve funkcije se razlikuju, stavka funkcija odgovora je na skali od 0 do 1, a funkcija odgovora testa na skali od 0 do 5. Funkcija odgovora testa je suma pet stavskih funkcija odgovora i bila bi mnogo strmija da je crtana u istom opsegu kao stavke funkcije. Kada test ima pet 1PL stavki, funkcija odgovora testa postaje 0 za ispitanika čija je sposobnost  $-\infty$ , 5 za ispitanika čija je sposobnost  $+\infty$ , i uzima neke druge vrednosti između 0 i 5 za sve ostale ispitanike. U ovom primeru, to će biti negde između 0.565 i 4.548 za ispitanika u opsegu sposobnosti  $(-3, +3)$ .



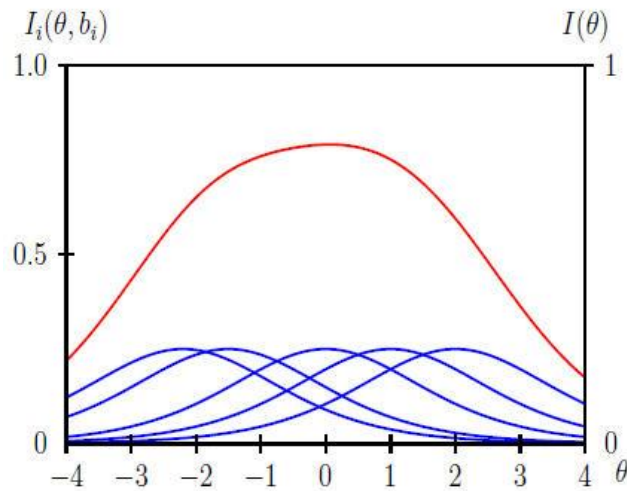
Slika 5.6: Funkcije stavskih odgovora za pet stavki prema 1PL modelu [111]

### 5.3.3. Funkcija informacije testa 1PL modela

Funkcija odgovora testa je suma funkcija stavskih odgovora, a funkcija informacije testa je suma funkcija informacije stavke:

$$I_j(\theta_j) = \sum_i I_{ij}(\theta_j, b_i). \quad (13)$$

Slika 5.7 prikazuje funkciju informacije testa za pet stavki koje su prikazane na slici 5.5.



Slika 5.7: Funkcija informacije stavke i funkcija informacije testa za pet stavki u 1PL modelu [111]

Test, u celini daleko je informativniji nego svaka pojedinačna stavka i daje informacije u širem opsegu sposobnosti. Informacija koju daje svaka stavka, nasuprot tome, koncentrisana je oko nivoa sposobnosti koji je blizak njegovoj težini.

#### 5.3.4. Preciznost i greška merenja

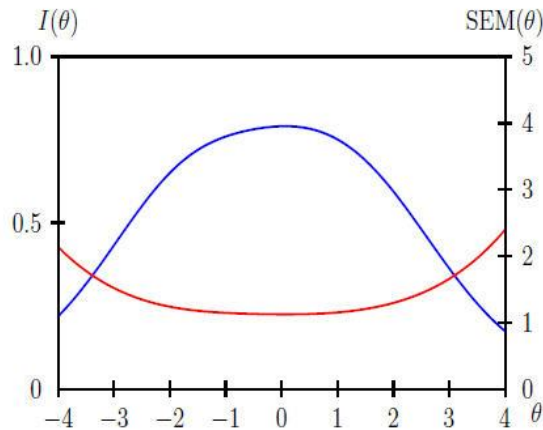
Varijansa procene sposobnosti  $\hat{\theta}$  može se odrediti kao recipročna vrednost funkcije informacije testa za  $\theta$ , što je dato sledećom jednačinom:

$$\text{Var}(\hat{\theta}) = \frac{1}{I(\hat{\theta})} \quad (14)$$

Kako je standardna greška merenja (engl. *standard error of measurement SEM*) jednaka kvadratnom korenu varijanse, za 1PL model biće:

$$\text{SEM}(\theta) = \sqrt{\frac{1}{I(\hat{\theta})}} = \frac{1}{\sqrt{\sum_i P_i(\theta, b_i) Q_i(\theta, b_i)}} \quad (15)$$

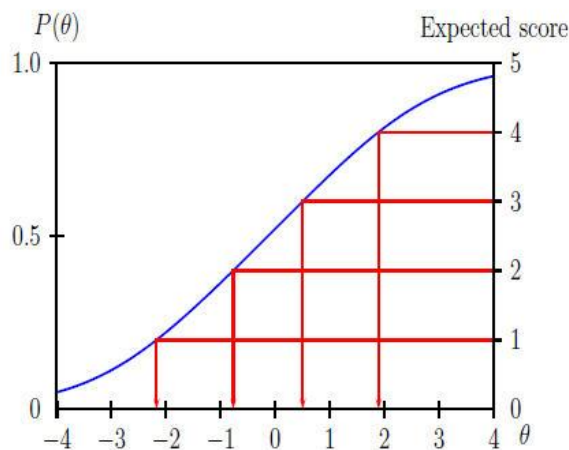
Slika 5.8 prikazuje funkciju informacije testa i standardnu grešku merenja za pet stavki sa slike 5.5. Funkcija informacije testa predstavljena je plavom bojom, a njene vrednosti mogu se očitati sa ose na levoj strani grafika. Standardna greška merenja obeležena je crvenom bojom, a njene vrednosti očitavaju se sa ose sa desne strane grafika. Može se primetiti da je standardna greška merenja skoro ravna za vrednost sposobnosti  $\theta$  koji se nalazi u opsegu  $(-2, +2)$ , a raste za vrednosti  $\theta$  van ovog opsega.



Slika 5.8: Funkcija informacije testa i standardna greška merenja 1PL modela za pet stavki [111]

### 5.3.5. Procena sposobnosti u 1PL modelu

Procena sposobnosti je verovatno najvažniji deo teorije odgovora na stavke. Razmotrimo procenu sposobnosti ispitanika koji rešava test od pet pitanja po 1PL modelu. Rezultat svakog ispitanika određen je brojem datih tačnih odgovora, a to je 0, 1, 2, ... 5 tačnih odgovora. Za 1, 2, 3 i 4 tačna odgovora možemo odrediti procenjenu sposobnost u tački gde je očekivani rezultat testa jednak posmatranom rezultatu. Drugim rečima, procena sposobnosti za dati posmatrani rezultat 1, 2, 3, 4 su one vrednosti  $\theta$  za koje funkcija rezultata testa uzima vrednosti 1, 2, 3, 4, kao što je prikazano na slici 5.9.



Slika 5.9: Nalaženje procene sposobnosti [111]

Funkcija odgovora testa u 1PL modelu za pet stavki teži 0 za ispitanika čija je sposobnost  $-\infty$ , a 5 za osobu čija je sposobnost  $+\infty$ . Na osnovu ovoga može se zaključiti da će u slučaju ni jednog tačnog odgovora procenjena sposobnost biti  $-\infty$ , a u slučaju svih tačnih odgovora (u posmatranom slučaju 5), biće  $+\infty$ .

### 5.3.6. Funkcija verovatnoće

Osoba koja rešava 1PL test sa  $k$  stavki može ostvariti jedan od  $k+1$  očekivanih rezultata (0, 1, ...,  $k$ ). Međutim, broj mogućih obrazaca odgovora (engl. *response pattern*) je znatno veći i iznosi  $2^k$ . Za test od pet stavki moguća su 32 različita obrasca odgovora. Svaki od njih ima određenu verovatnoću. Kako svaki ispitanik mora da ima neki od obrazaca odgovora, a obrasci se međusobno razlikuju, suma njihovih verovatnoća je 1. Ovo važi za skup podataka u celini, kao i za bilo koji određeni nivo sposobnosti.

Pitanje koje se postavlja je na koji način se može odrediti verovatnoća da će određena osoba sa sposobnošću  $\theta_j$  dati odgovore po određenom obrascu, npr. tačno, tačno, netačno, tačno, netačno. Teorija odgovora na stavke čini pretpostavku lokalne nezavisnosti. Drugim rečima, odgovori dati na pojedinačna pitanja u testu međusobno su nezavisni za datu sposobnost.

Kako su  $P(\theta_j, b_1)$ ,  $P(\theta_j, b_2)$ , ...  $Q(\theta_j, b_5)$  funkcije od  $\theta_j$ , možemo naći njihov proizvod kako bi dobili verovatnoću celog šablona (engl. *pattern*). Ovo sledi iz pretpostavke uslovne nezavisnosti prema kojoj odgovori dati na pojedinačna pitanja u testu su međusobno nezavisni s obzirom na  $\theta$ .

Funkcija:

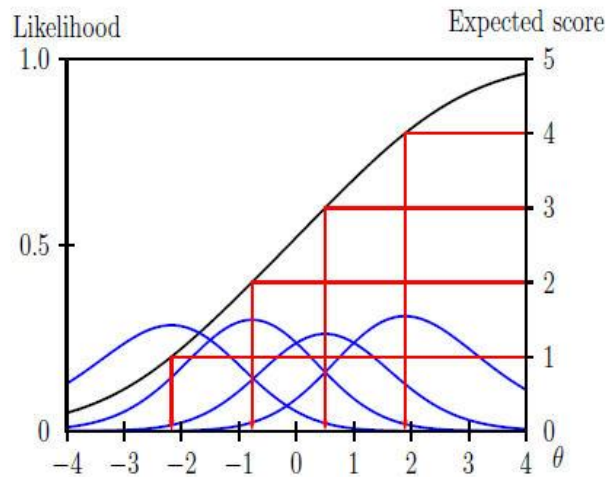
$$L(\theta) = \prod_i P_i(\theta, b_i)^{u_i} Q_i(\theta, b_i)^{1-u_i}, \quad u_i \in (0,1)$$

naziva se funkcija verovatnoće (engl. *likelihood function*), a  $u_i$  je rezultat za stavku  $i$ . To je verovatnoća šablona odgovora za datu sposobnost  $\theta$  i naravno parametara stavke. Postoji jedna funkcija verovatnoće za svaki šablon odgovora i suma svih takvih funkcija jednaka je 1 za bilo koju vrednost  $\theta$ .

### 5.3.7. Maksimalna verovatnoća procene sposobnosti

Na slici 5.10 plavom bojom prikazane su funkcije verovatnoće za sledeće šablone odgovora (T,F,F,F,F), (T,T,F,F,F), (T,T,T,F,F) i (T,T,T,T,F). Lako se uočava da funkcija verovatnoće ima maksimum za vrednosti procene sposobnosti koje su ranije određene (slika 5.9). Stoga će maksimalna funkcija verovatnoće dati istu procenu sposobnosti kao i prethodno razmatrani metod.

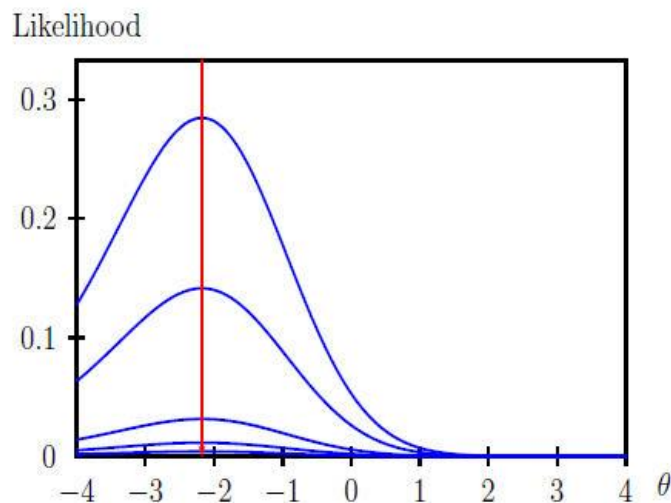




Slika 5.10: Pronalaženje procene sposobnosti po maksimalnoj verovatnoći [111]

U 1PL modelu procena sposobnosti zavisi samo od broja datih tačnih odgovora, a ne na koja pitanja je dat tačan odgovor. To ne znači da je funkcija verovatnoće invarijantna na šablon odgovora, nego samo da funkcije verovatnoće šablona sa jednakim brojem tačnih odgovora imaju maksimum za isti nivo sposobnosti.

Slika 5.11 prikazuje funkcije verovatnoće za pet šablona odgovora sa istim ukupnim brojem bodova 1. Svih pet funkcija daju jednaku procenu sposobnosti iako one nisu iste funkcije.



Slika 5.11: Funkcije verovatnoće za različite šablone odgovora sa ukupnim brojem bodova 1 [111]

Lako se može videti zašto se funkcije verovatnoće razlikuju: kada ispitanik može da ima samo jedan tačan odgovor očekujemo da to bude dat tačan odgovor na najlakše pitanje, pa donekle deluje iznenađujuće ako se ispostavi da je dat tačan odgovor na najteže pitanje.

## 5.4. Dvoparametarski model 2PL

Dvoparametarski model (2PL model) predstavlja generalizaciju 1PL modela i predviđa verovatnoću tačnog odgovora na bilo koju stavku na osnovu sposobnosti i dva parametra stavke.

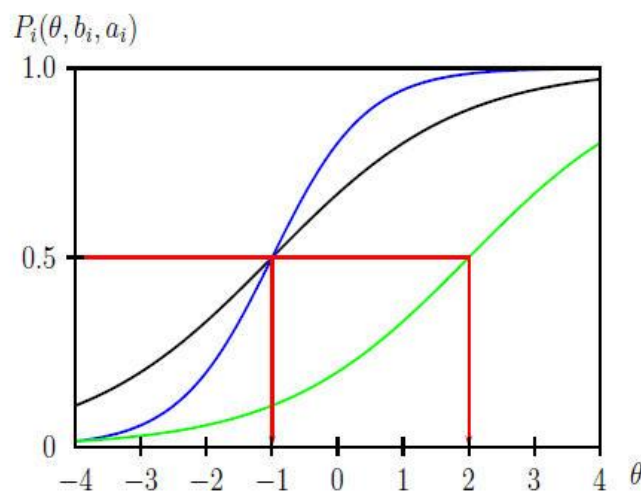
### 5.4.1. Funkcija odgovora stavke u 2PL modelu

Funkcija odgovora stavke u 2PL modelu definiše se na sledeći način:

$$P_{ij}(\theta_j, b_i, a_i) = \frac{\exp[a_i(\theta_j - b_i)]}{1 + \exp[a_i(\theta_j - b_i)]} \quad (16)$$

Kao i u 1PL modelu,  $b_i$  je parametar težine, a  $a_i$  je diskriminacioni parametar. Izgled funkcije odgovora za tri stavke prikazan je na slici 5.12. Na slici se može videti da dve od tri stavke imaju istu težinu -1, što daje verovatnoću od 0.5. Međutim, može se primetiti da je kriva obeležena plavom bojom mnogo strmija nego crna. Razlog je što stavka za plavu liniju ima znatno viši diskriminacioni faktor nego za crnu. Diskriminacioni parametri  $a_i$  ponekad se nazivaju i parametri nagiba (engl. *slope parameters*). Nagib funkcije odgovora stavke kod  $b$  je  $a/4$ . Zelena kriva ima isti nagib kao i crna, ali pomerena je udesno stoga stavka sa zelene krive ima isti diskriminacioni parametar kao i stavka sa crne krive, ali veće težine.

Za razliku od 1PL modela gde su krive koje predstavljaju funkciju odgovora stavke paralelne i nikada se ne seku, kod 2PL modela može doći do presecanja krivih. U slučaju prikazanom na slici 5.12, crna i plava kriva se seku. To znači da je stavka sa crne linije teža za ispitanika slabijih sposobnosti, dok stavka sa plave krive je teža za ispitanika većih sposobnosti.



Slika 5.12: Funkcije odgovora stavke za 2PL model [111]

### 5.4.2. Funkcija odgovora testa 2PL modela

Funkcija odgovora testa 2PL modela definiše se na isti način kao i u slučaju 1PL modela: ona predstavlja sumu stavskih funkcija odgovora za sve stavke testa i predviđa očekivani rezultat testa kao funkciju latentne sposobnosti i parametara stavke. Funkcija odgovora testa približava se 0 kada sposobnost teži  $-\infty$ , a približava se maksimalnom rezultatu (koji je jednak broju stavki) kada sposobnost teži  $+\infty$ . Stepen težine za svaku stavku je od  $-\infty$  do  $+\infty$ , međutim u praksi težine stavki imaju tendenciju da budu  $\leq 2$  za pitanja višestrukog izbora [55]. U normalnom slučaju kada su svi diskriminacioni parametri  $a_i$  pozitivni, funkcija odgovora testa je rastuća funkcija sposobnosti.

Stvarni oblik funkcije odgovora testa, odnosno njen nagib na bilo kom konkretnom nivou sposobnosti  $\theta$ , zavisi od parametara stavke. U idealnom slučaju bi trebalo da bude glatka i stalno rastuća kriva. Kriva sa skokovima ukazuje na to da očekivani rezultat testa reaguje na stvarnu sposobnost neravnomerno. Kada je kriva ravna, očekivani rezultat nije previše osetljiv na razlike u stvarnoj sposobnosti. Strmija kriva ukazuje da je očekivani rezultat osetljiviji na razlike u sposobnostima. Drugim rečima, test „diskriminiše” ili pravi bolju razliku između osoba različitih sposobnosti, što objašnjava termin *diskriminacioni parametar*.

### 5.4.3. Funkcija informacije stavke 2PL modela

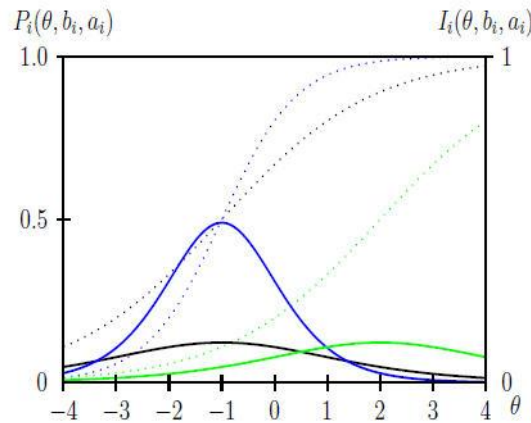
Funkcija informacije stavke 1PL modela definisana je kao:

$$I_i(\theta_i, b_i) = P_i(\theta, b_i)Q_i(\theta, b_i) \quad (17)$$

Za 2PL model funkcija informacije stavke postaje:

$$I_i(\theta_i, b_i, a_i) = a_i^2 P_i(\theta, b_i)Q_i(\theta, b_i). \quad (18)$$

Diskriminacioni faktor  $i$  u ovoj jednačini figuriše i njegov uticaj je popriličan s obzirom da se u formuli javlja kvadrat njegove vrednosti. To znači da će vrednost ovog parametra  $<1$  značajno smanjiti funkciju informacije, dok će vrednost parametra  $>1$  značajno povećati. Ovo možemo da vidimo na slici 5.13 gde je funkcija odgovora stavke predstavljena tačkastom linijom a boja krivih usklađena je sa pripadajućom funkcijom informacije stavke.



Slika 5.13: Funkcije odgovora stavke i funkcije informacije stavke 2PL modela za 3 stavke [111]

U 1PL modelu sve funkcije informacije stavki imaju isti oblik, isti maksimum 0.25 i samo su pomerene duž ose sposobnosti tako da svaka funkcija informacije stavke ima maksimum u tački u kojoj je sposobnost jednaka težini stavke. U 2PL modelu funkcija informacije stavke i dalje dostiže maksimum na težini stavke. Međutim, njihov oblik i vrednost maksimuma jako zavise od diskriminacionog parametra. Kada je diskriminacija visoka (i funkcija odgovora stavke strma) stavka daje više informacija o sposobnosti i informacija je koncentrisana oko težine stavke. Stavke sa niskim diskriminacionim parametrima pružaju manje informacija, a informacija je rasejana duž većeg dela opsega sposobnosti.

#### 5.4.4. Funkcija informacije testa 2PL modela

Funkcija informacije testa 2PL modela definiše se na isti način kao i kod 1PL modela: to je suma funkcija informacije stavki koje čine test. Međutim, kako se funkcija informacije stavke razlikuje u 1PL i 2PL modelu, to će funkcija informacije testa u slučaju 2PL modela biti:

$$I_j(\theta_j) = \sum_i I_{ij}(\theta_j, b_i, a_i) = \sum_i a_i^2 P(\theta, b_i, a_i) Q(\theta, b_i, a_i) \quad (19)$$

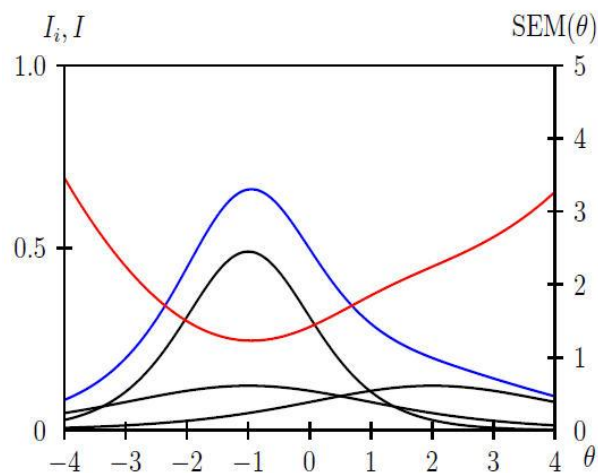
Kako u 2PL modelu funkcija informacije stavke je zavisna od diskriminacionih parametara  $a_i$ , oblik funkcije informacije testa može postati nepredvidiv, naročito u slučaju testova sa manjim brojem stavki. U praksi, trebalo bi da imamo funkciju informacije testa relativno glatku u opsegu sposobnosti koji je od značaja, tj. (-3,3). Ovo bi moglo biti savršeno ostvareno sa velikim brojem stavki koje imaju visoke diskriminacione parametre i stepen težine ravnomerno raspoređene duž opsega sposobnosti. Stavke sa veoma niskim diskriminacionim parametrima najčešće se eliminišu iz praktične upotrebe.

### 5.4.5. Standardna greška merenja 2PL modela

Kao i u slučaju 1PL modela, varijansa procene sposobnosti u 2PL modelu može se odrediti kao recipročna vrednost funkcije informacije testa za procenjenju sposobnost  $\hat{\theta}$ . Standardna greška merenja (SEM) jednaka je kvadratnom korenu varijanse, pa je u slučaju 2PL modela:

$$SEM = \sqrt{\frac{1}{\sum_i a_i^2 P_i(\theta, b_i, a_i) Q(\theta, b_i, a_i)}} \quad (20)$$

Na slici 5.14 predstavljene su funkcije informacije stavke (crna), funkcija informacije testa (plava) i funkcija standardne greške merenja (crvena) za tri stavke koje su predstavljene na slici 5.12. Kako ima mali broj stavki, težine su neravnomerno raspoređene i parametri diskriminacije se u velikoj meri razlikuju, na funkciju informacije testa i SEM funkciju dominantan uticaj ima jedna visoko diskriminantna stavka.



Slika 5.14: Funkcije informacija i SEM za stavke sa slike 5.12 [111]

### 5.4.6. Procena sposobnosti u 2PL modelu

Kao što je već ranije rečeno za 1PL model, procena maksimalne verovatnoće (engl. *maximum likelihood estimation*, MLE) sposobnosti:

1. može se naći na skali sposobnosti u tački u kojoj je očekivani rezultat jednak sa posmatranim rezultatom
2. zavisi samo od broja tačnih odgovora, a ne i na koje odgovore je dat tačan odgovor

U 2PL modelu, prva od ove dve stavke primenjuje se u izmenjenom obliku, dok druga uglavnom nije tačna. Formalno, prva stavka tvrdi da za 1PL model:

$$\sum_i P(\hat{\theta}, b_i) = \sum_i u_i \quad (20)$$

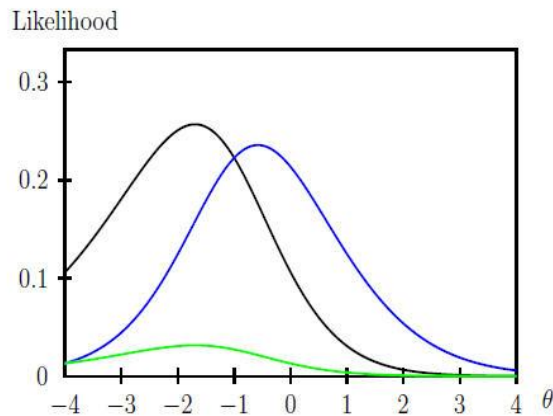
gde  $u_i \in (0,1)$  je rezultat za stavku  $i$ . U slučaju 2PL modela, ova jednačina postaje:

$$\sum_i a_i P(\hat{\theta}, b_i, a_i) = \sum_i a_i u_i \quad (21)$$

Tako da umesto jednostavne sume sada imamo ponderisane sume sa diskriminacionim parametrima stavki  $a_i$  kao ponderima.

Sve dok se parametri  $a_i$  razlikuju, što je najčešće slučaj, različiti šabloni odgovora sa jednakim brojem tačnih odgovora više neće dovesti do jednake procene sposobnosti. Sada, u slučaju 2PL modela, važno je i na koju stavku je dat tačan odgovor a ne samo koliko tačnih odgovora ima.

Ovo je jasno uočljivo sa slike 5.15 na kojoj su prikazane funkcije verovatnoće za tri šablona odgovora koji daju jednak posmatrani rezultat 1, tj. (T,F,F), (F,T,F) i (F,F,T).



Slika 5.15: Funkcije verovatnoće za tri funkcije odgovora koje imaju jednak posmatrani rezultat 1 [111]

Kriva obeležena zelenom bojom je vrlo ravna, što odražava malu verovatnoću tačnog odgovora na teže pitanje a javlja se zajedno sa netačnim odgovorima na dva lakša pitanja. Vrlo je važno primetiti da svaka od krivih ima maksimum u različitoj tački na osi sposobnosti, što nije bio slučaj kod 1PL modela.

Princip maksimalne verovatnoće i dalje je u važnosti i procena maksimalne verovatnoće sposobnosti može se odrediti u tački gde funkcija verovatnoće za posmatrani šablon odgovora dostiže maksimum. Na primer, test sa pet pitanja ima 32

moguća šablona odgovora. Procena maksimalne verovatnoće za šablon (F,F,F,F,F) je -1, a za šablon (T,T,T,T,T) je +1, a procena maksimalne verovatnoće za preostalih 30 šablona može se odrediti povećanjem verovatnoće.

## 5.5. Troparametarski model 3PL

Troparametarski model predstavlja zapravo 2PL model čija je funkcija odgovora stavke izmenjena tako da donja asimptota bude veća od nule. Drugim rečima, verovatnoća tačnog odgovora više se ne približava nuli kada prava sposobnost teži  $-\infty$ . Umesto toga, ona teži nekoj pozitivnoj vrednosti, najčešće  $1/k$ , gde je  $k$  broj ponuđenih odgovora kod pitanja sa višestrukim izborom. Razlog za to je što će ispitanici vrlo niske sposobnosti verovatno preći na pogađanje odgovora, a to im pruža verovatnoću od  $1/k$  da će odabrati („pogoditi“) tačan odgovor.

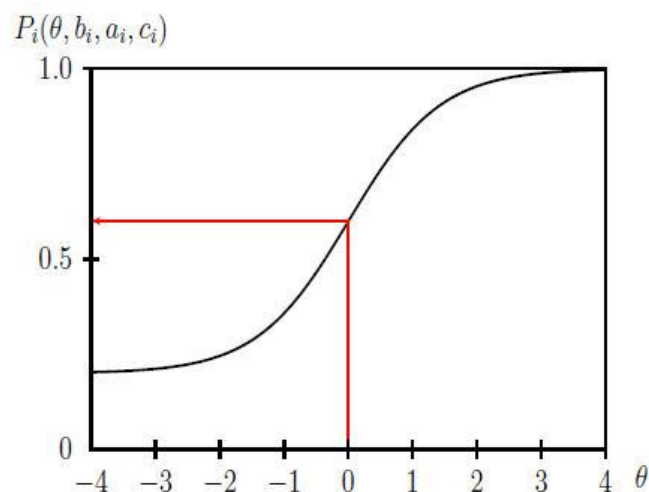
### 5.5.1. Funkcija odgovora stavke 3PL modela

Funkcija odgovora stavke 3PL modela je:

$$P(\theta, a, b, c) = c + (1 - c) \frac{\exp(a(\theta - b))}{1 + \exp(a(\theta - b))} \quad (22)$$

Treći parametar  $c$ , postavlja donju asimptotu, tj. verovatnoću tačnog odgovora kada prava sposobnost teži  $-\infty$ . Deo u jednačini koji se množi sa  $(1-c)$  predstavlja funkciju odgovora stavke 2PL modela sa različitim numeričkim vrednostima za  $a$  i  $b$ .

Na slici 5.16 prikazana je funkcija odgovora stavke 3PL modela za  $a=1.4$ ,  $b=0$  i  $c=0.2$ . Kao i kod 1PL i 2PL modela kriva menja oblik iz konveksnog u konkavni za  $\theta=b$ , međutim sada verovatnoća tačnog odgovora za  $\theta=b$  nije više 0.5, nego je sada  $c+(1-c)/2=0.6$ . Osim toga, u  $b$  je nagib sada  $(1-c)a/4$  a ne  $a/4$ .



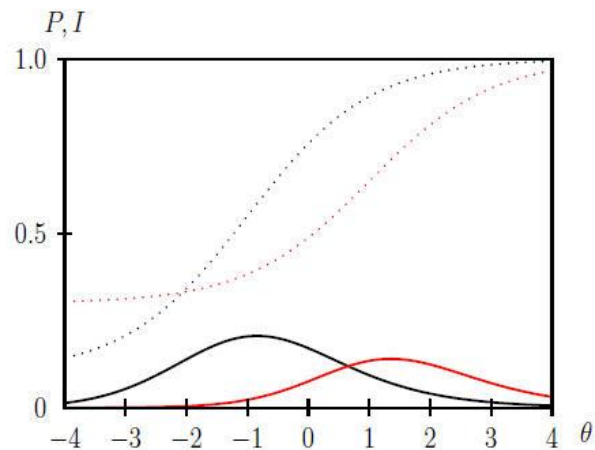
Slika 5.16: Funkcija odgovora stavke u 3PL modelu [111]

### 5.5.2. Funkcija informacije stavke 3PL modela

Funkcija informacije stavke 3PL modela data je sledećom jednačinom:

$$I(\theta, a, b, c) = a^2 \frac{Q(\theta)}{P(\theta)} \left[ \frac{P(\theta) - c}{1 - c} \right]^2 \quad (23)$$

Slika 5.17 prikazuje funkciju odgovora stavke (tačkasta linija) i funkciju informacije stavke (puna linija) za dve 3PL stavke. Stavka predstavljena crnom bojom ima sledeće parametre  $a=1$ ,  $b=-1$ ,  $c=0.1$ , dok su parametri stavke predstavljene crvenom bojom  $a=1$ ,  $b=+1$ ,  $c=0.3$ . Parametar  $b$  pomera krivu na levo ili desno, ali ne utiče na njen oblik. Dve posmatrane stavke imaju jednake parametre  $a$ , dok im se parametar  $c$  razlikuje. Veća vrednost za  $c$  vodi ka ukupnom smanjenju informacije o stavki.



Slika 5.17: Funkcija odgovora i funkcija informacije stavke 3PL modela [111]

### 5.5.3. Funkcija odgovora testa 3PL modela

Kao i u 1PL i 2PL modelu, funkcija odgovora testa definiše se kao suma funkcija odgovora stavki, svih stavki koji čine test i predviđa očekivani rezultat testa kao funkciju latentnih osobina i parametara stavki. Jedinu razliku između ova tri modela čine funkcije stavskih odgovora koje se sumiraju. Kada su svi diskriminacioni parametri  $a_i$  pozitivni, funkcija odgovora testa je rastuća funkcija sposobnosti. Njen stvarni oblik zavisi od parametara stavke.

### 5.5.4. Funkcija informacije testa 3PL modela

Funkcija informacije testa 3PL modela definiše se slično kao i kod 1PL i 2PL modela: to je suma funkcija informacije stavke svih stavki koje čine test. Ono što razlikuje ova tri modela su funkcije informacije stavki, pa je za 3PL model funkcija informacije testa data jednačinom:



$$I_j = \sum_i I_{ij}(\theta_j, a_i, b_i, c_i) = \sum_i a^2 \frac{Q(\theta)}{P(\theta)} \left[ \frac{P(\theta) - c}{1 - c} \right]^2 \quad (24)$$

Funkcija informacije stavke jako zavisi od diskriminacionih parametara  $a_i$ . U 3PL modelu, postoji dodatni uticaj parametra „pogađanja“  $c_i$ . Veća vrednost za  $c_i$  smanjuje funkciju informacije stavke i pomera njen maksimum dalje od  $b_i$ . U praktičnim primenama, poželjno je imati funkciju informacije testa koja je relativno glatka unutar opsega sposobnosti koji je od interesa, tj. (-3,+3).

### 5.5.5. Standardna greška merenja 3PL modela

Standardna greška merenja (SEM) jednaka je kvadratnom korenu varijanse, te je za 3PL model data jednačinom:

$$SEM(\theta) = \sqrt{\frac{1}{\sum_i a^2 \frac{Q(\theta)}{P(\theta)} \left[ \frac{P(\theta) - c}{1 - c} \right]^2}} \quad (25)$$

### 5.5.6. Procena sposobnosti u 3PL modelu

Analogno 1PL i 2PL modelu, funkcija verovatnoće u 3PL modelu definiše se na isti način, samo što sada verovatnoće P i Q odgovaraju 3PL modelu:

$$L(\theta) = \prod_i P_i(\theta, b_i, a_i, c_i)^{u_i} Q_i(\theta, b_i, a_i, c_i)^{1-u_i} \quad (26)$$

gde je  $u_i \in (0,1)$  rezultat za stavku  $i$ . Kao i u slučaju 2PL modela, procena maksimalne verovatnoće ne zavisi samo od broja tačnih odgovora, nego i na koje pitanje je dat tačan odgovor. Neželjena komplikacija koja može da nastane kod 3PL modela je da funkcija verovatnoće može da ima lokalne maksimume.

### 5.5.7. Pogađanje i 3PL model

U 3PL modelu uvođenjem trećeg parametra stavke, asimptote  $c$ , pokušava se prilagoditi pogađanju. Međutim, nije naročito zgodno pridružiti pogađanje kao osobinu stavke, jer je to osobina ljudi. Neki ljudi su skloniji pogađanju, a neki ne, uz sve druge osobine jednake. Međutim, mogu nastati situacije kada je svaki ispitanik prisiljen da pogađa, kao na primer u slučaju računarski administriranog testa, kada softver ne dozvoljava nastavak testiranja dok se ne da odgovor na postavljeno pitanje. U tom slučaju, ispitanik ako ne zna odgovor, prisiljen je da „pogađa“ kako bi nastavio test.

Kada se dešava pogađanje dolazi do mešanja dva modela. Jedan od modela, model sa pogađanjem, predviđa verovatnoću tačnog odgovora koja ne zavisi od sposobnosti i jednaka je  $1/k_i$ , gde je  $k_i$  broj ponuđenih odgovora na  $i$ -to pitanje. Drugi model može da bude 2PL model:

$$P(\theta_j, a_i, b_i, \pi_{ij}) = \pi_{ij} \frac{1}{k_i} + (1 - \pi_{ij}) \frac{\exp(a_i(\theta_j - b_i))}{1 + \exp(a_i(\theta_j - b_i))} \quad (27)$$

Faktori  $\pi_{ij}$  i  $1 - \pi_{ij}$  mogu se protumačiti kao verovatnoća da osoba odgovara prema modelu pogađanja ili prema 2PL modelu. Ovi faktori zavise od osobe koja se podvrgava testu jer nema svako jednaku sklonost ka „varanju“. Posmatrani faktori zavise od interakcije između sposobnosti i težine pitanja, pošto varanje ima smisla samo kada je pitanje preteško za datu sposobnost. Na taj način dolazimo do nečega što liči na 3PL model ali neuporedivo je složenije, jer sada umesto  $c_i$  figuriše  $\pi_{ij}$  koji na složen način zavisi i od osobe i od pitanja.

U praksi, stvari se mogu još više zakomplikovati kada se pogađanje ne dešava slučajno. Naime, u zavisnosti od sposobnosti ispitanika, ispitanik može logičkim razmišljanjem da odbaci najmanje verovatne odgovore a zatim da pogađa tačan odgovor među preostalim odgovorima. Step en uspeha ispitanika u ovom slučaju zavisi od pitanja i biće veći ako su neki od distraktora loše odabrani.

## 6. Dizajniranje računarskog adaptivnog testa

Većina CAT sistema koji su realizovani u praksi ne nalazi se u slobodnoj upotrebi, tako da ukoliko neko želi da koristi neki od CAT sistema mora da izvrši plaćanje licence za korišćenje, ili da sam razvija svoj sistem za adaptivno testiranje. Postoji nekoliko platformi za adaptivno testiranje koje su otvorenog koda i u nastavku je dat njihov kratak prikaz.

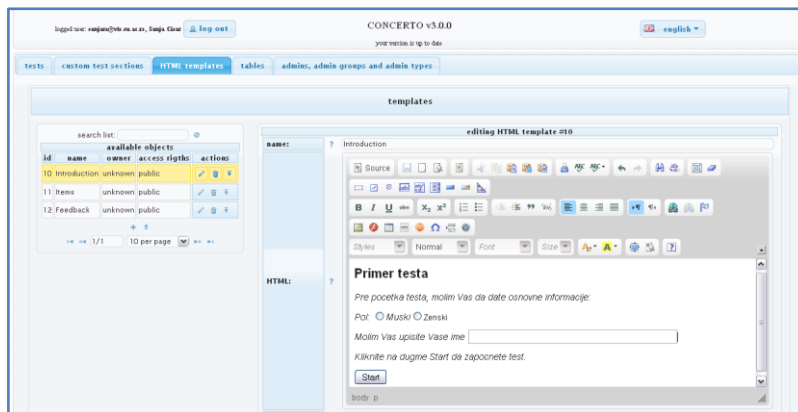
### 6.1. Primeri aplikacija za adaptivno testiranje

#### 6.1.1. Concerto

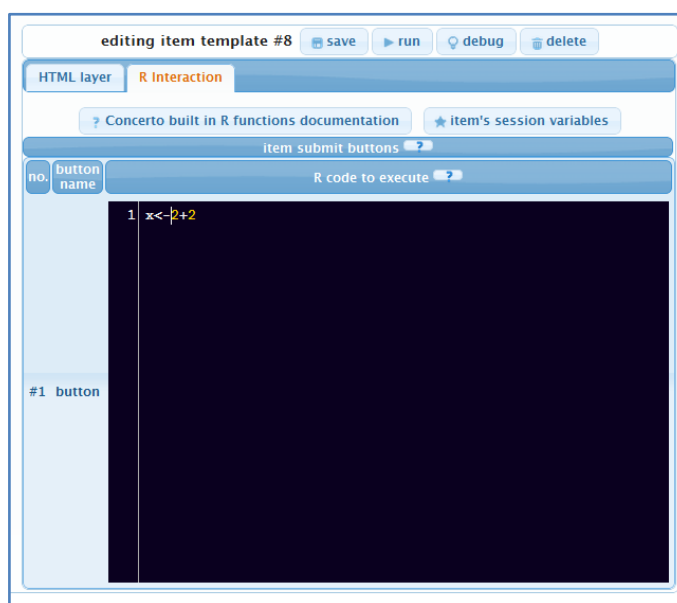
Jedan od adaptivnih testova koji su otvorenog koda (engl. *open source*) jeste *Concerto*, koji je dostupana na sledećoj adresi <http://www.psychometrics.cam.ac.uk>. Ideja tima koji je radio na stvaranju *Concerta*, bila je da se stvori jednostavan alat za razvoj adaptivnog testa. Platforma se oslanja na tri glavna elementa: HTML prezentacioni sloj, R programski jezik (koji je postao standard među statističarima koji razvijaju statistički softver [45] i SQL za komunikaciju sa bazom podataka. Projekat je razvijan u Psihometrijskom centru Univerziteta Kembridž (*The Psychometrics Centre of the University of Cambridge*). Sistemski zahtevi za ovu aplikaciju su PHP (verzija v5.3 ili novija), MySQL (verzija v5.0 ili novija) i R (verzija v2.12 ili novija, može se naći na adresi <http://www.r-project.org>).

Administrativni panel omogućava stvaranje korisničkih naloga, podešavanje prava pristupa, pravljenje ili izmena testa, kao i korišćenje R studija. Postoje određeni problemi u radu aplikacije ako se koristi čitač *Mozilla Firefox*, te se preporučuje *Google Chrome*. Administrativni panel ima višejezičku podršku, za sada na engleskom, poljskom, bugarskom i kineskom (pojednostavljena verzija) jeziku, a u toku je prevođenje na francuski i hrvatski jezik. *Concerto* daje mogućnost pravljenja jednostavnih testova dihotomnog tipa, kao i adaptivnog testa.

Administrativni panel (slika 6.1) omogućava kreiranje i editovanje testova, pravljenje korisničkih naloga, postavljanje prava pristupa i korišćenje R studija. Tab **Items** predstavlja glavni deo *Concerta*, koji omogućava kreiranje/izmene/debugovanje HTML šablona i pisanje R koda (slika 6.2) za izvršavanje akcija koje su dodeljenje kontrolama koje korisnik može da aktivira (npr. aktivirano dugme na formi).

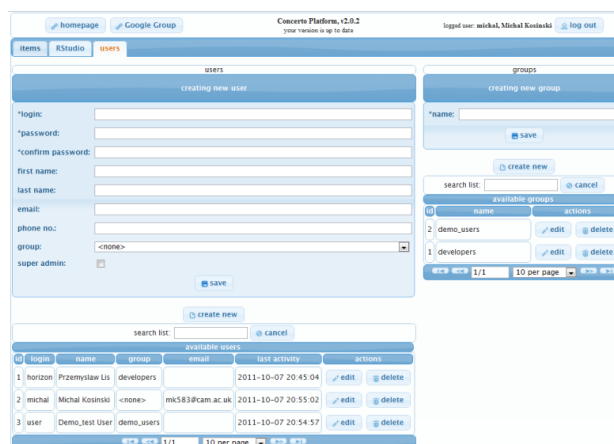


Slika 6.1: Administrativni panel Concerta



Slika 6.2: Tab R interakcija

Tab **Users** dostupan je samo administratoru i omogućava mu da dodaje/briše/menja podatke o korisniku i korisničkim grupama, kao i da definiše prava pristupa (slika 6.3).

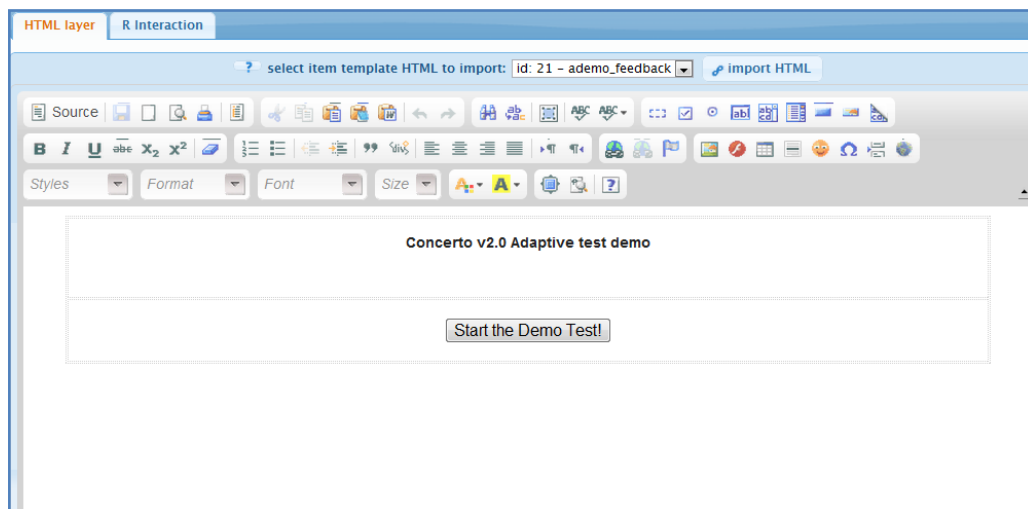


Slika 6.3: Tab Users

Concerto nudi mogućnost pravljenja adaptivnog testa koji se bazira na Teoriji odgovora na stavke i dvoparametarskom modelu, koji je objašnjen u poglavlju 5.4. Pravljenje testa zahteva postojanje tri HTML šablona:

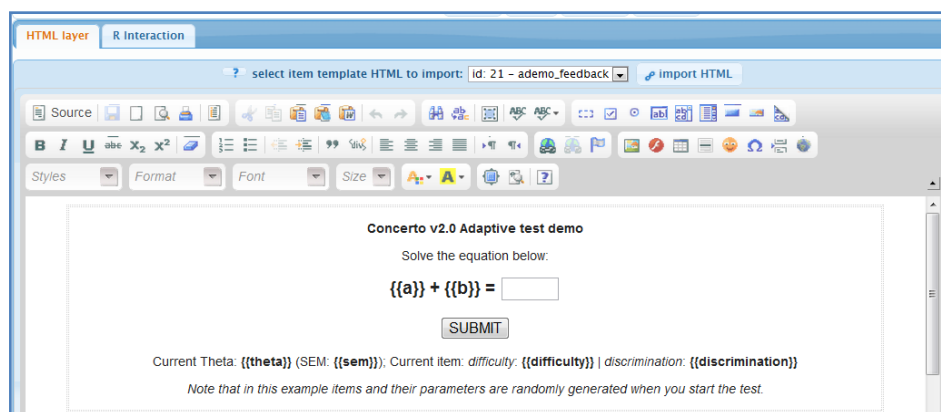
- **Uvod (Introduction)** koji omogućava korisniku da pokrene test
- **Test pitanja (Test questions)** definisanje pitanja
- **Povratna sprega (Feedback)** koja daje prikaz rezultata korisnika.

Prvi HTML šablon sadrži pozdravnu poruku i dugme koje omogućava pokretanje testa, kao što je prikazano na slici 6.4.



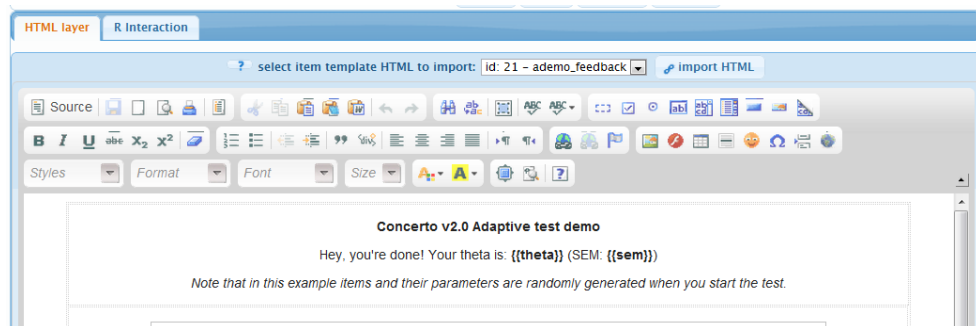
Slika 6.4: Uvodni ekran adaptivnog testa napravljenog uz pomoć Concerta

Drugi HTML šablon koristi se za definisanje pitanja koje će biti postavljeno. Pri definisanju pitanja mora se poštovati R sintaksa (slika 6.5). U šablon se postavlja tekstualno polje u koje korisnik unosi svoj odgovor. Ovo tekstualno polje trebalo bi da se zove „response“ jer se tako referiše u R kodu. Ukoliko se dodeli neko drugo ime, potrebno je voditi računa da se i u kodu izvrši izmena. Na kraju, dodaje se dugme za potvrđivanje odgovora.



Slika 6.5: HTML šablon za definisanje pitanja

Treći HTML šablon daje korisniku povratnu informaciju da li je njegov odgovor tačan ili ne, i daje odgovarajuće vrednosti za  $\theta$  (procenjena sposobnost) i SEM (standardna greška merenja), što ilustruje slika 6.6.



Slika 6.6: HTML šablon koji prikazuje povratnu informaciju korisniku

Nakon što se naprave ova tri šablona, koji određuju grafički izgled testa, neophodno je dodeliti odgovarajući R kod za dugmad koja su postavljena na prvi i drugi šablon. Da bi mogao da se napiše odgovarajući R kod, neophodno je odabrati tab *R interaction* i dodati ga u polje koje se zove *R code to execute*. Za prvi šablon, koji predstavlja uvodni ekran, za dugme *Start*, potrebno je dodati sledeći kod:

```
### Define the stopping rules - they will be used later
max_items <- 15
min_sem <- 0.2
### Simulate the item parameters
ibsize <- 100 # size of the item bank to generate
### make up parameters
discrimination <- sample(100:200, ibsize)/100 ## random values
from 1.00 to 2.00
difficulty <- sample(-200:200, ibsize)/100 ## random values
from -2.00 to 2.00
guessing <- sample(0,ibsize, replace=T) ## we use 2pl model
so the guessing is always 0
inattention <- sample(1,ibsize, replace=T) ## 2pl model:
inattention always 1

### simulate item contents(a + b = response)
a <- sample(1:10, ibsize, replace = T)
b <- sample(1:10, ibsize, replace = T)
### compute correct responses
correct.response <- a+b

# merge parameters into itemBank
itemBank <- cbind(discrimination, difficulty, guessing, inattention)

# catR function: make an ItemBankObject needed in the nextItem()
function
catBank<-createItemBank(itemBank, model="2pl")

### CHOOSE THE STARTING ITEM
current_item <- order(abs(difficulty))[1] ## choose the item with
difficulty closest to 0

### Once you've chosen an item to show, you have to let HTML layer
### know what it has to show. Set a and b variables to appropriate
### content to be shown in the {{a}} and {{b}} fields
set.var("a", a[current_item])
```

```

set.var("b", b[current_item])

### DEFINE SOME VARIABLES FOR LATER
items<-current_item #item history
responses<-NULL      #responses history

### Now, let's fill the fields you've added to the second template
set.var("difficulty", difficulty[current_item])
set.var("discrimination", discrimination[current_item])
set.var("theta", "not available")
set.var("sem", "not available")

### Concerto function: Now select the second template
### PUT SECOND TEMPLATE'S ID IN THE BRACKETS (you can find
### the ID on the template's list on the left)
set.next.template(ENTER ID OF THE NEXT TEMPLATE HERE)

```

Za drugi šablon, na kome se nalazi pitanje i jedno dugme za potvrđivanje odgovora, kod je sledeći:

```

### "response" variable is provided to R by HTML Layer
### (Remember the text field called "response?")
### However, it comes as a string, better change it to number
response <- as.numeric(response)

### If user gave no response, set it to -99
if (is.na(response)){response <- -99}

### add current response to the response history
responses<-c(responses, response)

### Make a vector of correct/incorrect responses (with 1 or 0
values) ### using response history and vector of correct responses
correct<-as.numeric(correct.response[items]==responses)
#### catR magic ####
# (1) select from the item bank only the items administered so far
it<-matrix(itemBank[items, ], nrow=length(items))

# estimate theta using thetaEst (catR) function
# it is the matrix of parameters of items that were administered so
far
# correct is the vector with 0/1 values computed before
# catBank is the ItemBankObject generated on the previous item
# other parameters are just defaults, check catR manual for details
theta<-thetaEst(it, correct, D=1, method="BM", priorDist="norm",
priorPar=c(0,1), range=c(-4,4), parInt=c(-4,4,33))

# estimate sem using semTheta (catR) function
sem<- semTheta(theta, it, correct, D=1, method="BM",
priorDist="norm", priorPar=c(0,1), parInt=c(-4,4,33))

# select next item using nextItem (catR) function
# note the [[1]] at the end of the function - it returns the whole
list
# of parameters, and we just need one here
current_item <- nextItem(catBank, theta, out=as.numeric(items),
x=NULL, criterion="MFI", method="BM", priorDist="norm",
priorPar=c(0,1), D=1, range=c(-4,4), parInt=c(-4,4,33),
infoType="observed")[[1]]

# and the newly selected item to the item history
items<-c(items, current_item)

#### send item content to the HTML layer

```

```

set.var("a", a[current_item])
set.var("b", b[current_item])

### Now, let's fill the fields you've added to the second template
### format(x, digits=3) is used to make it look prettier
set.var("difficulty", difficulty[current_item])
set.var("discrimination", discrimination[current_item])
set.var("theta", format(theta, digits=3))
set.var("sem", format(sem, digits=3))

### STOPPING CONDITIONS
### Simple: if length is higher than limit of items or sem is lower
### than a border value, set next template to the third
### (feedback) template
### note that max_items and min_sem were set in the code assigned
### to the button on the previous template

if (length(items) > max_items || sem < min_sem) {
  set.next.template(---FEEDBACK TEMPLATE ID HERE---)
}

```

U oba koda, neophodno je samo izvršiti izmenu funkcije `set.next.template()`, kako bi odgovarala id-u odgovarajućeg šablona. Kod koji je potrebno da se dodeli odgovarajućim dugmadima može se naći u tutorijalu za *Concerta*, te nije neophodno poznavanje sintakse R programskog jezika, ali svakako da bi bilo poželjno da korisnik može da pročita kod i da razume šta se u njemu dešava. Nakon definisanja sva tri šablona, potrebno je proveriti da li postoje sintaksne greške koristeći naredbu `DEBUG`, a zatim i pokrenuti test sa `RUN`. Ukoliko je sve uspešno urađeno, test izgleda kao što je prikazano na slici 6.7.

Solve the equation below:

$10 + 9 =$

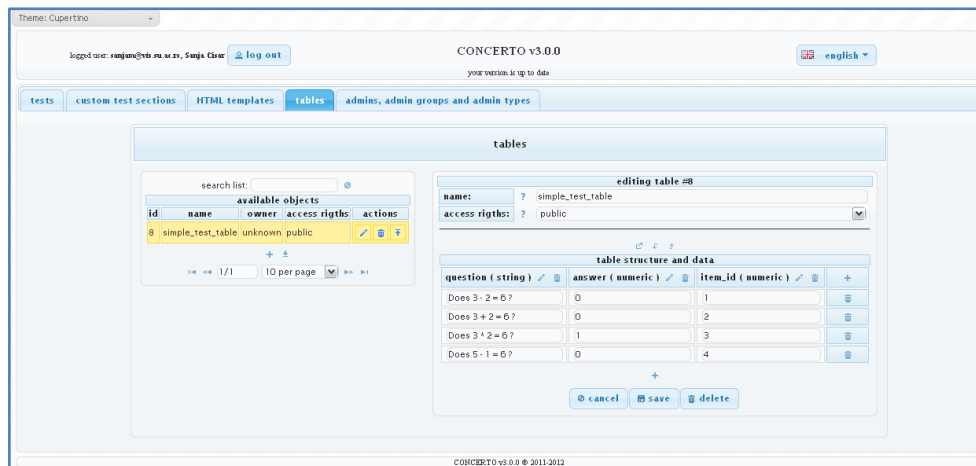
Current Theta: **not available** (SEM: **not available**); Current item: *difficulty: 0 | discrimination: 1.82*

*Note that in this example items and their parameters are randomly generated when you start the test.*

Slika 6.7: Prikaz pitanja

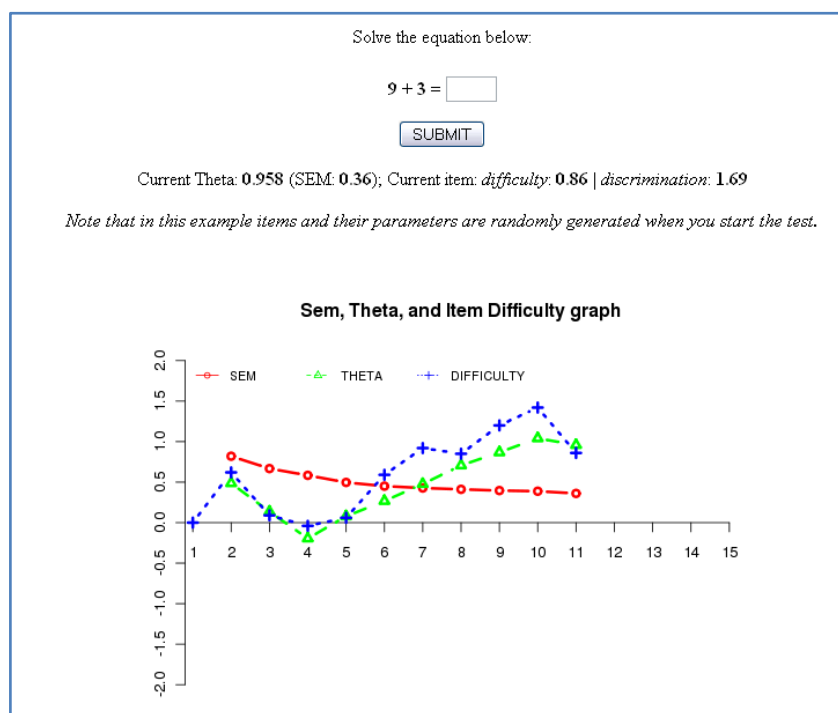
Upravljački panel *Concerta* ima tab **tables**, prikazan na slici 6.8, koji daje mogućnost pravljenja baze pitanja. Pitanja su dihotomnog tipa, gde vrednost 0 označava netačan odgovor a 1 tačan.





Slika 6.8: Tab tables za pravljenje baze pitanja

Programski jezik R ima ugrađene grafičke mogućnosti, te se može dodati jednostavan grafički prikaz uz svako pitanje, kako se menja SEM,  $\theta$  i težina pitanja u zavisnosti da li ispitanik daje tačan ili netačan odgovor. Ovo je ilustrovano na slici 6.9.



Slika 6.9: Grafički prikaz standardne greške merenja, procenjene sposobnosti i težine pitanja

Dodavanje grafikona zahteva da se u prvi šablon (*Introduction*) doda sledeći kod:

```
## fancy graph function
plot_graph<-function(){
  ### set the working directory where the graph will be save
  ### use the following for the demo installation of Concerto on our
  server
  ### IMPORTANT: this directory has to be visible from the web!
  setwd("/var/www/vhosts/dev.miyqtest.org/httpdocs/concerto/temp/")
}
```

```

### prepare the graph area
### x axis as long as max_items parameter (should be defined
### earlier in this script - check the previous tutorial
x_axis<-1:max_items

### y axis from -2 to 2... well, there must be a better way of
### doing it. Any ideas?
y_axis<-sample(c(-2,2),max_items, replace=T)

### get 3 colors
colors<-rainbow(3)

### Set the name for the graph
### IMPORTANT: each time the user clicks the file with the graph
### has to change its name. Otherwise, the browser will not
refresh
### the picture. The file with the graph will be called
### SessionID_item.png
### where "SessionID" is the id of a current session and item is
### the number of items already administered
file.name<- paste(SessionID, "_", length(items), ".png", sep="")
### send the item name to HTML layer:
set.var("image_source", file.name)

### set the output to the png file
png(filename = file.name, width = 600, height = 450, pointsize =
12, res=80, bg = "white") #quality = 100,

#### set up plot area
plot(x_axis, y_axis, type="n", axes = F, xlab="responses",
ylab="", frame.plot=F)

### add axes
axis(1, 1:max_items, pos=0)
axis(2, -4:4/2, pos=0.8)

### add a title and subtitle
title("Sem, Theta, and Item Difficulty graph")

### add a legend
legend(1, 2, c("SEM", "THETA", "DIFFICULTY"), cex=0.8, col=colors,
pch=1:3, lty=1:3, bty="n", horiz =T)

### PLOT LINES
lines(1:length(items), xsem, type="b", lwd=3, lty=1,
col=colors[1], pch=1)
lines(1:length(items), xtheta, type="b", lwd=3, lty=2,
col=colors[2], pch=2)
lines(1:length(items), difficulty[items], type="b", lwd=3, lty=3,
col=colors[3], pch=3)

### close the connection to the png file
dev.off()
#### END OF THE plot_graph FUNCTION
}

### Add variables to store theta and sem history
xtheta<-NA      #define theta history
xsem<-NA        #define sem history

### run the function
plot_graph()

```

Potrebno je izvršiti izmenu i drugog šablona (*Test questions*), tako da se doda slika i kod koji definiše gde se slika nalazi, kao i definisati širinu i visinu slike. U kod koji se odnosi na drugi šablon potrebno je da se doda sledeći kod:

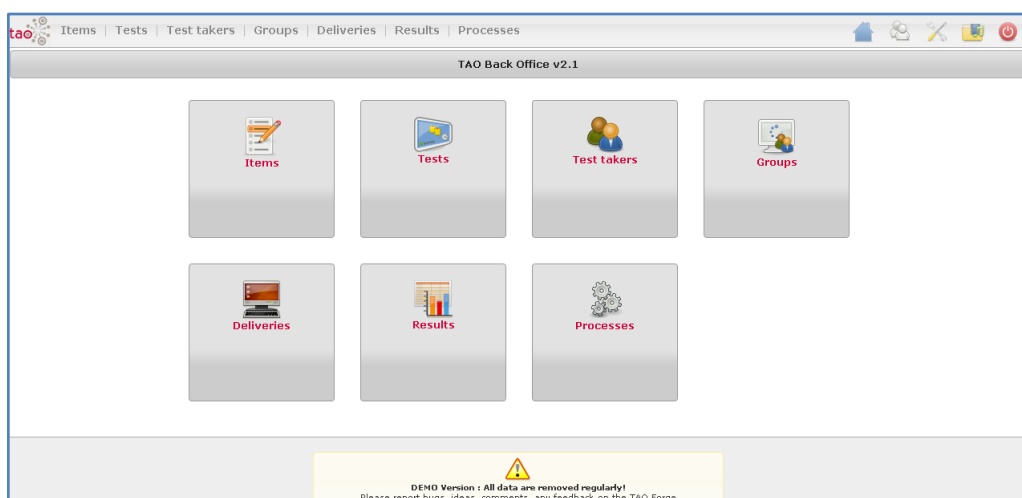
```
## update theta and sem history
xtheta<-c(xtheta,theta)
xsem<-c(xsem,sem)

## plot!
plot_graph()
```

### 6.1.2. TAO

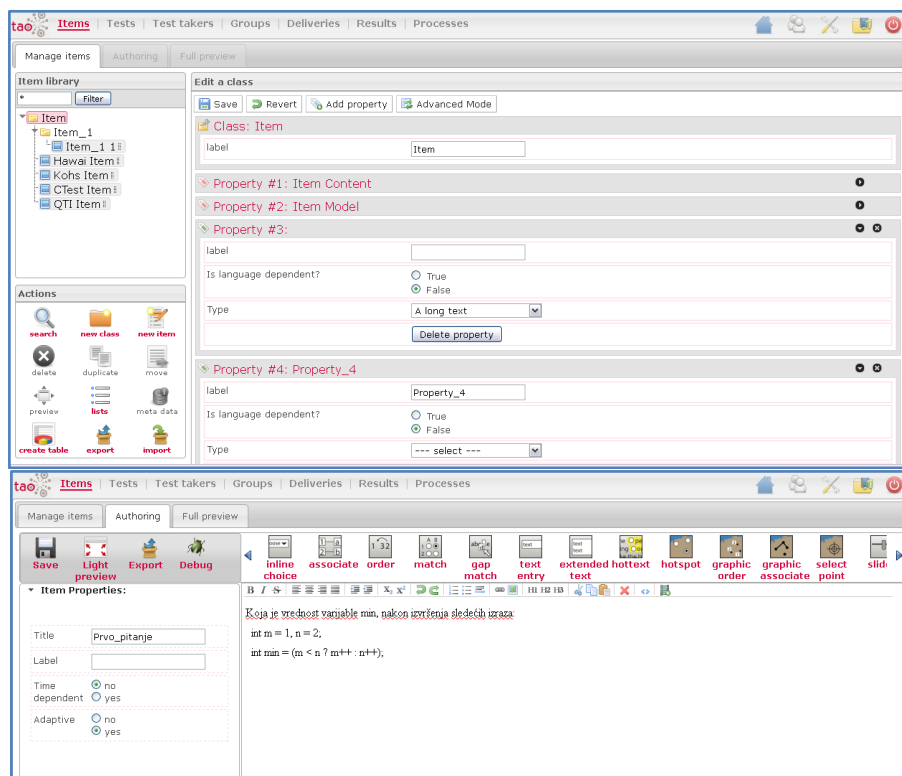
TAO je francuski akronim za *Testing Assisté par Ordinateur*, što označava testove koji se rade na računaru (engl. *Computer Based Testing*). TAO je platforma otvorene arhitekture za razvoj i isporuku testova koji se rade na računaru, sa mogućnošću definisanja pitanja sa višestrukim odgovorima (engl. *Multiple Choice Questions*), pitanja otvorenog tipa (engl. *Open Ended Questions*), popunjavanja praznina (engl. *Fill-in-the-gap*) kao i testove čiji je akronim Hawaii (engl. *Hyper Adaptive Work Area Item*) što je za sada u eksperimentalnoj fazi. TAO je zajednički projekat Univerziteta Luksemburg (engl. *the University of Luxembourg*) i Centra za javna istraživanja Henri Tudor (engl. *the Centre de Recherche Public Henri Tudor*). Testiranje je moguće obaviti na bilo kom operativnom sistemu (PC, Mac OSX, Linux) i koristeći bilo koji web čitač. Postoji mogućnost *online*, kao i *offline* testiranja. Da bi TAO platforma mogla da se koristi, neophodno je zadovoljiti sledeće tehničke karakteristike: Apache (preporučena verzija 2.2.9), PHP 5.3 i MySQL 5.0.

TAO je dizajniran kao modularni sistem koji čini šest modula: **Items** (pitanja), **Tests** (testovi), **Test takers** (ispitanici), **Groups** (grupe), **Delivery** (isporuka) i **Results** (rezultati), kao što se može videti na slici 6.10.



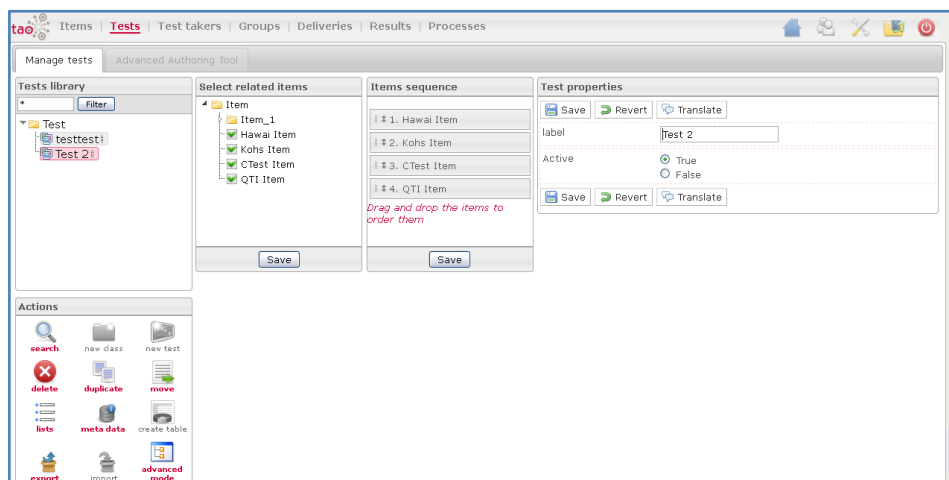
Slika 6.10: Glavni panel TAO-a

Modul **Item** omogućava definisanje pitanja uz mogućnost dodavanja multimedijalnih elemenata (slika, zvuk, video, animacija, interakcija). Podržano je više tipova pitanja, kao na primer pitanja sa više ponuđenih odgovora od kojih je samo jedan tačan, više tačnih odgovora, dihotomna pitanja, uparivanje pojmova, *puzzle*, popunjavanje praznina. Za pitanje je moguće podesiti da li postoji vremensko ograničenje koje stoji na raspolaganju ispitaniku za davanje odgovora. Izgled modula **Item** prikaz je na slici 6.11.



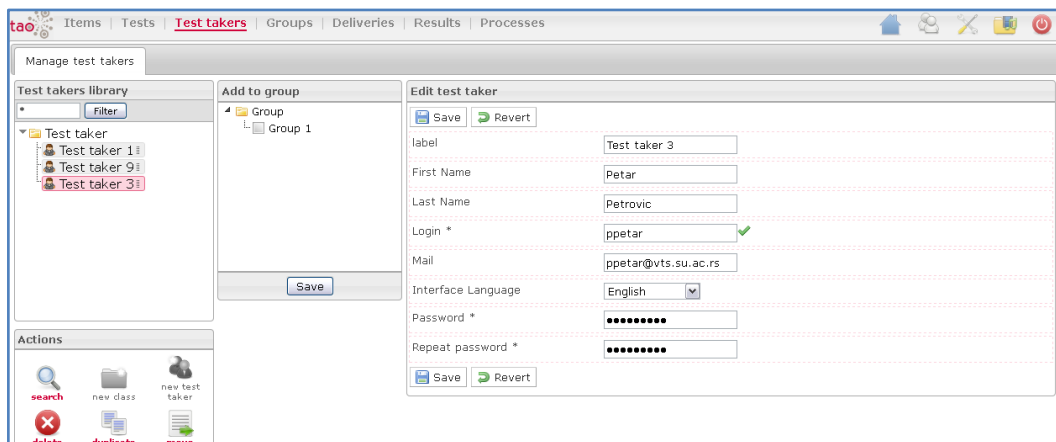
Slika 6.11: Modul Item u kome se definišu pitanja

Modul **Tests** koristi se za definisanje redosleda pitanja u testu, bodovanja i pregled izgleda testa u toku faze dizajna. Njegov izgled prikazan je na slici 6.12.



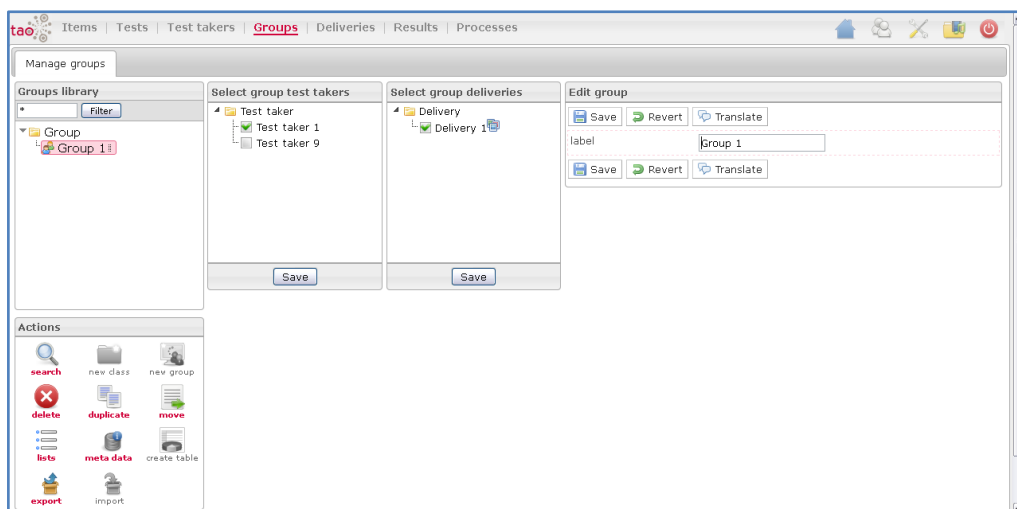
Slika 6.12: Izgled modula Tests u kome se definišu pitanja koja čine test

Modul **Test takers** koristi se za definisanje korisničkih naloga ispitanika i definisanje grupa korisnika prema predmetima, kao što je prikazano na slici 6.13.



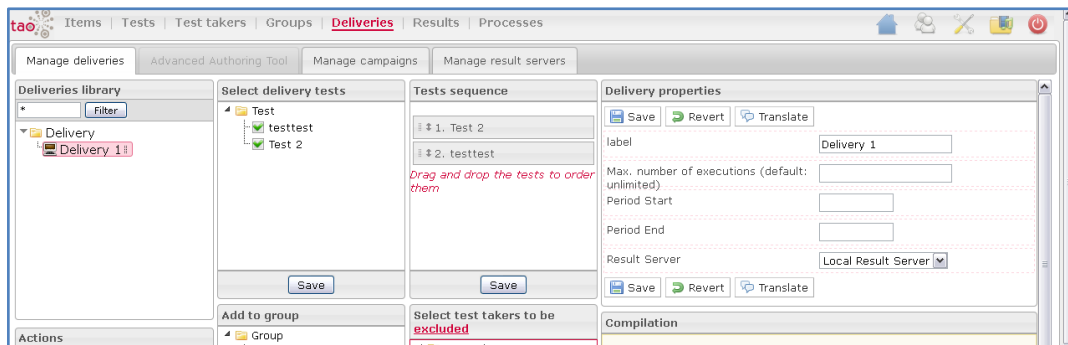
Slika 6.13: Modul za definisanje korisničkih naloga

Modul **Groups** dodeljuje testove grupama ispitanika koji mogu da polažu određeni test, što ilustruje slika 6.14.



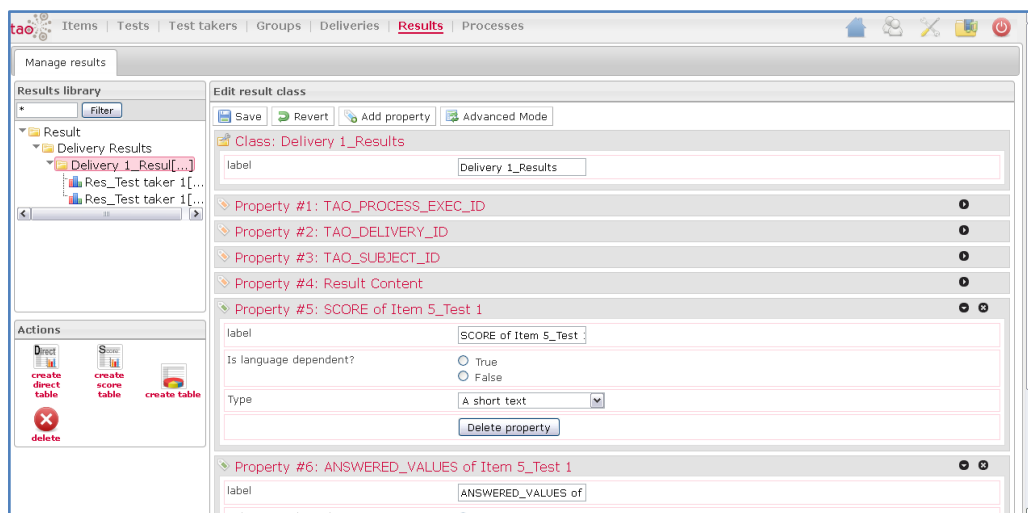
Slika 6.14: Modul Groups

Modul **Deliveries** omogućava definisanje vremena od kada je test aktivan, koliko dugo (vremenski) je dostupan za polaganje, kao i maksimalan broj pokušaja koje ispitanik ima za polaganje testa. Izgled modula prikazuje slika 6.15.



Slika 6.15: Definisane periode od kad do kad je test aktivan

Modul **Results** (slika 6.16) omogućava pravljenje tabelarnog prikaza rezultata, snimanje rezultata u csv dokumentima (excel, open office) i pravljenje izveštaja.

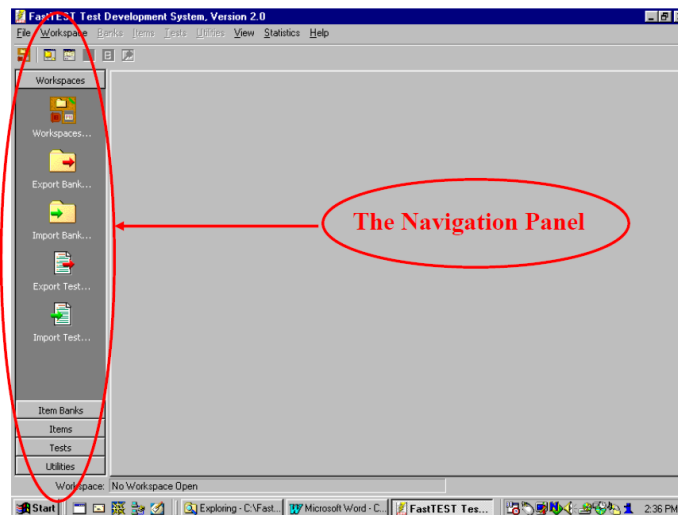


Slika 6.16: Definisane načina prikaza rezultata testa

### 6.1.3. FastTEST

Aplikacija *FastTEST Web* nudi mogućnost *online* testiranja, kao i izradu testova za klasičan oblik testiranja. Nastala je kao rezultat saradnje dve organizacije *Assessment Systems Corporation* (<http://www.assess.com>) i *4ROI* ([www.4roi.com](http://www.4roi.com)). Pored mogućnosti realizovanja klasičnih testova provere znanja ugrađena je podrška i za adaptivno testiranje (<http://www.fasttestweb.com>) koje se bazira na IRT modelu. Aplikacija *FastTEST* ima višezjezičnu podršku (za sada nema za srpski jezik) i mogućnost provere pravopisa za sledeće jezike engleski (američki i britanski), nemački, holandski, francuski, španski, češki i italijanski, kao i rečnik engleskog jezika za medicinske i pravne pojmove. Aplikacija radi pod Windows operativnim sistemom (Windows 95, 98, NT, ME, 2000, XP, Vista, Windows 7).

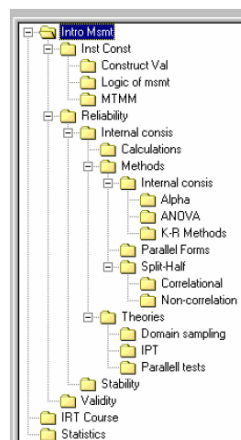
Glavni prozor *FastTEST Web*-a je navigacioni panel, koji omogućava pristup radnom prostoru, bazi pitanja, pitanjima i testovima, kao što je prikazano na slici 6.17.



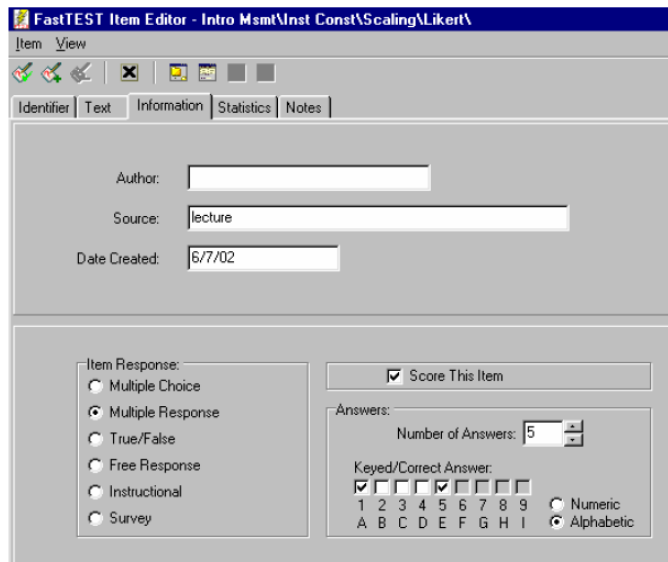
Slika 6.17: Navigacioni panel [188]

Može se definisati više radnih prostora i svaki od njih zaštititi lozinkom, kako bi se zaštitio pristup bazi pitanja i testovima. Nakon stvaranja radnog prostora, sledeći korak je kreiranje baze pitanja (*Item bank*) iz koje će se birati pitanja za test. Baza pitanja može da ima hijerarhijsku strukturu do 10 nivoa, sa različitim kategorijama na svakom nivou, kako što je prikazano na slici 6.18.

Sledeći korak je definisanje pitanja u bazi pitanja. Pitanja se mogu unositi koristeći editor teksta koji postoji u aplikaciji FastTEST. Zatim, ukoliko je u pitanju manji broj pitanja, svako pojedinačno pitanje može se snimiti kao RTF fajl i uvesti u bazu. Za veći broj pitanja koristi se opcija uvoza pitanja iz nekog tekst procesora. FastTEST omogućava da se pored teksta postavi i slika uz pitanje. Podržane su sledeće ekstenzije: .JPG, .BMP, .EMF, .ICO i .WMF. Za svako pitanje potrebno je definisati koji je tip pitanja, i u zavisnosti od tipa pitanja uneti tačan odgovor. Na slici 6.19 je prikazan primer pitanja sa pet ponuđenih odgovora od kojih su dva tačna.

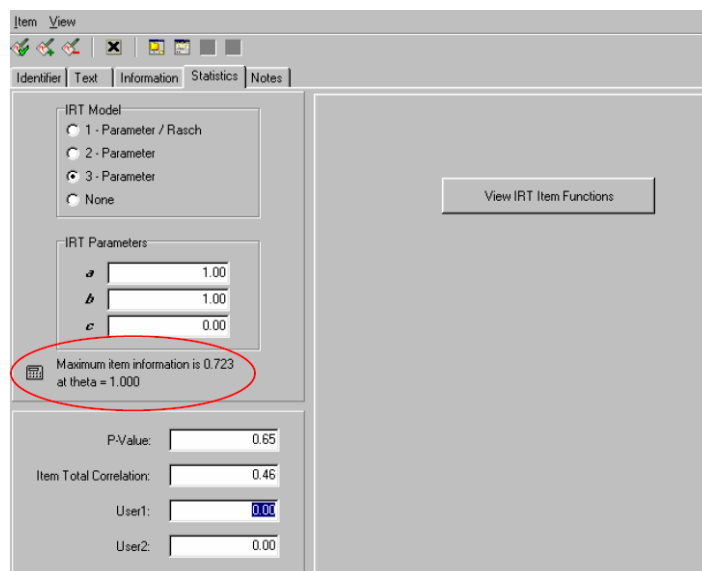


Slika 6.18: Baza pitanja sa podkategorijama [188]



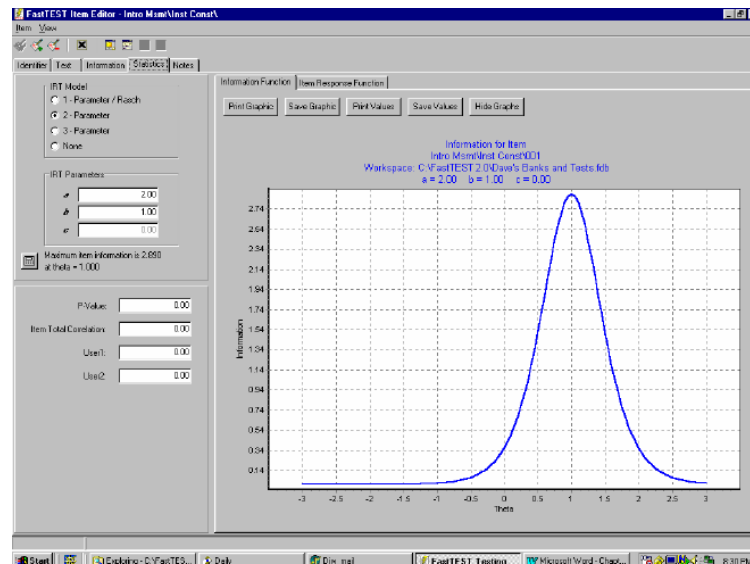
Slika 6.19: Editor pitanja [188]

Tab **Statistics** daje mogućnost da se odabere da li će test da bude adaptivan ili klasičan, kao što se može videti na slici 6.20. U slučaju adaptivnog testa, koji se bazira na IRT modelu, bira se da li se primenjuje jednoparametarski (1PL), dvoparametarski (2PL) ili troparametarski model (3PL), kao i parametri procene  $a$ ,  $b$  i  $c$ . Detaljnije o ovim modelima i parametrima procene pisano je u poglavlju 5. Za unete parametre proračunava se maksimalna vrednost funkcije informacije i individualni nivo sposobnosti  $\theta$ . Kada se odabere IRT model koji će se primeniti, postaje aktivno dugme *View IRT Item Functions*, koje omogućava grafički prikaz funkcije informacije pitanja (slika 6.21).



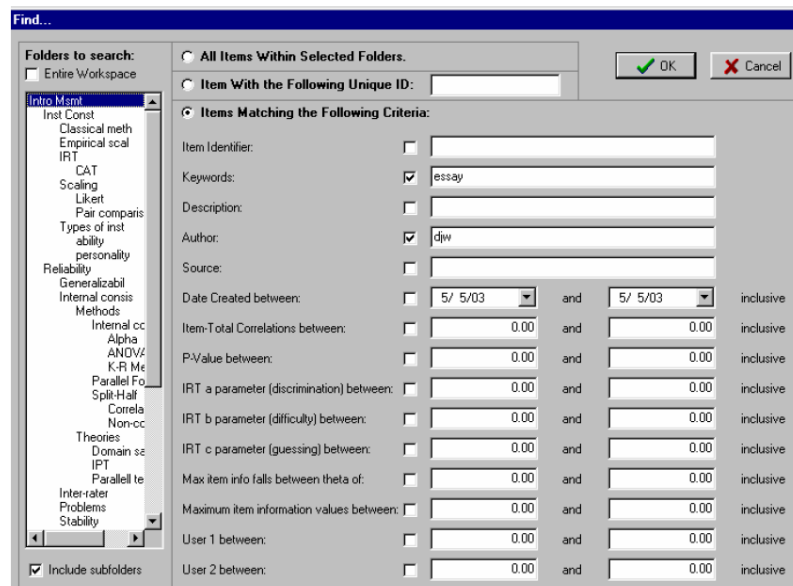
Slika 6.20: Tab Statistics [188]



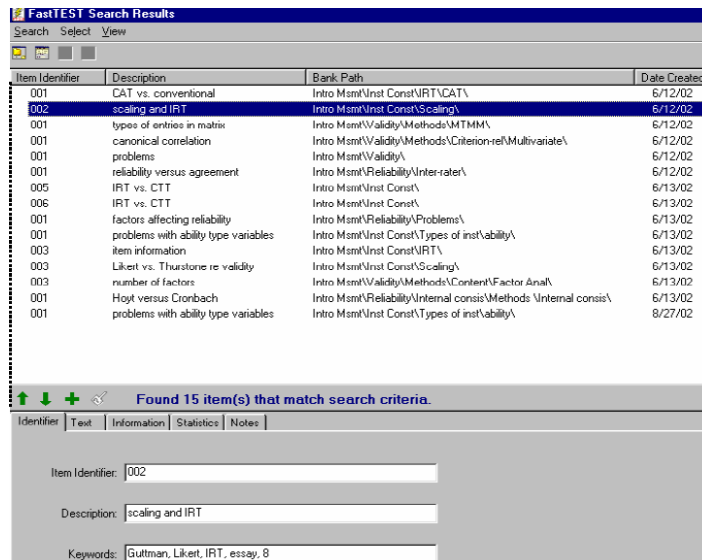


Slika 6.21: Grafički prikaz funkcije informacije pitanja za odabrani IRT model

Sledeći korak u izradi testa je odabir pitanja za test, što se započinje odabirom taba **Test** u navigacionom panelu. Nakon definisanja imena testa i kratkog opisa (najviše 80 karaktera), sledi izbor pitanja tako što se iz menija odabere *Items*. Pitanja iz odabrane baze pitanja i kategorije, mogu se birati pojedinačno jedno po jedno, ili se može odabrati opcija *Random* uz definisanje broja pitanja koje je potrebno odabrati za test. Pretraga pitanja može da bude rađena u jednoj bazi pitanja, ili više, kao i po više kategorija, kao što se može videti na slici 6.22,. Rezultat pretrage za zadati kriterijum prikazan je na slici 6.23.



Slika 6.22: Pretraga pitanja [188]

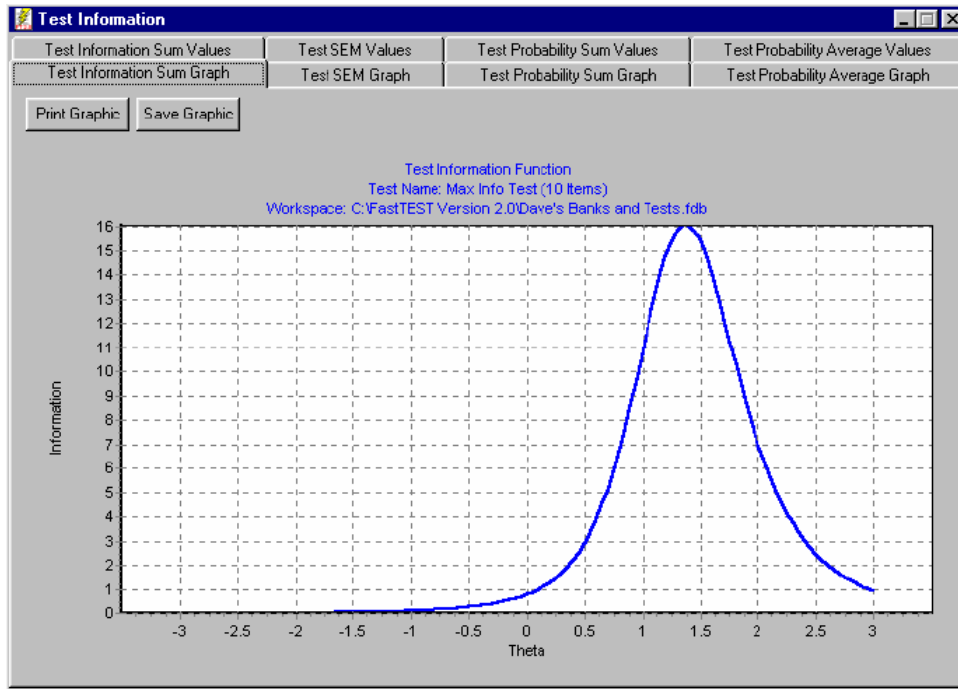


Slika 6.23: Rezultat pretrage za zadati kriterijum [188]

U slučaju adaptivnog testa, izbor pitanja može da se radi tako da bude zadovoljen kriterijum da test daje određenu vrednost funkcije informacije testa. Na primer, želi se napraviti test koji ispitanike klasifikuje u dve kategorije pao/položio i definiše se granična vrednost  $\theta=1.5$  koja čini granicu između ove dve kategorije. Definiše se opseg za  $\theta$ , kao što se vidi na slici 6.24. Rezultat pretrage daće sva pitanja koja se nalaze u bazi pitanja koja zadovoljavaju kriterijum da je maksimum informacije pitanja između vrednosti 1.25 i 1.75. Dalje, za odabrana pitanja računa se funkcija informacije testa, standardna greška merenja i funkcija odgovora testa, a moguće je videti i grafički prikaz ovih proračuna (slika 6.25).

Max item info falls between theta of:	<input checked="" type="checkbox"/>	<input type="text" value="1.25"/>	and	<input type="text" value="1.75"/>	inclusive
Maximum item information values between:	<input type="checkbox"/>	<input type="text" value="0.00"/>	and	<input type="text" value="0.00"/>	inclusive

Slika 6.24: Definisavanje opsega za  $\theta$



Slika 6.25: Grafički prikaz funkcije informacije testa za zadati opseg vrednosti  $\theta$  [188]

Za odabrana pitanja koja čine adaptivni test moguće je uraditi i statistiku pitanja koja je prikazana na slici 6.26.

Item No.	Proportion Correct	Item-Total Correlation	IRT Parameters			User 1	User 2	Max. Info.	At Theta
			a	b	c				
1	0.5000	0.4500	1.0000	0.0000	0.0000	0.66	0.31	0.72	0.0
2	0.4700	0.5400	2.0000	1.2500	0.2000	0.35	0.55	1.97	1.3
3	0.8100	0.4700	1.5000	0.0250	0.1700	0.12	0.76	1.17	0.1
4	0.2200	0.4700	1.0000	-1.2500	0.1800	0.76	0.67	0.51	-1.1
5	0.4700	0.4400	2.5000	-1.7500	0.2100	0.62	0.26	3.02	-1.6
6	0.5100	0.5500	1.5000	1.2900	0.1600	0.87	0.78	1.20	1.3
7	0.6500	0.4600	1.0000	1.0000	0.4000	0.37	0.73	0.33	1.2
8	0.7100	0.3400	1.0000	1.4400	0.0000	0.63	0.36	0.72	1.4
9	0.4700	0.2200	1.6600	1.3300	0.1000	0.56	0.65	1.64	1.3
10	0.3700	0.5400	2.3000	-2.2200	0.2200	0.72	0.27	2.51	-2.1
11	0.7400	0.4700	1.5000	1.2700	0.1700	0.47	0.74	1.17	1.3
12	0.8100	0.5600	1.0000	1.6000	0.1800	0.65	0.56	0.51	1.7
13	0.2700	0.6100	2.1100	1.2100	0.2100	0.77	0.66	2.15	1.2
14	0.1750	0.4500	1.7500	1.7500	0.1750	0.45	0.45	1.58	1.8
15	0.2500	0.5200	1.0000	0.2500	0.2600	0.76	0.54	0.44	0.4
16	0.1750	0.1750	1.7500	1.7500	0.1750	0.17	0.17	1.58	1.8
17	0.7500	0.7500	3.0000	1.6000	0.0000	0.77	0.66	6.50	1.6
18	0.4500	0.4400	2.5000	1.4000	0.0000	1.20	-1.20	4.52	1.4
Min	0.1750	0.1750	1.0000	-2.2200	0.0000	0.12	-1.20		
Max	0.8100	0.7500	3.0000	1.7500	0.4000	1.20	0.78		
Mean	0.4889	0.4697	1.6706	0.6636	0.1561	0.61	0.44		
S.D.	0.2175	0.1315	0.6249	1.2377	0.1043	0.26	0.45		

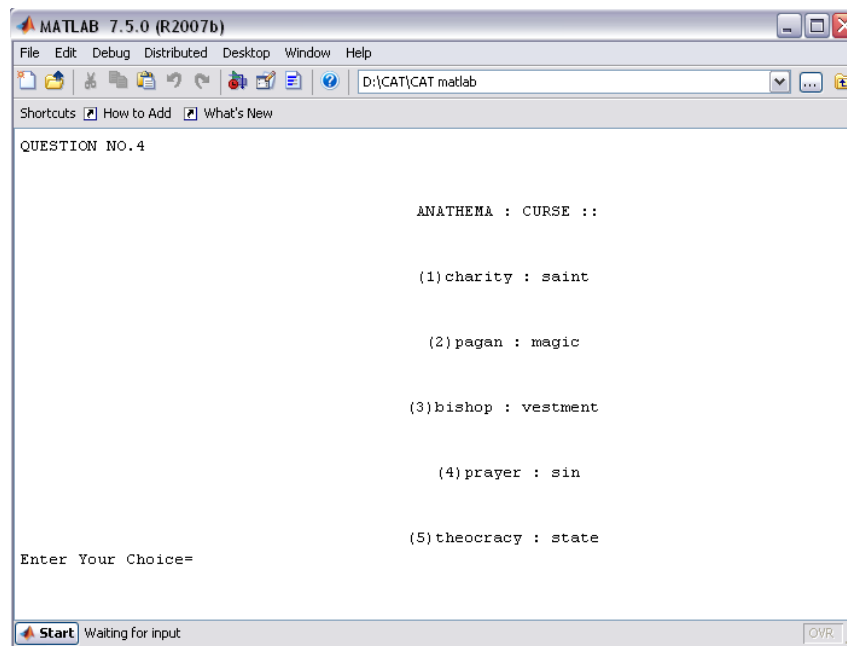
Slika 6.26: Statistika pitanja prema parametrima  $a$ ,  $b$  i  $c$  [188]

## 6.2. Odabrano rešenje

I pored postojanja opisanih aplikacija koje imaju mogućnost adaptivnog testiranja, autor se opredelio za sopstveno rešenje. Naime, aplikacija *Concerto* ima mogućnost postavljanja samo pitanja otvorenog tipa, što nije odgovaralo vrsti testa koji se želeo sprovesti. Takođe, neophodno je savladati osnovne veštine programskog jezika R što bi svakako uticalo na produženje vremenskog perioda koje je planirano za istraživanje, što je bilo neprihvatljivo. Aplikacija *FastTEST Web* nudi brojne mogućnosti u pogledu izrade adaptivnog testa, međutim nije u potpunosti besplatna. Verzija koja daje mogućnost *online* testiranja do 200 ispitanika košta 1500\$ za godinu dana. Moguće je koristiti demo verziju aplikacije besplatno u periodu od 30 dana, ali to je bio nedovoljno dug vremenski period da bi ispitivač ovladao korišćenjem ove aplikacije, da se obuče ispitanici kako se aplikacija koristi i da se sprovede samo istraživanje. Na kraju, ostaje aplikacija TAO za koju bi sigurno bilo potrebno nekoliko meseci da se ovlada tehnikom kreiranja adaptivnog testa, kao i jednoparametarskim, dvoparametarskim i troparametarskim modelom kako bi se na adekvatan način mogli primenjivati u izradi testa.

Za potrebe utvrđivanja efekata primene računarskog adaptivnog testa za proveru znanja realizovan je adaptivni test u programskom paketu MATLAB (skraćenica od MATrix LABoratory) verzija 7.10 (R2010a, dostupan od 5.marta 2010. godine). MATLAB je okruženje za numeričke proračune firme MathWorks [181]. Iako osnovna namena MATLAB-a nije pravljenje konzolnih aplikacija, odabran je za realizaciju računarskog adaptivnog testa jer ga studenti koriste tokom nastave iz predmeta Osnovi računarstva, koji je obavezan za sve studente u prvom semestru. Autor je smatrao da će radno okruženje koje je poznato svim studentima doprineti pozitivnom stavu i smanjenju napetosti tokom testa. U prvom semestru studenti imaju i predmet Računarska laboratorija I, koji ima za cilj osposobljavanje studenata za ovladavanje osnovnim veštinama rada na računaru, te je time smatrano da je ispunjen uslov da svi studenti poseduju potrebne veštine da na adekvatan način koriste računarski adaptivni test.

Računarski adaptivni test koji je korišćen u eksperimentalnom delu rada predstavlja modifikaciju adaptivnog testa koji može da se nađe na web adresi [182]. Originalni test predstavlja adaptivnu verziju GRE (engl. *Graduate Record Exam*) testa i pokreće se u komandnom prozoru MATLAB-a. Ispitanik odgovor daje ukucavanjem slova sa tastature koji stoji pored odgovora za koji smatra da je tačan, kao što se može videti na slici 6.27.

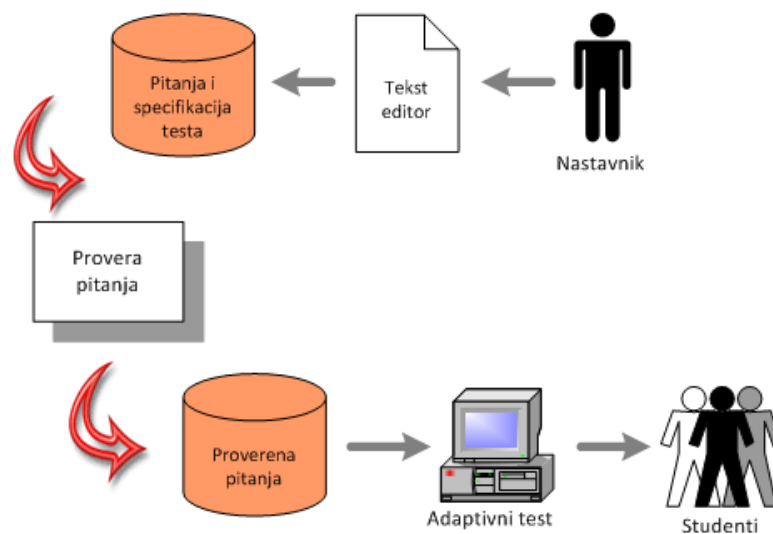


Slika 6.27: Adaptivna verzija GRE testa napravljena u MATLABu

Postojeća aplikacija je modifikovana kako bi odgovarala potrebama kursa C++ i Java. Test sadrži pitanja višestrukog izbora s pet ponuđenih odgovora. Realizovan je i odgovarajući GUI koji prati ovu aplikaciju [64]. Primenjen je i novi algoritam selekcije pitanja. Naime, kod adaptivnih testova ukoliko ispitanik tačno odgovori na pitanje, pitanje koje sledi bira se i grupe pitanja koja su za jedan nivo veće težine od pitanja na koje je ispitanik tačno odgovorio. Analogno tome, ako je odgovor na postavljeno pitanje netačan, pitanje koje se sledeće postavlja je za jedan nivo manje težine. Kod adaptivnog testa ne postoji mogućnost vraćanja na prethodno postavljena pitanja i menjanje prethodno datih odgovora. Novi algoritam koji je primenjen u aplikaciji za adaptivno testiranje, bazira se na tome da se tek nakon dva uzastopno data netačna odgovora na istom nivou težine prelazi na pitanje za jedan nivo manje težine. Dakle, ako je odgovor netačan, ne menja se nivo težine pitanja. Nakon tačnog odgovora sledi pitanje iz grupe za jedan nivo težih pitanja. Na primer, neka su pitanja podeljena u pet grupa prema težini (1 laka, 5 teška pitanja). Niz odgovora 2F, 2T, 3T, 4F, 4F, 3F, 3F, 2T, 3T, 4T, 5F ...znači da je ispitanik netačno odgovorio na prvo pitanje težine 2, nakon čega je dato sledeće pitanje sa istog nivoa težine 2, na koje je dat tačan odgovor. Kako je odgovor na pitanje težine 2 bio tačan, prelazi se na nivo težine pitanja 3 i ispitanik daje tačan odgovor. Sledeće pitanje je iz grupe pitanja težine 4 na koje ispitanik nije znao tačan odgovor. Stoga se zadržava na istom nivou težine pitanja, te je pitanje koje sledi ponovo težine 4. Odgovor nije bio tačan, te kako su bila dva uzastopno netačna odgovora, nivo težine pitanja se smanjuje i sledi pitanje težine 3. Odgovor je bio netačan, nivo težine pitanja se zadržava, dakle sledi pitanje težine 3 na koje je dat netačan odgovor. Zbog dva uzastopno data netačna odgovora nivo težine pitanja se smanjuje, prelazi se nivo 2. Usledio je tačan odgovor, nivo težine pitanja se povećava na 3. Tačan odgovor na ovo pitanje menja nivo težine pitanja na 4, tačan odgovor dovodi do promene nivoa pitanja na 5, itd.

Autor je svestan nedostataka primenjenog rešenja u pogledu HCI, ali prioritet pri razvoju aplikacije bila je brzina učenja. Kako je u pitanju veliki broj korisnika, bilo bi potrebno utrošiti dosta vremena na njihovu obuku, te se pristupilo minimalističkom dizajnu u radnom okruženju poznatom svim studentima (MATLAB). Cilj je bio jednostavnost obuke, nizak nivo mogućih grešaka od strane korisnika, jednostavan interfejs za upotrebu bez dodatnih ikona i dugmića kako bi se korisnik fokusirao na postavljeno pitanje. U daljim fazama razvoja aplikacije treba svakako uzeti u obzir i usavršavanje aplikacije prema HCI principima.

Na slici 6.28 prikazana je arhitektura sistema. Nastavnik pitanja piše u nekom od tekst editora na unapred definisan način. Za ovo je dovoljna osnovna računarska pismenost i poznavanja editora teksta (npr. Notepad ili MS Word). Potrebno je prvo zadati tekst pitanja, listu od pet mogućih odgovora, a zatim i sam tačan odgovor na postavljeno pitanje. Pitanja za test podeljena su prema težini u tri klastera (laka, srednje teška i teška pitanja), tako da ukupno ima tri tekstualna fajla koja sadrže pitanja. Sva pitanja su proverena i ona pitanja koja su prošla proveru čine bazu proverenih pitanja. Iz baze proverenih pitanja zadaju se pitanja kada se pokrene računarski adaptivni test. U nastavku sledi objašnjenje kalibracije pitanja, zadavanje početnog pitanja, izbor sledećeg pitanja i zaustavljanje testa.



Slika 6.28: Arhitektura sistema

### 6.3. Kalibracija pitanja

Za svako pitanje koje se nalazi u bazi pitanja potrebno je izvršiti kalibrisanje. Prema [72] postoje tri načina kalibrisanja pitanja: konvencionalno, ekspertska i *online*.

**Konvencionalna kalibracija** uključuje metode kao što su procena maksimalne pridružene verovatnoće (engl. *joint maximum likelihood*, JML), maksimalna uslovna verovatnoća (engl. *conditional maximum likelihood*, CML) i marginalna maksimalna

verovatnoća (engl. *marginal maximum likelihood*, MML). Broj ispitanika koji su potrebni da bi se izvršila kalibracija parametara pitanja na ovaj način kreće od najmanje 1000 [99], [146], dok se u [60] preporučuje od 200 do 1000.

**Kalibracija eksperta** podrazumeva kalibraciju IRT parametara od strane stručnjaka za oblast iz koje se daju pitanja. U radu [30] opisana je kalibracija pitanja za računarski adaptivni test koji se bazira na 3-PL modelu.

**Online kalibracija** podrazumeva korišćenje odgovora ispitanika na prethodno kalibrisana pitanja kako bi se procenili parametri novih pitanja tokom testa [146].

Za formiranje baze pitanja odabrana je kalibracija eksperta kao najjednostavnija za realizaciju. Kalibracija pitanja bazirala se na Blumovoj taksonomiji kognitivnih veština [22]. Pri sastavljanju i odabiru pitanja za ovo istraživanje, vodilo se računa da se odaberu pitanja koja mogu da daju efikasnu procenu prve tri kognitivne sposobnosti Blumove taksonomije: znanje, razumevanje i primena. U tabeli 6.1 prikazana je podela pitanja prema kognitivnim sposobnostima.

Tabela 6.1: Težina pitanja prema kognitivnim sposobnostima

Težina	Sposobnost	Kratak opis
1	znanje	Sposobnost da se seti i/ili opozove ranije učeni materijal
2	razumevanje	Sposobnost da se tumači i/ili prevede ranije učeni materijal
3	primena	Sposobnost da se naučeno primeni u novim situacijama

Baza pitanja formirana je od pitanja koja su proverena u praksi (koja su bila na kolokvijumima i ispitu iz predmeta Objektno orijentisano programiranje i Java) u periodu od 2005. do 2010. godine, kao i nova pitanja koja su nastala u toku istraživanja. Baza je sadržala 210 pitanja za predmet Objektno orijentisano programiranje i 150 pitanja za Javu. Pitanja su definisali predmetni nastavnici ovih predmeta, a obuhvatala su gradivo koje je izučavano na časovima. Pitanjima su dodeljeni težinski koeficijenti prema tabeli 6.1, svrstavajući ih tako u tri klastera težine: 1 (laka), 2 (srednja) i 3 (teška). Pitanja koja su proverena u praksi sa ranijim generacijama studenata u godinama koje su prethodile samom eksperimentalnom istraživanju, klasifikovana su nakon statističke obrade procenta tačnosti odgovora studenata na njih. Nova pitanja nastala u toku eksperimentalnog istraživanja klasifikovana su prema subjektivnom ubeđenju autora. Postanaliza eksperimentalnih rezultata ukazala je na potrebu prekvalifikacije određenog broja pitanja iz jednog klastera u drugi. Pitanja za test pisala su se u .txt fajl, za svaki nivo težine pitanja postoji poseban fajl, dakle ukupno 3.

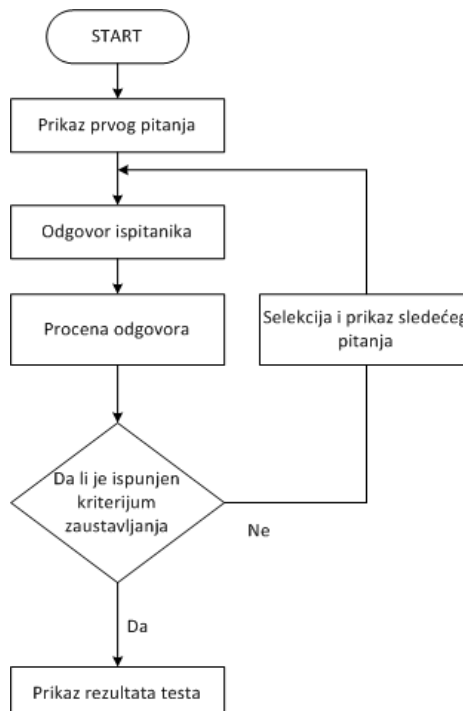
## 6.4. Početak testa

U računarski podržanom adaptivnom testu izbor sledećeg pitanja za administraciju zavisi od serije pitanja na koje je prethodno dat odgovor. Ostaje problem kako odabrati pitanje kojim će se započeti test, iako prema [85], ukoliko test nije isuviše kratak, loš izbor prvog pitanja ima mali uticaj na krajnji ishod testa.

Jedna od mogućnosti izbora prvog pitanja bila je da pitanje bude nasumice izabrano iz baze pitanja. Potencijalno ograničenje ovog izbora može da bude nasumični izbor sa jednog ili drugog kraja skale, tj. izbor suviše lakog, ili suviše teškog pitanja. Prema [28] pitanja koja se nalaze na krajevima skale težine smatraju se manje korisnim od pitanja srednje težine. Druga mogućnost jeste izbor pitanja na osnovu informacija koje postoje o ispitaniku, npr. njegov uspeh na prethodnom testu, ili uspeh postignut na nekom sličnom predmetu. Međutim, kako ovi podaci o ispitanicima nisu uvek dostupni, primenjen je pristup izbora pitanja srednje težine (klaster težine 2) kao pitanja kojim započinje test.

## 6.5. Izbor sledećeg pitanja za administraciju

Na slici 6.29 prikazan je dijagram toka adaptivnog testa.



Slika 6.29: Dijagram toka adaptivnog testa [108]

Test započinje pitanjem srednje težine (klaster 2). Primenjen je algoritam kod koga se tek nakon dva uzastopna netačna odgovora prelazi na pitanje iz klastera manje težine. Ukoliko ispitanik tačno odgovori na pitanje prelazi se na prvi sledeći nivo težine. Ukoliko je ispitanik odgovorio tačno, prelazi se na pitanje iz klastera 3. Ako je odgovor tačan, postavljaju se pitanja iz klastera 3 dok ispitanik ne da prvi netačan odgovor.



Kada ispitanik da netačan odgovor, pitanje koje sledi je iz klastera 3. Ako je odgovor tačan, zadržava se nivo težine pitanja (klaster 3), a ako je odgovor netačan prelazi se na pitanje iz klastera 2. Ako je odgovor na ovo pitanje tačan, prelazi se na pitanja iz klastera 3. Ako je odgovor netačan, nivo težine pitanja se zadržava i postavlja se sledeće pitanje iz klastera 2. Ako se dobiju dva uzastopno netačna odgovora na nivou težine 2, prelazi se na pitanja iz klastera 1. Test se zaustavlja kada se postavi unapred definisan broj pitanja, izračunava se rezultat i prikazuje se studentu.

Dakle, u najboljem slučaju kada ispitanik da tačne odgovore na sva postavljena pitanja, scenario bi bio sledeći: 2T, 3T, 3T, 3T, 3T, 3T, 3T, 3T, 3T, 3T, 3T, 3T, 3T, 3T, 3T, 3T, 3T, 3T, 3T. U najgorem slučaju, kada ispitanik da sve netačne odgovore, scenario je sledeći: 2F, 2F, 1F, 1F, 1F, 1F, 1F, 1F, 1F, 1F, 1F, 1F, 1F, 1F, 1F, 1F, 1F, 1F, 1F, 1F. Na primer, za sledeću sekvencu odgovora 2T, 3F, 3T, 3F, 3F, 2F, 2T, 3F, 3F, 2F, 2T, 3F, 3F, 2F, 2F, 1T, 2F, 2F, 1F, 1T vidi se da je na drugo pitanje dat netačan odgovor, nivo težine pitanja je zadržan i tek je nakon dva uzastopno data netačna odgovora promenjen, što se desilo na šestom pitanju.

## 6.6. Zaustavljanje testa

Računarski adaptivni test može da se završi kada se odgovori na fiksni, unapred određen broj pitanja, ili ako istekne unapred predviđeno vreme za test. Ovakva vrsta testova naziva se računarski adaptivni test fiksne dužine. Takođe je moguće da se adaptivni test završi kada se dostigne zadovoljavajući nivo preciznosti merenja, npr. kada standardna greška procene sposobnosti ispitanika dostigne prethodno definisan nivo. Ovakva vrsta testova naziva se računarski adaptivni test promenljive dužine. Moguće je definisati i pravilo zaustavljanja koje predstavlja kombinaciju dva pravila, npr. da se test završi kada ispitanik odgovori na sva pitanja predviđena za administraciju, ili kada istekne vreme određeno za rešavanje testa, u zavisnosti koji od ova dva uslova se prvi ispuni.

U radu [99] ukazuje se na to da u slučaju testa promenljive dužine, ispitanici slabijih sposobnosti rešavaju znatno kraće testove nego oni koji su sposobniji, dok se u radu [53] ukazuje na činjenicu da ispitanici sa sumnjom gledaju na testove koji kratko traju. Različita dužina testa izaziva sumnju kod ispitanika u pravičnost ocenjivanja.

Računarski adaptivni testovi fiksne dužine jednostavniji su za implementaciju, pored toga adaptivni testovi koji koriste fiksni broj pitanja kao pravilo zaustavljanja omogućavaju ispitivaču da predvidi koliko je pitanja potrebno da bude u bazi pitanja. Prema [26] preporučuje se tri do četiri puta veći broj pitanja od broja pitanja koji se administrira na testu, a prema [99] broj pitanja u bazi trebalo bi da bude veći 5 do 10 puta od broja pitanja koja se postavljaju u testu.

Imajući u vidu sve prethodno navedeno, odabran je test fiksne dužine trajanja gde je za pravilo zaustavljanja odabrano da je ispitanik odgovorio na 20 pitanja, uz ograničenje da za to ima na raspolaganju maksimalno 30 minuta.

## 6.7. Pojašnjenje programskog koda

U nastavku sledi objašnjenje pojedinih delova koda. Deo programskog koda dat je u Prilogu.

Pri pokretanju programa pored osnovnog prozora, prikazuje se i dijalog za unos osnovnih podataka o studentu, a potom se fokus prebacuje na dugme *Početak testa* kako bi samo *Enter* bio dovoljan da korisnik pređe na izradu testa (slika 6.30).

```
try
    [ime prezime br_indeksa]=unos_podataka;
    set(figura, 'Name', ['Student: ', ime, ' ', prezime, ',
    Br.indeksa:', br_indeksa])
end
uicontrol(pocetak_testiranja)
```



Slika 6.30: Izgled početnog ekrana aplikacije za unos osnovnih podataka o studentu

Ovaj dijalog se zapravo dobija pozivanjem funkcije `unos_podataka` koja se nalazi u posebnoj fajlu `unos_podataka.m`.

Da se na izradu testa ne bi prešlo pritiskanjem bilo kog tastera na tastaturi, rešeno je u `KeyPressFcn` funkciji dugmeta `pocetak_testiranja` tako što se po registrovanju aktiviranja tastera proverava da li je on `'return'`, odnosno da li je to bio `Enter` (ili `Spacebar`) ili je aktiviran neki drugi taster na koji ne treba da se reaguje.

```
function keypress(source, eventdata)
    keypressed=eventdata.Key;
    if strcmp(keypressed, 'return')
        grupa_pitanja=ones(1,20);
```

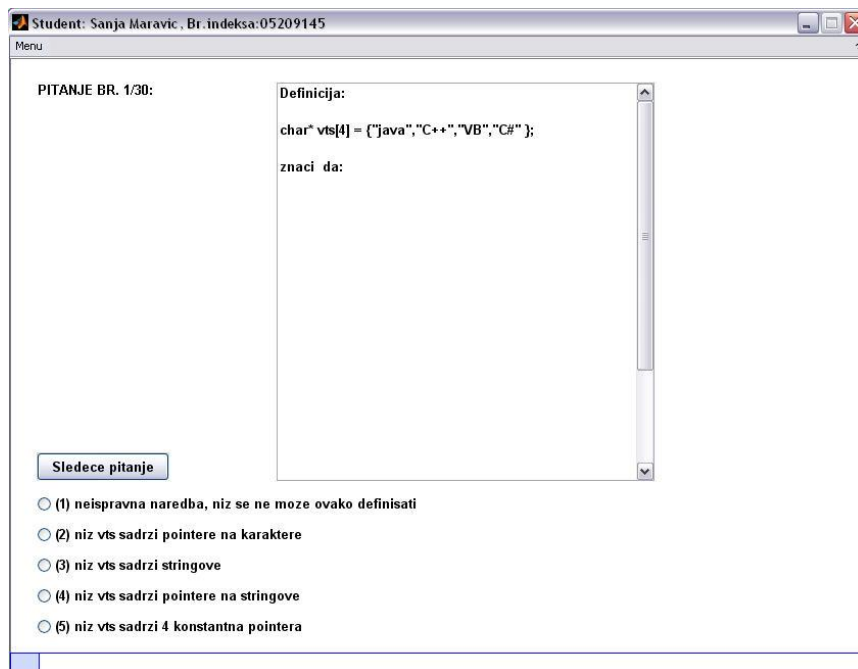
```

    pocetak_testa
    end
end

```

U ovoj funkciji ujedno se zadaje i niz `grupa_pitanja` kojim se određuje ukupan broj pitanja koji će biti postavljen na testu. U našem slučaju to je 20 pitanja. Ako ima više tipova pitanja, onda se zadaje i kojim redosledom da se smenjuju tipovi pitanja.

Nakon klika na dugme za početak testa, poziva se funkcija `pocetak_testa` u kojoj se najpre vidljivost objekata koji više ne trebaju podesi na `'off'` a vidljivost `edit` kontrole (za prikaz pitanja), `optionbutton`-a (za prikaz ponuđenih odgovora), `patch` objekta i sl. podesi na `'on'`. Nakon toga, poziva se funkcija `final_test` koja ima dva izlazna parametra: niz sa odgovarajućim brojem tačnih/netačnih odgovora i niz koji sadrži vreme po pitanju (izraženo u sekundama) koje je proteklo dok student nije dao odgovor na postavljeno pitanje. Na slici 6.31 prikazan je izgled prozora sa pitanjem.



Slika 6.31: Prikaz pitanja

```

function [ans_array,time_arr]=final_test
[a b c t1]=ask_qn(1,1,grupa_pitanja,ones(4,4),1);
    ans_array = b;
    time_arr = t1;
end

```

Kao što se vidi, sledeća funkcija koja se poziva je funkcija `ask_qn` koja ima pet ulaznih parametara i četiri izlazna parametra. Za slučaj kada postoje pitanja samo jednog tipa koristi se samo drugi i četvrti parametar.

U funkciji `ask_qn` sve se odvija u jednoj `for` petlji koja se ponavlja onoliko puta koliko ima pitanja (`comm_arr`). Pri svakom prolasku, najpre se svojstvo `Value` svih `radiobutton`-a podesi na nula, a zatim se vrše ostala izračunavanja i pozivanja nekih drugih funkcija.

Prvo izračunavanje se tiče određivanja težine pitanja koje treba da se postavi. Parametar, `question_set`, koji može biti 1, 2 ili 3 se uveća (ili smanji u nekom od sledećih prolazaka kroz petlju) za onoliko koliko je naveden sledeći parametar iste funkcije, parametar `ask_1` (u ovom slučaju je to 1). Pošto je u ovom programu `question_set` na samom početku 1 i `ask_1` takođe je 1, znači da se test započinje pitanjem težine 2 (srednje teško pitanje).

```
deciding_factor=ask_1;
question_set=normalize_qno(question_set,deciding_factor,1,3);
```

Pored seta pitanja, podešava se i vrsta pitanja iz koje će se birati sledeće pitanje. U ovom testu reč je o samo jednoj vrsti pitanja pa je suvišno svaki put odlučivati o vrsti pitanja, ali za slučaj budućih dorađivanja GUI-a i dodavanja više vrsta pitanja, ovakva logika će biti od velike pomoći.

```
ask_1_char_type=question_type(com_arr(i));
```

Niz `com_arr(i)` sačinjen je samo od jedinica, pa je, kako se i može pretpostaviti, rezultat funkcije uvek isti:

```
ask_1_char_type = 'pitanja'

function question_str=question_type(number)
if(number==1)
    question_str='pitanja';
end
```

Drugi podatak koji je potrebno izračunati pre nego što se pristupi isčitavanju pitanja jeste `question_status`. Ovaj parametar čuva podatke organizovane u matricu (kada, pored različitih težina zadataka postoje i različiti tipovi pitanja), ali je konkretno u ovom slučaju u pitanju vektor vrsta inicijalizovan sa `ones(1,3)`, gde jedinica označava samo jednu vrstu pitanja, a 3 znači da postoje tri nivoa težine pitanja.

Konačno, poziva se funkcija:

```
[ask_1
q_time]=ask(ask_1_char_txt,question_status(com_arr(i),question_set))
;
```

Dakle, prvi parametar funkcije daje informaciju iz kog .txt fajla da se vrši isčitavanje pitanja dok drugi parametar, `question_status(com_arr(i),question_set)`, daje informaciju od kog reda u navedenom fajlu treba početi sa isčitavanjem. Po dobijanju izlaznih parametara, vrši se njihovo smeštanje u promenljive `qn_time` i `ans_array` na sledeći način:

```
qn_time=[qn_time q_time];
ans_array=[ans_array ask_1];
```

Poslednji koraci u jednom prolasku *for* petlje su:

- osvežavanje rednog broja početne linije za sledeće isčitavanje u aktuelnom .txt fajlu. Ovde se to radi dodavanjem broja 35 (29 linija za postavku pitanja + 5 linija teksta sa ponuđenim odgovorima + 1 linija koja se korisniku ne prikazuje nego se čuva u promenljivoj `num_tline` i predstavlja tačan odgovor).

```
question_status(com_arr(i),question_set)=question_status(com_arr(i),
question_set)+35;
```

- uvećavanje promenljive koja daje informaciju o broju postavljenih pitanja

```
ques_num_start=ques_num_start+1;
```

Samo isčitavanje iz .txt fajlova pomoću funkcije `ask` izvršava se na sledeći način:

- otvori se odgovarajući fajl

```
fid=fopen(ask_1_char_txt);
```

- nakon toga se pomoću `strvcat` formira string pitanja u jednoj *for* petlji koja se ponavlja 35 puta

```
for(k=1:35+start_line_number-1) %
tline = fgetl(fid);
    if strcmp(tline,'')
        pitanja=strvcat(pitanja,' ');
    else
        pitanja=strvcat(pitanja,tline);
    end
        lin_count=lin_count+1;
end
```

Prikazivanje isčitanoog stringa u edit prozoru:

```
colwidth = 74;
set(postavka_pitanja,'Units','characters')
string={pitanja(start_line_number:(start_line_number+28),:)};
outstring1 = textwrap(postavka_pitanja,string,colwidth);
newpos1 = get(postavka_pitanja,'Position');
newpos1(3) = colwidth;
```

```
newpos1(4) = length(outstring1)+1;
set(postavka_pitanja, 'String', outstring1, 'Position', newPos1)
```

`Textwrap` za ograničavanje širine prozora na 74 karaktera koristiti se iz razloga što se bez njega tekst pri ispisu prelomi na neočekivanim mestima (što je u slučaju prikazivanja teksta samo estetski problem, ali u slučaju prikazivanja nekog programskog koda stvara veliki problem).

Kao što se može primetiti, prikazano je samo 29 linija stringa, što znači da je ostalo još da se prikažu ponuđeni odgovori:

```
set(option_button(1), 'String', pitanja((start_line_number+29), :))
set(option_button(2), 'String', pitanja((start_line_number+30), :))
set(option_button(3), 'String', pitanja((start_line_number+31), :))
set(option_button(4), 'String', pitanja((start_line_number+32), :))
set(option_button(5), 'String', pitanja((start_line_number+33), :))
```

Preostalo je da se otpočne sa merenjem vremena koje prolazi dok korisnik ne da bilo kakav odgovor (tj. dok ne pritisne dugme *Sledeće pitanje/Prikaži rezultate*). Vreme se meri tako što se sa `tic` startuje tajmer a potom se u `while` petlji na svakih 0.02s proverava stanje promenljive `dat_odgovor`. Ova promenljiva je inicijalizovana na 0, a za slučaj da je korisnik pritisnuo dugme *Sledeće pitanje*, vrednost promenljive se menja na 1 što će:

- zaustaviti `while` petlju,
- resetovati `dat_odgovor` za sledeću upotrebu
- zapisati izmereno proteklo vreme: `q_time=[q_time toc];i`
- potom se proverava da li se vrednost korisnikovog odgovora podudara s tačnim odgovorom (`if(taster==num_tline)`), ako odgovara, podesiće `output_check=1;`

```
if(rem(lin_count,35)==0)
    num_tline=tline-48;
tic
while(dat_odgovor==0)
    pause(0.02)
    userdt=get(ex, 'UserData');
    if (userdt==0)
        delete(figura)
        return
    end
end
dat_odgovor=0;
q_time=[q_time toc];
if(taster==num_tline)
    output_check=1;
end
end
```

Promenljiva `taster` se izračunava pritiskom dugmeta *Sledeće pitanje*:

```
taster=str2num(korisnikov_odgovor);
```

gde `korisnikov_odgovor` odgovara tagu izabranog `optionbutton`-a (ako je selektovan):

```
korisnikov_odgovor=get(source,'Tag');
```

a ako nije selektovan, `taster` će biti 0 (pošto je tako podešeno na samom početku):

```
korisnikov_odgovor='0';
```

Nakon što je korisnik prešao preko svih pitanja, program nastavlja da se odvija u funkciji `pocetak_testa` od onog dela gde je pozvana funkcija :

```
[a b]=final_test;
```

Kada je formiran niz sa tačnim odgovorima (a) i niz sa izmerenim vremenom po svakom pitanju (b), potrebno je izračunati konačan rezultat i prikazati ga korisniku. Potrebno je prvo podesiti na `'off'` vidljivost nepotrebnih elemenata GUI-ja, kako bi se tabela i dva teksta mogla prikazati na istoj figuri. Zatim sledi izračunavanje konačnih rezultata pozivanjem funkcije `totaling` koja kao parametar uzima niz `a` s odgovorima.

```
total_marks=int2str(totaling(a));
```

Prikaz tabele i konačnog rezultata:

```
columnname1 = {'Tezina pitanja','Tacno/Netacno','sec/pitanju'};
t1 = uitable('Units','normalized','Position',[0 0.001 0.58 0.9],...
            'ColumnName', columnname1,...
            'ColumnWidth',{90.5},...
            'Visible','on');
dat=[tezina_pitanja' a' b'];
set(t1,'Data',dat)
final_score=strvcat('Krajnji rezultat:',[ ' ', total_marks
'/100']);
Text3 = uicontrol(panel_dod,'Style',...
                 'text','String',final_score,...
                 'BackgroundColor','white',...
                 'HorizontalAlignment','left',...
                 'FontName','arial',...
                 'FontWeight','bold',...
                 'HorizontalAlignment','center',...
                 'Visible','on');
```

```
'FontSize', 14, ...
'Visible', 'on', ...
'Position', [330,100,200,50]);
```

Ako korisnik odabere opciju da sačuva rezultate testa, poziva se funkcija koja za parametre uzima sedam elemenata (tri koja je korisnik sam uneo kroz dijalog za unos podataka, a četiri podatka su se izračunala tokom testiranja):

```
sacuvaj_rezultate(ime,prezime,br_indeksa,a,tezina_pitanja,b,total_marks)
```

Izgled ekrana aplikacije sa prikazom rezultata dat je na slici 6.32.

	Tezina pitanja	Tacno/Netacno	sec/pitanju
1	2	0	19.9536
2	1	0	6.3628
3	1	0	9.1372
4	1	1	5.3806
5	2	0	5.9608
6	1	0	3.9407
7	1	1	7.4421
8	2	1	5.5676
9	3	0	6.0992
10	2	1	5.8967
11	3	1	19.6430
12	3	1	8.3513
13	3	1	3.3496
14	3	1	5.3362
15	3	0	21.7765
16	2	0	7.5058
17	1	0	5.6952
18	1	1	8.2771
19	2	0	8.7138
20	1	1	15.5513
21	2	0	7.1675
22	1	1	9.3100
23	2	1	4.1316
24	3	0	7.8651
25	2	1	10.2798
26	3	1	6.1906
27	3	1	6.9241
28	3	0	6.5549
29	2	1	5.3610
30	3	0	11.0152

**Krajnji rezultat:  
16/30**

Slika 6.32: Prikaz rezultata

U dnu GUI-ja dodat je i jedan *progress bar* koji slikovito prikazuje koji deo testa je korisnik prešao i koliko pitanja mu još ostaje do kraja. To je postignuto kreiranjem *patch* objekta kome se menja dužina u zavisnosti od broja pređenih pitanja:

```
ose0 = axes ('Parent',figura,...
            'Position',[0 0 1 .03],...
            'Visible','on');

axis off
xdata0 = [0;0;4;4];
ydata0 = [0;1;1;0];
zdata0 = ones(4,1);
```



```

okvir=patch(xdata0,ydata0,zdata0,'w','EdgeColor','w','FaceColor','white');

x=100/(20*100);
ose = axes ('Parent',figura,...
            'Position',[0 0 x .03],...
            'Visible','on');

axis off

xdata = [0;0;4;4];
ydata = [0;1;1;0];
zdata = ones(4,1);
progres=patch(xdata,ydata,zdata,'w','EdgeColor','w','FaceColor','w');
;
    
```

Vrednost  $x$  (tj. dužina *patch* objekta) se osvežava pri svakom od 20 prolazaka *for* petlje u funkciji `ask_qn`. Kao što se vidi, da bi se menjale dimenzije *patch* objekta, menjaju se samo dimenzije osa koje ga „nose”.

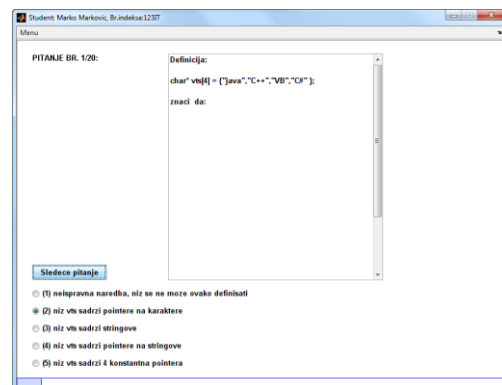
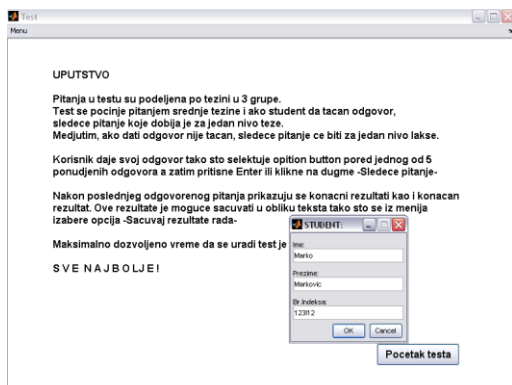
```

x=ques_num_start*(100/(length(grupa_pitanja)*100));
set(ose,'Position',[0 0 x .03]);
axis off
    
```

Nakon završetka testa, moguće je iz menija odabrati opciju da se rezultati sačuvaju u .txt fajlu, što omogućava dalju obradu rezultata.

## 6.8. Primer testa

U nastavku je dat primer jednog testa koji ukupno ima 20 pitanja (slika 6.33). Slike predstavljaju *screenshot*-ove testa, počevši od uvodnog ekrana sa osnovnim instrukcijama za ispitanika i formom za unos imena, prezimena i broja indeksa ispitanika, do prikaza rezultata koji je student postigao na testu.



Student: Marko Markovic, Brindevka123IT

Menu

PITANJE BR. 2/20:

Sta ce biti ispisano nakon izrsenja sledecih linija koda:

```
int a,b,c;
int f=(1,2,3);
int *p=&f[0];

a="*+*+*";b="*+*+*";
cout<<a<<b<<c;
```

Sledece pitanje

(1) 321

(2) 220

(3) 232

(4) 223

(5) 123

Student: Marko Markovic, Brindevka123IT

Menu

PITANJE BR. 3/20:

Ako imamo funkcije

```
void ispis(char *slova, int a=1, float b=3.14) i
void ispis(char *slova, int a=1)
```

koji poziv funkcije nece javiti grešku:

Sledece pitanje

(1) ispis("VTS");

(2) ispis("VTS", 2);

(3) ispis("VTS", 2, 3);

(4) za sve ce javiti grešku

(5) ispis();

Student: Marko Markovic, Brindevka123IT

Menu

PITANJE BR. 4/20:

Ako se kreira novi objekat korisciteci izraz:

```
x = new triplo;
```

onda za pristup f() funkciji clanici ovog objekta, treba koristiti izraz:

Sledece pitanje

(1) triplo.f();

(2) triplo->f();

(3) x.f();

(4) x->f();

(5) X.f();

Student: Marko Markovic, Brindevka123IT

Menu

PITANJE BR. 5/20:

Ako se izvise izrazi:

```
int j, k;
j = 123;
k = 234;
int *q, *r;
cout << *q << " " << *r;
```

Sta ce se ispisati:

Sledece pitanje

(1) Vrednosti q i r

(2) Adrese j i k

(3) 123, 234

(4) 234, 123

(5) Nista od gore navedenog

Student: Marko Markovic, Brindevka123IT

Menu

PITANJE BR. 6/20:

Simbol '&' u programskom jeziku C++ oznacava pristup promenljivoj:

Sledece pitanje

(1) po vrednosti

(2) po referenci

(3) preko konstruktora

(4) koja je definisana kao lokalna

(5) koja je definisana kao globalna

Student: Marko Markovic, Brindevka123IT

Menu

PITANJE BR. 7/20:

Konstruktor ima ime:

Sledece pitanje

(1) kao i klasa kojoj pripada

(2) void inicijalizovoid

(3) constructor ime\_klase(void)

(4) const ime\_klase(void)

(5) int ime\_klase(void)

Student: Marko Markovic, Brindevka123IT

Menu

PITANJE BR. 8/20:

Pa UML standardno kako se obelezavaju private, public i protected clanovi klase:

Sledece pitanje

(1) private: +, public: -, protected: ~

(2) private: +, public: -, protected: #

(3) private: -, public: +, protected: ~

(4) private: -, public: +, protected: #

(5) private: -, public: +, protected: ~#

Student: Marko Markovic, Brindevka123IT

Menu

PITANJE BR. 9/20:

Koju vrednost mora da vrati destruktor?

Sledece pitanje

(1) pointer na klasu

(2) jedan objekat klase

(3) kod koji odredjuje da li se klasa ispravno izvrsila

(4) destruktor ne vraca vrednost

(5) tip povratne vrednosti odredjuje tok poziva destruktor

Student: Marko Marković, Br. indeksa:123BT

PITANJE BR. 10/20: Koja od sledećih naredbi omogućava pristup privremenoj var koja pripada strukturi b?

Sledeće pitanje

(1) b[var];  
 (2) b.var;  
 (3) var->b;  
 (4) b\*var;  
 (5) var.b;

Student: Marko Marković, Br. indeksa:123BT

PITANJE BR. 11/20: Koja od sledećih naredbi daje memorijku adresu celobrojne promenljive a?

Sledeće pitanje

(1) \*a;  
 (2) \*a;  
 (3) &a;  
 (4) add(a);  
 (5) address(a);

Student: Marko Marković, Br. indeksa:123BT

PITANJE BR. 12/20: Koja od sledećih naredbi predstavlja ispravno definisanu strukturu?

Sledeće pitanje

(1) struct {int a};  
 (2) struct a\_struct {int a};  
 (3) struct a\_struct int a;  
 (4) struct a\_struct {int a};  
 (5) struct a;

Student: Marko Marković, Br. indeksa:123BT

PITANJE BR. 13/20: Apstrakcija predstavlja:

Sledeće pitanje

(1) zanemarivanje nebitnih delova objekta i koncentrisanje na one koji ga razlikuju od ostalih objekata  
 (2) niz operacija koje klijenti mogu obaviti nad objektom  
 (3) skrivanje konkretne implementacije od okoline  
 (4) razlaganje sistema u skup relativno nezavisnih modula koji objedinjuje klasu  
 (5) jedan od metoda na koji objekat može da reaguje na zahtev koji mu je upućen

Student: Marko Marković, Br. indeksa:123BT

PITANJE BR. 14/20: Podaci koji su u strukturi, odnosno klasi, deklarisani kao public dostupni su:

Sledeće pitanje

(1) iz svih tacaka programa  
 (2) samo funkcijama članicama  
 (3) samo privatnim funkcijama članicama  
 (4) samo preko konstruktora  
 (5) samo preko destruktora

Student: Marko Marković, Br. indeksa:123BT

PITANJE BR. 15/20: Ako se kreira novi objekat koristeći izraz `a = new X;` onda za pristup g() funkciji članici u ovom objektu, treba koristiti izraz:

Sledeće pitanje

(1) X.g();  
 (2) X->g();  
 (3) a.g();  
 (4) a->g();  
 (5) X:g();

Student: Marko Marković, Br. indeksa:123BT

PITANJE BR. 16/20: Ako imamo klasu C koja je izvedena javno iz klase B, koja je opet izvedena iz klase A isto na javni način, i ako sve klase imaju konstruktore, kako će se one izvršiti ako se kreira jedan objekat tipa C:

Sledeće pitanje

(1) izvršava se samo konstruktor u C-u.  
 (2) izvršava se samo konstruktor u roditelju, to jest iz A.  
 (3) prvo konstruktor iz A, pa iz B i na kraju iz C.  
 (4) prvo konstruktor iz C, pa iz B i na kraju iz A.  
 (5) izvršava se samo konstruktor u B

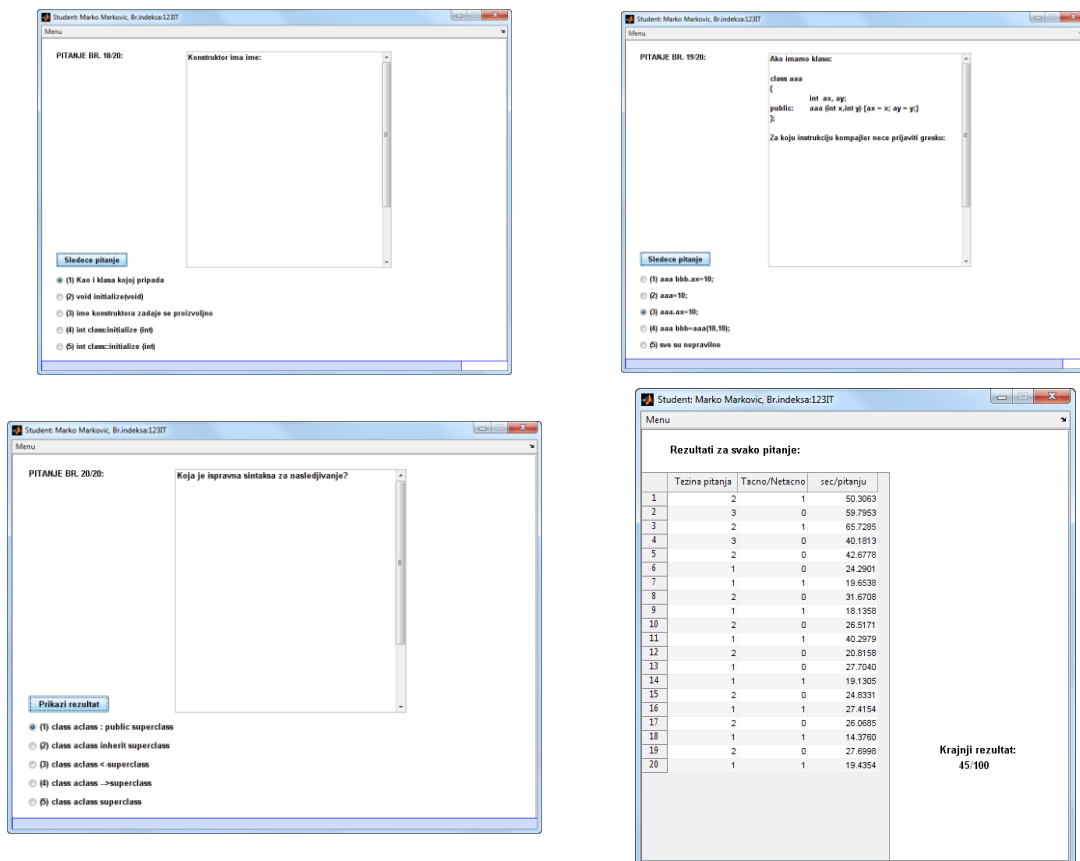
Student: Marko Marković, Br. indeksa:123BT

PITANJE BR. 17/20: Odgovarajući destruktor za datu klasu aaa bio bi:

```
class aaa
{
public: aaa(const int size);
private:
int _abc;
float _bcd;
char c;
};
```

Sledeće pitanje

(1) ~aaa(){}(delete abc);  
 (2) ~aaa(){}(delete bcd);  
 (3) ~aaa(){}(delete c);  
 (4) ~aaa(){}(delete const int size);  
 (5) ~aaa(){}(delete abc);



Slika 6.33: Izgled testa

## 6.9. Poređenje aplikacije sa sličnim rešenjima

Računarski adaptivni test realizovan za potrebe istraživanja, upoređen je sa dva slična adaptivna testa koji su se koristili (ili se koriste) na visokoškolskim ustanovama za proveru znanja. Prvo sledi opis ova dva sistema za adaptivno testiranje, a zatim njihovo poređenje sa adaptivnim testom realizovanim u MATLABu. Poređenje je urađeno za četiri glavne komponente koje svaki adaptivni test mora da ima, a to su: baza pitanja, procedura za selekciju pitanja, metode za procenjivanje sposobnosti i pravilo zaustavljanja [119].

PAT (engl. *Programming Adaptive Testing*) je sistem za adaptivno testiranje koji je razvijen na Univerzitetu Makedonija u Solunu, na Departmanu za Informacione sisteme (engl. *University of Macedonia, Information Systems Department, Thessaloniki*). Sistem je rađen u Flash MX alatu a za programiranje je korišćen *Action Script*. Adaptivni test može biti instaliran na računaru, ili mu se može pristupiti putem weba. Sistem je namenjen za proveravanje znanja programiranja i klasifikovanje ispitanika prema programerskim veštinama u tri kategorije: niska, srednja i visoka. Cilj testiranje je bio da se na osnovu uspeha postignutog na ovom testu predvidi uspeh ispitanika na nacionalnom testu Panhelenik (engl. *Panhellenic National Exam*) koji se polaže nakon treće godine srednje škole i koji je uslov za upis na visokoobrazovne ustanove u Grčkoj [28]. Test je namenjen početnicima u programiranju.

Pitanja su podeljena u tri kategorije po težini, A=laka pitanja, B=pitanja srednje težine i C=teška pitanja. Pitanja su dizajnirana u skladu sa Blumovom taksonomijom i prva tri kognitivna nivoa: prisećanje ili prepoznavanje informacija, razumevanje i primena. Za prvi kognitivni nivo, koji se odnosi na prisećanje, razvijena su pitanja koja su se odnosila na sintaksu pseudo jezika koji su ispitanici učili (*Glossa*), osnove strukturnog programiranja i podprograme.

Za drugi kognitivni nivo (razumevanje) pitanja su se odnosila na razumevanje dela koda, npr. određivanje izlaza nakon izvršenja nekog programskog koda. Ova pitanja bila su za nivo B i C.

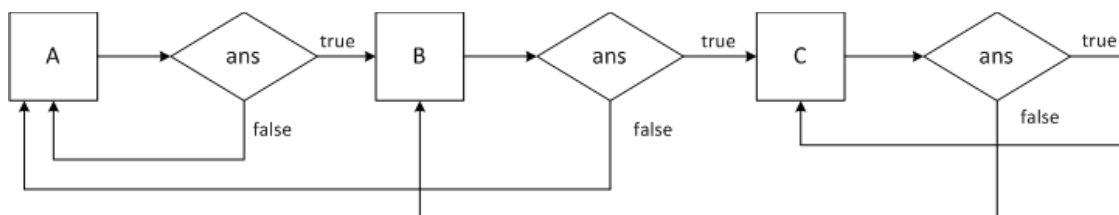
Pitanja koja su se odnosila na primenu naučenog za rešavanje novih problema, činila su pitanja koja su zahtevala od ispitanika da za određeni deo koda odaberu ekvivalentne naredbe, logički dijagram za koji je bilo potrebno da se odaberu naredbe koje ga predstavljaju, kao i popunjavanje praznina u programskom kodu. Ukupan broj pitanja u bazi bio je 443, a broj pitanja u svakoj od tri kategorije prikazan je u tabeli 6.2:

Tabela 6.2: Broj pitanja prema kategorijama

Kognitivni nivo	A	B	C	Ukupan broj pitanja
Prisećanje	186	93	30	309
Razumevanje	0	35	18	53
Primena	0	20	61	81
Ukupan broj pitanja po nivoima težine	186	148	109	443

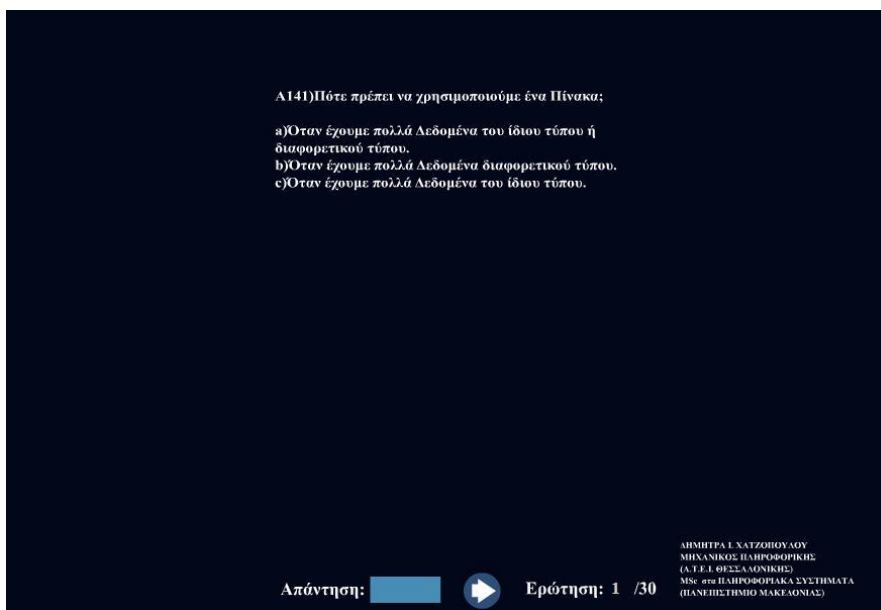
Test su činila dva tipa pitanja: pitanja višestrukog izbora i tačno/netačno. Pitanja tačno/netačno nisu bila predviđena za proveravanje znanja iz trećeg Blumovog domena koji se odnosi na primenu naučenog na rešavanje novih problema. Pitanja višestrukog izbora razlikovala su se u broju ponuđenih odgovora, u zavisnosti na koji Blumov nivo su se odnosila. Tako pitanja kategorije A su bila sa 3 ponuđena odgovora, kategorije B sa četiri a za kategoriju C bilo je 5 mogućih odgovora.

Struktura sistema prikazana je na slici 6.34. Algoritam izbora pitanja je takav, da ukoliko je odgovor tačan prelazi se na pitanje iz teže kategorije, a ukoliko je netačan na pitanje iz lakše kategorije. U svakoj kategoriji pitanja se biraju slučajno, a ukupan broj pitanja na testu je 30. Test započinje slučajnim izborom jednog pitanja iz kategorije A.



Slika 6.34: Struktura modela sistema [28]

Na slici 6.35 je prikazan primer jednog pitanja. Svako pitanje ima oznaku kojoj kategoriji pripada i redni broj pitanja unutar kategorije kojoj pripada. Na slici je prikazano pitanje koje ima oznaku A141, što označava da je iz kategorije lakih pitanja (A) i njegov redni broj unutar kategorije u kojoj se nalazi je 141. Za pitanja iz grupe A ponuđena su tri odgovora. Ispitanik daje odgovor ukucavanjem slova (a, b ili c ) koje stoji pored odgovora za koji smatra da je tačan. Ne postoji mogućnost vraćanja na prethodno pitanje, što je karakteristično za adaptivne testove.



Slika 6.35: Primer pitanja [28]

Broj poena koji se može dobiti ako je odgovor tačan zavisi od kategorije kojoj pitanje pripada. Tako, tačan odgovor na pitanje iz kategorije A vredi jedan poen, iz kategorije B dva poena, a iz kategorije C tri poena. Nakon završetka testa, ispitanik dobija izveštaj o svom uspehu na testu, kao što je prikazano na slici 6.36.

Name : John		Surname : Papadopoulos		Class : G2		
Your Total Result is : 6 /30			Final Score: 6 /87=6 %			
TRY HARDER! LOW PROGRAMMING SKILLS SPECIFICALLY			Correct Answers in Level A: 6 /24 Correct Answers in Level B: 0 /6 Correct Answers in Level C: 0 /0			
	QUESTIONS	WRONG ANSWERS	QUESTIONS / ANSWERS			
CHAPTER 1	2	2	A62	a	A139	a
CHAPTER 2	7	6	A177	b	B2	d
CHAPTER 3	3	2	A60	c	A103	a
CHAPTER 4	1	1	A185	a	A56	b
CHAPTER 6	3	2	A105	a	B135	c
CHAPTER 7	1	1	A133	b	A141	a
CHAPTER 8	3	1	A89	a	A75	a
CHAPTER 9	7	5	B16	c	A3	a
CHAPTER 10	3	3	A119	a	B89	b
			B145	b	A60	c
			A71	a	A72	a
			A28	c	A32	b
			B70	d	A13	c
			A24	a	A72	a
			A51	c	A114	a

Slika 6.36: Izveštaj o uspehu na testu [28]

Ispitaniku se prikazuju sledeći podaci:

- Ukupan rezultat (x): broj tačnih odgovora od ukupnog broja od 30 pitanja. Maksimalna vrednost je 30.
- Broj tačnih odgovora po nivoima, dat u odnosu na ukupan broj pitanja postavljenih na datom nivou. Na primer 6/24 znači da je na određenom nivou ispitanik imao 6 tačnih odgovora od 24 pitanja koja su bila postavljena na tom nivou.
- Finalni rezultat (y) određuje se na sledeći način: broj tačnih odgovora na nivou A + broj tačnih odgovora na nivou B + broj tačnih odgovora na nivou C. Maksimalna vrednost je 87.
- Klasifikacija ispitanika prema programerskim veštinama u jednu od tri kategorije, a zavisi i od ukupnog rezultata i od finalnog. Klasifikacija se radi na sledeći način:
  - Ako je ( $0 \leq x \leq 17$ ) i ( $0 \leq y \leq 33$ ) nizak nivo programerske veštine
  - Ako je ( $16 \leq x \leq 20$ ) i ( $34 \leq y \leq 51$ ) srednji nivo programerske veštine
  - Ako je ( $21 \leq x \leq 30$ ) i ( $52 \leq y \leq 87$ ) visok nivo programerske veštine
- Analitički deo rezultata prikazuje ukupan broj pitanja koji je postavljen iz svakog poglavlja (ima ih ukupno 10) i broj netačno datih odgovora za svako poglavlje. Na ovaj način ispitanik može da vidi na koja poglavlja je potrebno da obrati pažnju u zavisnosti od broja netačnih odgovora koje je imao.

Sledeća aplikacija koja je odabrana za predstavljanje je aplikacija koja je razvijena na Univerzitetu Hertfordšajar u Velikoj Britaniji (engl. *the University of Hertfordshire*). Matematički model koji je korišćen za razvoj adaptivnog testa bazira se na troparametarskom modelu (3PL) teorije odgovora na stavke (IRT). Jednačina 3PL

funkcije pokazuje verovatnoću da korisnik nepoznatog nivo znanja tačno odgovori na pitanje težine  $b$ , parametra diskriminacije  $a$  i parametra pogađanja  $c$  ( $0 \leq c \leq 1$ ) [79].

$$P(\theta) = c + \frac{1 - c}{1 + e^{-1.7a(\theta - b)}}$$

Za svaki odgovor koji ispitanik da, matematička funkcija koju predstavlja jednačina koristi se za procenjivanje nivoa znanja ispitanika. Kako 3PL model zahteva poznavanje parametara  $a$ ,  $b$  i  $c$ , podaci o vrednostima ovih parametara čuvaju se u bazi pitanja zajedno sa samim pitanjem [13]. Za pitanja bez istorijskih podataka, inicijalna vrednost parametra težine pitanja  $b$ , definisana je na osnovu procene stručnjaka i kretala se od -3 (najmanja vrednost) do +3 (najveća vrednost). Kalibracija pitanja bazirala se na Blumovoj taksonomiji kognitivnih sposobnosti i data je u tabeli 6.3 [75]:

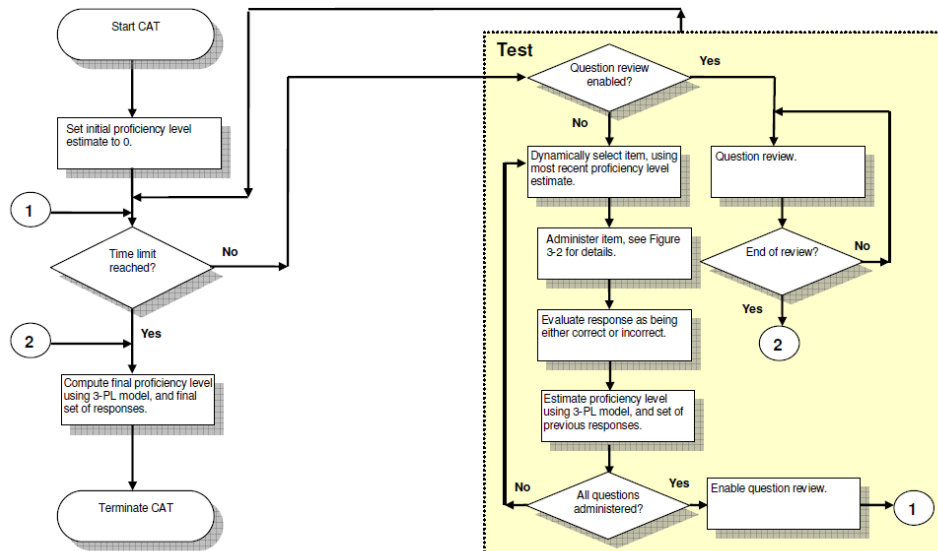
Tabela 6.3: Kalibracija pitanja

Težina	Kognitivna veština
$-3 \leq b \leq -1$	znanje
$-1 \leq b \leq +1$	razumevanje
$+1 \leq b \leq +3$	primena

Test je prvo podeljen prema oblastima, a zatim su za svaku oblast definisana pitanja koja su pokrivala nastavni materijal za sva tri nivoa težine pitanja. Ukupan broj pitanja u bazi je 150. Test započinje slučajnim izborom jednog pitanja iz grupe pitanja srednje težine ( $b=0$ ). Na osnovu odgovora studenta, procenjuje se nivo znanja prema jednačini, a zatim se iz baze pitanja bira ono čiji je parametar težine  $b$  najpribližnije odgovara vrednosti procenjenog nivoa znanja. Kao pravilo zaustavljanja koristi se kombinovana metoda vremenskog ograničenja trajanja testa ili datog odgovora na sva pitanja u testu, u zavisnosti koji uslov se prvi ispuni. Algoritam po kome radi adaptivni test, prikazan je na slici 6.37.

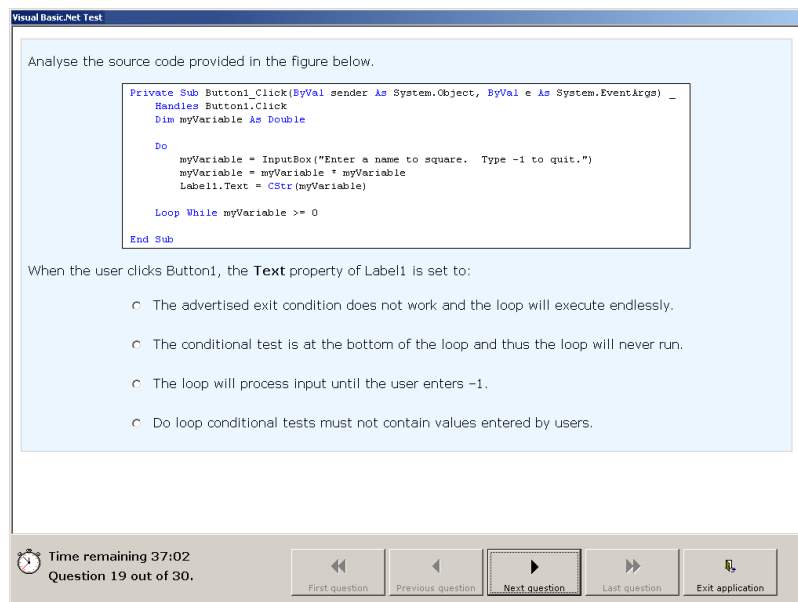
Aplikacija je rađena za Microsoft Windows platformu u Visual Basicu verzija 6. Za bazu pitanja koristi se Microsoft Access.





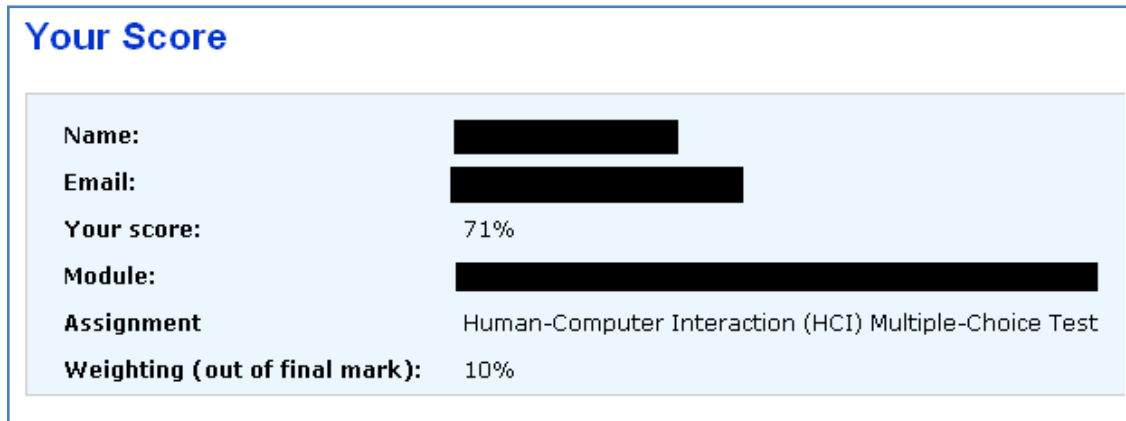
Slika 6.37: Algoritam adaptivnog testa [78]

Pitanja su bila tipa pitanja sa višestrukim izborom sa četiri ponuđena odgovora. Izgled testa prikazan je na slici 6.38. U donjem levom uglu ispitanik ima informaciju koliko je vremena preostalo do kraja testa, kao i na koliko je pitanja odgovorio od ukupnog broja predviđenih pitanja (u ovom slučaju 30). Navigacija omogućava da se korisnik kreće samo na sledeće pitanje i da napusti aplikaciju. Za potrebe istraživanja kako na rezultat ispitanika utiče mogućnost da se vrati na prethodno odgovorena pitanja [76], postavljene su i kontrole koje omogućavaju vraćanje na prethodna pitanja.



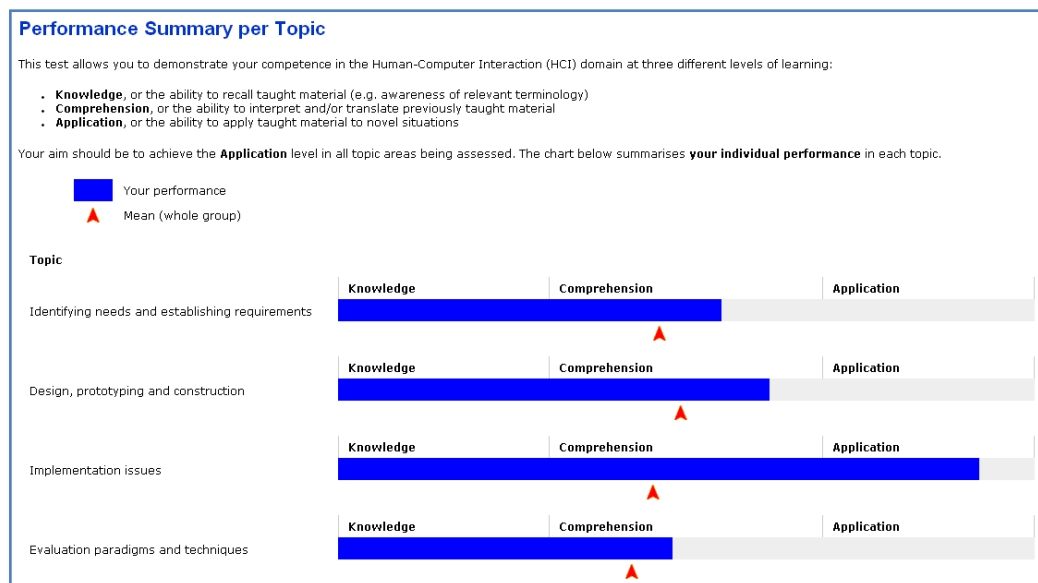
Slika 6.38: Grafički korisnički interfejs računarskog adaptivnog testa [11]

Nakon završetka testa, ispitanik dobija izveštaj o uspehu na testu, kao što je prikazano na slici 6.39.



Slika 6.39: Prikaz rezultata testa [11]

Odgovori ispitanika grupišu se prema oblastima kojoj pitanje pripada i određuje se nivo znanja za svaku od oblasti, kao što je prikazano na slici 6.40. Procenjeni nivo znanja zatim se mapira prema Blumovoj taksonomiji kognitivnih veština. Ispitanik ima informaciju o svom uspehu na testu (crvena strelica) u odnosu na prosečan uspeh cele grupe (plavi pravougaonik).



Slika 6.40: Prikaz uspeha na testu prema Blumovoj taksonomiji [11]

Kako bi se povratna informacija koju ispitanik dobija o svom uspehu na testu učinila približno jednaka onoj koju dobija u direktnom razgovoru sa nastavnikom, aplikacija pruža informaciju ispitaniku na koje oblasti da obrati pažnju. Naime, pretpostavka je da će u direktnom razgovoru sa ispitanikom nastavnik objasniti pitanje na koje ispitanik nije znao odgovor, a ne samo dati informaciju koji odgovor je tačan. Stoga se za svako pitanje u bazi podataka pored tačnog odgovora, nalazi i rečenica koja objašnjava odgovor i u kom poglavlju u udžbeniku je obrađeno pitanje. Na slici 6.41 je ilustrovan primer povratne sprege koju dobija ispitanik za netačne odgovore koje je dao na testu.

**Recommended Points for Revision**

Your personalised revision plan comprises four sections:

- [Identifying needs and establishing requirements](#)
- [Design, prototyping and construction](#)
- [Implementation issues](#)
- [Evaluation paradigms and techniques](#)

**Identifying needs and establishing requirements**

Did you know this?	Further action that you should do...
Use cases are usually employed for capturing the functional requirements of a system. A typical use case should convey a scenario's typical course of events.	Read Sections 7.6.2 and 7.6.3 from "Interaction Design: Beyond Human-Computer Interaction" and identify the difference between a <b>use case</b> and an <b>essential use case</b> .
<b>Utility</b> is a usability goal that refers to the extent to which the system provides the right kind of functionality.	Read Section 1.5.1 from "Interaction Design: Beyond Human-Computer Interaction".
'The system must support a user who is likely to be a well-trained engineer or scientist who is competent to handle technology' is an example of a <b>user requirement</b> .	Identify a user requirement for your Semester B [redacted] project.
A user-centered approach is characterised by "Early focus on users and tasks", "Empirical measurement" and "Iterative design".	Read Chapter 9 from "Interaction design: beyond human-computer interaction", focusing on Section 9.3 ("What is a user-centered approach?").

Back to [top](#).

Slika 6.41: Primer za „Preporučene tačke za reviziju“ [11]

U tabeli 6.4 je dat pregled aplikacije PAT, adaptivnog testa sa Univerziteta Hertfordšajar i MATLAB testa za četiri komponente koje svaki adaptivni test mora da ima: baza pitanja, procedura selekcije pitanja, metod procenjivanja i pravilo zaustavljanja.

Tabela 6.4: Poređenje aplikacija

	Baza pitanja	Procedura selekcije pitanja	Metod početne procene sposobnosti	Pravilo zaustavljanja
PAT	443 pitanja u bazi; pitanja višestrukog odgovora	Ako je odgovor tačan, sledi pitanje iz teže grupe. U suprotnom, pitanje iz lakše grupe.	Nasumično odabrano pitanje iz grupe lakih pitanja	Odgovoreno na sva pitanja (30)
Univerzitet Hertfordšajar	150 pitanja u bazi; pitanja višestrukog odgovora	3PL model i IRT	Nasumično odabrano pitanje iz grupe srednje teških pitanja	Kombinovana metoda
MATLAB test	210 pitanja (150); pitanja višestrukog odgovora	Nakon dva uzastopno netačna odgovora sledi pitanje iz lakše grupe. Ako je odgovor tačan, pitanje iz teže grupe.	Prvo pitanje iz grupe srednje teških pitanja	Odgovoreno na sva pitanja (20)

Ako ove dve aplikacije za adaptivno testiranje uporedimo sa adaptivnim testom realizovanim u MATLABu, uočavamo da im je zajedničko da su namenjeni početnicima u programiranju. Shodno tome, kod sve tri aplikacije pitanja su odabrana tako da mogu da daju efikasnu procenu prve tri kognitivne sposobnosti Blumove taksonomije: znanje, razumevanje i primena. Pitanja su kategorisana u tri grupe; laka, srednja i teška pitanja. Tip pitanja koji se koristi za procenjivanje znanja su pitanja višestrukog izbora. Kod aplikacije PAT, broj ponuđenih odgovora zavisi od kategorije kojoj pitanje pripada (3, 4 ili 5), dok su kod aplikacije sa Univerziteta Hertfordšajar ponuđena četiri odgovora, a kod MATLAB aplikacije pet.

Kod aplikacije sa Univerziteta Hertfordšajar i MATLAB testa, test započinje pitanjem srednje težine, dok se kod aplikacije PAT kao prvo pitanje bira jedno pitanje iz grupe lakih pitanja. Broj pitanja u bazi u sva tri slučaja zadovoljava uslov da je ukupan broj pitanja u bazi veći 5 do 10 puta od ukupnog broja pitanja na testu.

Što se tiče pravila zaustavljanja testa, PAT i MATLAB test koriste pravilo da je odgovoreno na sva postavljena pitanja, a aplikacija sa Univerziteta Hertfordšajar primenjuje kombinovanu metodu (dat odgovor na sva pitanja, ili vremensko ograničenje trajanja testa).

Najveća razlika između aplikacija postoji u algoritmu koji se primenjuje za selekciju sledećeg pitanja. Kod testa PAT algoritam je da nakon tačnog odgovora sledi pitanje iz grupe pitanja za jedan nivo težih, a kod aplikacije sa Univerziteta Hertfordšajar koristi se 3PL metod i funkcija odgovora na stavke za izbor sledećeg pitanja. Kod MATLAB testa, primenjen je novi algoritam kod koga tek nakon dva uzastopno data netačna odgovora nivo težine pitanje se smanjuje, a ako je odgovor tačan sledi pitanje iz grupe za jedan nivo težih.

## 7. Istraživanje

Osnovni zadatak istraživanja koje je rađeno u okviru disertacije jeste da se na osnovu teoretskih istraživanja i mogućnosti primene računarski podržanog adaptivnog testa ukaže na statistički značajnu mogućnost podizanja sveobuhvatnog nivoa i kvaliteta nastavnog procesa.

Glavna hipoteza istraživanja je da su rezultati koji se dobijaju primenom računarskog adaptivnog testa validna i pouzdana alternativa klasičnom načinu provere znanja kada se test radi na papiru. Stoga je cilj da se nađe odgovor na pitanje da li postoje razlike u rezultatima ispitanika a koje su posledica različitog modaliteta isporuke testa. Formulirana je sledeća glavna (nulta) hipoteza istraživanja:

*„Ne postoji statistički značajna razlika u postignutim rezultatima studenata koji polažu test primenom računarski adaptivnog testa i koji polažu test na klasičan način (papir i olovka).“*

Pored glavne hipoteze, definisane su i pomoćne hipoteze:

H1: računarski adaptivni test skraćuje vreme potrebno za proveru usvojenosti gradiva kod studenata;

H2: računarski adaptivni test povećava motivisanost studenata za učenje.

### 7.1. Rezultati eksperimenta

Eksperimentalni deo istraživanja rađen je sa studentima Visoke tehničke škole strukovnih studija u Subotici (VTŠ), akademske 2009/10. i 2010/11. godine.

#### 7.1.1. Učesnici eksperimenta

Ispitanici su bili studenti I godine smera Informatika koji slušaju predmet Objektivno orijentisano programiranje u drugom semestru i studenti II godine istog smera koji slušaju predmet Java u četvrtom semestru. Jedan deo ispitanika činili su i studenti II godine Elektro smera koji ovaj predmet imaju kao izborni, ali kako su predavanja, auditorne i laboratorijske vežbe jednake za sve studente bez obzira na koji smer su upisani, pripadnost određenom smeru nije uzeta u razmatranje pri obradi rezultata.

Ukupan broj studenata koji je učestvovao u istraživanju bio je 352. Učesnici istraživanja predstavljaju ujednačenu grupu po godinama (prosek godina je oko 20). Kako studenti tokom prvog semestra imaju predmete Osnovni računarstva i Računarska laboratorija I, koji imaju za cilj osposobljavanje studenata za ovladavanje osnovnim veštinama rada na računaru, nije vršen preliminarni test kako bi se proverila njihova računarska pismenost.

Većinom studenti Visoke tehničke škole su muškog pola, što se odrazilo i na uzorak ispitanika od kojih su 90.34% bili mladići a 9.66% devojke (podaci studentske službe VTŠ). Ukupan broj studenata upisanih na VTŠ je 728, od toga je 663 studenata (91.07%) i 65 studentkinja (8.93%), tako da se ispitivani uzorak može smatrati reprezentativnim. Nastava na VTŠ se odvija na srpskom i na mađarskom nastavnom jeziku, a u istraživanju su učestvovali studenti obe grupe. Kako su na oba nastavna jezika predavanja, laboratorijske i auditorne vežbe usklađene, tokom obrade rezultata istraživanja nije se posebno razmatralo na kom jeziku je student slušao nastavu.

Akadske 2009/10. godine ukupan broj studenata koji su slušali predmet Objektivno orijentisano programiranje i koji su učestvovali u istraživanju bio je 108. Na drugoj godini studija bilo je ukupno 83 studenta koji su slušali predmet Java. Ukupan broj studenata koji bi trebalo da slušaju predmet Objektivno orijentisano programiranje i Java je veći, ali jedan deo studenata tokom semestra nije dolazio na predavanja, kao ni izlazio na kolokvijume tako da nisu ni učestvovali u istraživanju.

Akadske 2010/11. godine ukupan broj studenata koji su slušali predmet Objektivno orijentisano programiranje bio je 125, a od tog broja u istraživanju je učestvovao 91 student jer pojedini studenti tokom semestra nisu dolazili na predavanja, kao ni na kolokvijume. Na drugoj godini studija bilo je ukupno 78 studenata koji su slušali predmet Java, a od toga broja u istraživanju je učestvovalo 70 studenata.

Pregled broja studenata koji su učestvovali u istraživanju dat je u tabeli 7.1.

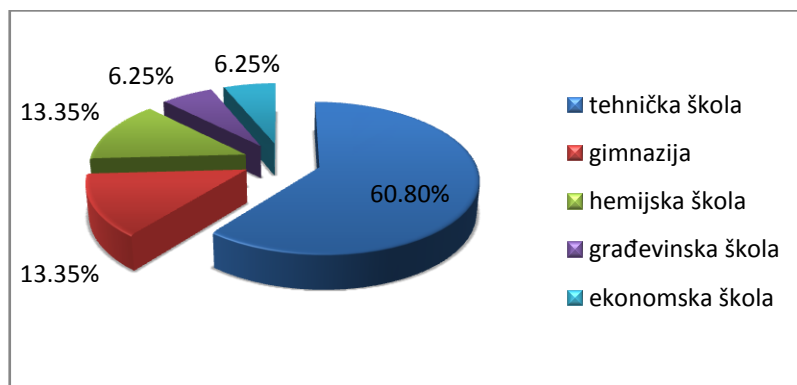
Tabela 7.1: Broj studenata koji je učestvovao u istraživanju

Kurs	Akademska 2009/2010 godina			Akademska 2010/2011 godina		
	Kontrolna grupa	Eksperimentalna grupa	Ukupno	Kontrolna grupa	Eksperimentalna grupa	Ukupno
OOP	60	48	108	46	45	91
Java	43	40	83	35	35	70

Istraživanje je organizovano po tipu eksperimenta sa paralelnim grupama. Kod eksperimenta sa paralelnim grupama postoje dve grupe ispitanika, kontrolna grupa koja je radila konvencionalan test na papiru i eksperimentalna grupa u kojoj je znanje studenata provereno primenom računarski podržanog adaptivnog testa.

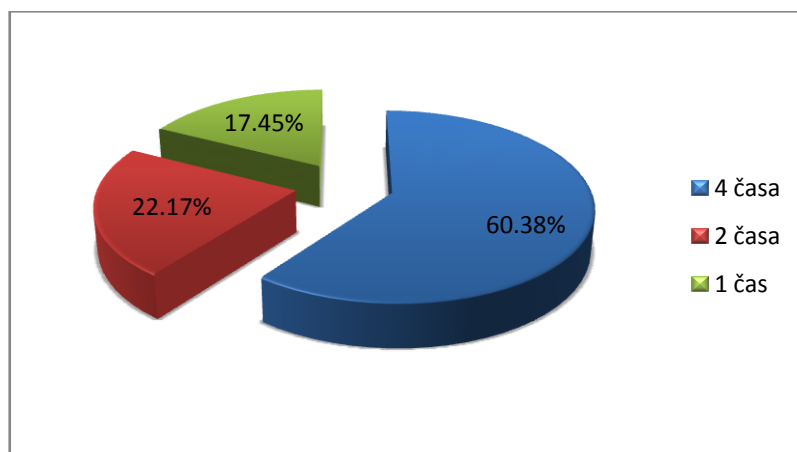
Nakon završetka testiranja svi učesnici eksperimenta popunjavali su anketu kako bi se dobili podaci o profilu studenata koji su učestvovali u eksperimentu. Visoku tehničku školu strukovnih studija mogu da upišu učenici koji su završili trogodišnju ili četvorogodišnju srednju školu, te je među ispitanicima bilo 21.02% studenata koji su završili trogodišnju školu i 78.98% onih koji su završili četvorogodišnju srednju školu. Najveći broj ispitanika završio je srednju tehničku školu elektro ili informatičkog usmerenje, ukupno 60.80%, zatim sledi gimnazija koju je završilo 13.35%, a isti

procenat ispitanika završio je i hemijsku školu. Iz srednje građevinske i ekonomske škole dolazi jednak broj studenata, 6.25%. Pregled završenih srednjih škola učesnika istraživanja može se videti na slici 7.1.



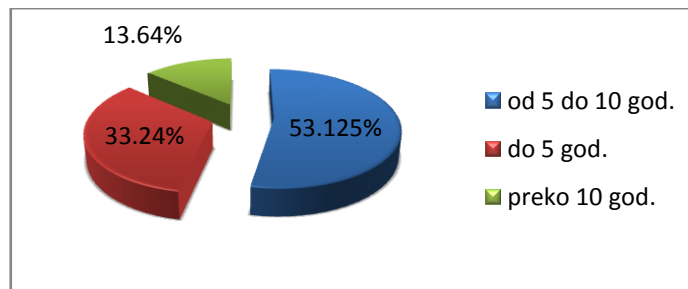
Slika 7.1: Pregled završenih srednjih škola učesnika istraživanja

Od ukupnog broja ispitanika 60.23% je imalo nastavu informatike u srednjoj školi, i to 60.38% četiri časa nedeljeno, 22.17% dva časa nedeljno i 17.45% jedan čas nedeljno, kao što se može videti na slici 7.2.



Slika 7.2: Nedeljni broj časova informatike u srednjoj školi

Od programskih jezika, studenti koji su imali nastavu informatike u srednjoj školi, naveli su da su učili Pascal, C#, C++ i Delphi. Na pitanje koliko dugo imaju računar, 53.125% studenata odgovorilo je da poseduje računar od 5 do 10 godina, 33.24% da ima računar 5 godina, a 13.64% da ima računar preko 10 godina, što je prikazano na slici 7.3.



Slika 7.3: Odgovor studenata na pitanje koliko dugo poseduju računar

U cilju boljeg razumevanja navika i iskustva studenata u korišćenju računara, postavljena su im neka pitanja u anketi koja su data u tabeli 7.2.

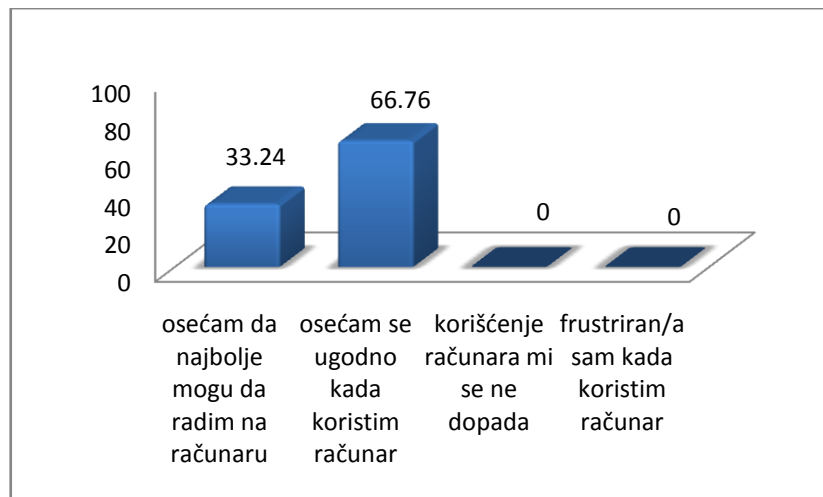
Tabela 7.2: Navike u korišćenju računara

Pitanje	Odgovor studenata
Posedujem sopstveni računar	340 (96.6%)
Koristim računar oko 5 sati dnevno	140 (39.77%)
Koji servis na internet najviše koristite - Facebook	235 (66.76%)

Na pitanje da li poseduje sopstveni računar 96.6% studenata (340 od 352) odgovorilo je potvrdno. Na pitanje koliko sati dnevno provode za računarom, skoro 40% studenata odgovorilo je da je to do 5 sati, a jedna trećina studenata je odgovorila da najviše vremena „krstari“ internetom (4 sata od 5, koliko u proseku dnevno koriste računar). Od servisa na internetu, najpopularniji je Fejsbuk koji koristi oko dve trećine studenata, zatim youtube, sajtovi za skidanje muzike, *online* igre, a oko 10% studenata je odgovorilo da internet koriste i da bi našli dodatni materijal za učenje.

Svi studenti imaju pozitivan stav prema korišćenju računara, dve trećine studenata odgovorilo je da se oseća ugodno kada koristi računar, a jedna trećina da oseća da najbolje može da radi na računaru. Ni jedan ispitanik nije izabrao ponuđen odgovor da mu se korišćenje računara ne dopada, ili da se oseća frustrirano kada koristi računar. Ovi rezultati prikazani su na slici 7.4.





Slika 7.4: Stav studenata prema korišćenju računara

Na prvoj godini studija, u prvom semestru, svi studenti imaju obavezan predmet Osnovi računarstva gde usvajaju teorijska znanja o osnovnim pojmovima informatike, informacionih sistema, zatim o hardverskim i softverskim komponentama, matematičko-logičkim osnovama računarstva, algoritmima i opisu algoritama u programskom paketu MATLAB. U prvom semestru, obavezan predmet je i Računarska laboratorija I kao stručno-aplikativni predmet, koji ima za cilj obrazovanje i osposobljavanje studenata za savladavanje teoretskog i praktičnog znanja iz oblasti obrade teksta i rada sa tabelama. Kako su ovi predmeti obavezni za sve studente nije vršena prethodna provera računarske pismenosti ispitanika jer je autor smatrao da imaju neophodna znanja o korišćenju računara kako bi mogli učestvovati u eksperimentu.

### 7.1.2. Pitanje sigurnosti procesa testiranja

Primena računarskog adaptivnog testa može da poveća sigurnost testiranja zbog činjenice da se pitanja dinamički selektuju iz baze pitanja prema individualnim sposobnostima ispitanika. Stoga je ispitanicima otežano da razmenjuju informacije o pitanjima i da prepisuju odgovore od kolega, te da na taj način poboljšaju svoj uspeh na testu. Tokom realizacije istraživanja, vodilo se računa da se obezbedi sigurnost testiranja. Zbog potencijalnih problema koji mogu da nastanu u radu mreže kada veći broj ispitanika (više od 40) pokuša istovremeno da pristupi aplikaciji za adaptivno testiranje, primenjen je decentralizovani pristup. Prekidi koji mogu da nastanu u radu zbog nestabilnosti računarske mreže, povećali bi nervozu kod ispitanika i negativno uticali na krajnji ishod testa, te stoga svaki računar koji se koristi za testiranje ima instaliranu aplikaciju i bazu pitanja. Tokom trajanja testa, pristup mreži bio je blokiran kako bi se onemogućilo da ispitanici međusobno razmenjuju podatke, a računarski stolovi i monitori bili su postavljeni tako da ispitanik nema dobar pregled na ekrane studenata koji sede pored njega.

## 7.2. Postupak ispitivanja i metodologija

U cilju provere znanja studenata napravljen je test od 20 pitanja sa višestrukim odgovorima. Na svako pitanje bilo je ponuđeno pet odgovora, od kojih je samo jedan bio tačan. Pri formulisanju distraktora za svako pitanje, autor se trudio da maksimalno poštuje smernice koje su date za njihovo pisanje [119]. Kontrolna grupa koja se proučava na tradicionalan način, test je radila na papiru, pitanja su odabrana iz baze pitanja ali po autorovom izboru, vodeći računa da budu zastupljena pitanja iz sva tri klastera težine. Eksperimentalna grupa radila je test uz pomoć aplikacije za testiranje i test je bio adaptivan. Vreme predviđeno za rešavanje testa bilo je trideset minuta za obe grupe ispitanika. Ručno ocenjivanje primenjeno je za klasičan test, dok je za eksperimentalnu grupu ocenjivanje rezultata testa izvršeno automatski pomoću računara. Bodovanje je rađeno u rasponu od 0 do 100. Kako bi se upoznali sa računarskim adaptivnim testom, studenti eksperimentalne grupe imali su priliku da vrše samoprocenu znanja i za to se koristila posebna baza pitanja sa 45 pitanja (po 15 pitanja iz svakog klastera) koja se nisu nalazila u bazi pitanja koja se koristila za procenu znanja. Studenti koji rade test na računaru mogli su da vide samo jedno pitanje i pet ponuđenih odgovora. Studenti su birali odgovor preko *radio buttona*, a zatim je bilo potrebno da kliknu na jedan *command button* kako bi se zabeležio njihov odgovor i kako bi prešli na sledeće pitanje. Sve dok se ne aktivira *command button* koji predstavlja prelazak na sledeće pitanje, studenti su u mogućnosti da promene svoj izbor odgovora koji smatraju tačnim. Nakon prelaska na sledeće pitanje ne postoji mogućnost izmene već datog odgovora na neko od prethodnih pitanja. Takođe, nije moguće preskočiti neko pitanje pa se kasnije vratiti na njega kako bi se dao odgovor.

Za netačan odgovor nisu računati negativni poeni bazirajući se na istraživanju [19] prema kome negativno bodovanje više kažnjava studente koji su pripremljeniji za test jer nastoje da na svako pitanje daju odgovor. Takođe, u radu [46] ukazuje se na to da negativno bodovanje diskriminiše ženski pol jer su žene opreznije u pogledu pogađanja.

Studenti eksperimentalne grupe testove su radili u računarskoj laboratoriji u kojoj su se i održavali časovi nastave. Pre svakog pristupanja testiranju na računaru, svi računari su provereni u pogledu tehničke ispravnosti, kako bi se osiguralo da je neophodan softver ispravno instaliran i kako bi se proverilo da su karakteristike monitora prihvatljive za nesmetan rad.

## 7.3. Analiza rezultata

U analizi rezultata primenjen je t-test koji predstavlja statistički postupak, a odnosi se na testiranje statistički značajnosti razlike između dve aritmetičke sredine pojava koje se testiraju. Statistička obrada podataka urađena je primenom programa *Microsoft Excel* iz programskog paketa *Microsoft Office 2010*.

Akademске 2009/10. godine u istraživanju je učestvovalo 108 studenata koji su slušali predmet Objektivno orijentisano programiranje i 83 studenta koji su slušali predmet

Java. Studenti su metodom slučajnog izbora bili dodeljeni eksperimentalnoj ili kontrolnoj grupi. Nastavnim planom i programom za oba predmeta je predviđeno da se rade dva obavezna kolokvijuma tokom semestra, čije je uspešno polaganje uslov izlaska na ispit. Prvi eksperiment dizajniran je tako da otkrije da li postoje razlike u postignutim rezultatima studenata koji rade računarski adaptivni test i onih koji rade test na papiru, kao i da se detektuje da li postoji razlika u vremenu potrebnom da se test reši. Vreme predviđeno za rešavanje testa za obe grupe limitirano je na 30 minuta.

Za predmet Objektno orijentisano programiranje kontrolna grupa imala je 60 studenata, a eksperimentalna 48. Kontrolnoj grupi koja je radila test na papiru, zadaci su podeljeni ali su dobili instrukcije da ne počnu sa radom dok svi studenti ne dobiju svoj primerak testa i dok se ne da signal za početak rešavanja testa. Za svakog pojedinačnog studenta kontrolne grupe na papiru sa testom u trenutku predaje rada zabeleženo je koliko mu je vremena bilo potrebno za rešavanje testa. Za studente eksperimentalne grupe, sama aplikacija beležila je koliko je vremena bilo potrebno studentu da odgovori na svako pojedinačno pitanje, a na osnovu toga je određeno i zbirno vreme trajanja testa. Statistička analiza rezultata prvog kolokvijuma data je u tabeli 7.3.

Tabela 7.3: Statistička analiza I kolokvijuma OOP 2009/2010

Rezultati I kolokvijuma OOP 2009/2010			
Kontrolna grupa		Eksperimentalna grupa	
n	60	n	46
Srednja vrednost	46.33	Srednja vrednost	52.56
Standardna greška	2.94	Standardna greška	4.02
Medijan	50	Medijan	55
Standardna devijacija	22.79	Standardna devijacija	27.87
t	-1.28		

Na osnovu rezultata datih u tabeli 7.3, može se videti da je srednja vrednost rezultata eksperimentalne grupe ( $\bar{X}=52.56$ ,  $SD=27.87$ ) veća za 6.23 poena od kontrolne grupe ( $\bar{X}=46.33$ ,  $SD=22.79$ ). Analiza podataka koja je rađena primenom t-testa dala je vrednost  $|t| = 1.28$ . Dobijena t-vrednost manja je od  $t_{95\%} = 1.96$ . Na osnovu ovog rezultata može se tvrditi sa sigurnošću od 95% i rizikom od 5%, da ne postoji statistički značajna razlika u postignutim rezultatima studenata kontrolne i eksperimentalne grupe za I kolokvijum iz predmeta Objektno orijentisano programiranje.

Drugi kolokvijum iz predmeta Objektno orijentisano programiranje radio se pet nedelja nakon prvog, a statistička analiza ovog kolokvijuma data je u tabeli 7.4.

Tabela 7.4: Statistička analiza II kolokvijuma OOP 2009/2010

Rezultati II kolokvijuma OOP 2009/2010			
Kontrolna grupa		Eksperimentalna grupa	
n	60	n	48
Srednja vrednost	38.5	Srednja vrednost	50.24
Standardna greška	2.84	Standardna greška	3.66
Medijan	37.6	Medijan	50
Standardna devijacija	21.98	Standardna devijacija	25.35
t	-2.66		

Na osnovu rezultata datih u tabeli 7.4, može se videti da je srednja vrednost rezultata eksperimentalne grupe ( $\bar{X}=50.24$ ,  $SD=25.35$ ) veća za 11.74 poena od kontrolne grupe ( $\bar{X}=38.5$ ,  $SD=21.97$ ). Analiza podataka koja je rađena primenom t-testa dala je vrednost  $|t| = 2.66$ . Dobijena t-vrednost veća je od  $t_{95\%}=1.96$ , kao i od  $t_{99\%}=2.61$ . Na osnovu ove vrednosti može se tvrditi sa sigurnošću od 99% i rizikom od 1%, da postoji statistički značajna razlika u postignutim rezultatima studenata kontrolne i eksperimentalne grupe za II kolokvijum iz predmeta Objektivno orijentisano programiranje.

Za predmet Java ukupan broj studenata koji je učestvovao u istraživanju akademske 2009/10. godine bio je 83. Kontrolnu grupu činila su 43 studenta, a eksperimentalnu 40. Statistička analiza rezultata I kolokvijuma iz Jave, data je u tabeli 7.5.

Tabela 7.5: Statistička analiza I kolokvijuma Java 2009/2010

Rezultati I kolokvijuma Java 2009/2010			
Kontrolna grupa		Eksperimentalna grupa	
n	43	n	40
Srednja vrednost	49.42	Srednja vrednost	60.75
Standardna greška	4.53	Standardna greška	3.56
Medijan	50	Medijan	57.5
Standardna devijacija	29.71	Standardna devijacija	22.55
t	-2.01		

Na osnovu rezultata datih u tabeli 7.5, vidi se da je srednja vrednost rezultata eksperimentalne grupe ( $\bar{X}=60.75$ ,  $SD=22.55$ ) veća za 11.33 poena od kontrolne grupe ( $\bar{X}=49.42$ ,  $SD=29.71$ ). Analiza podataka koja je rađena primenom t-testa dala je vrednost  $|t| = 2.01$ . Dobijena t-vrednost veća je od  $t_{95\%}=1.96$ , a manja od  $t_{99\%}=2.61$ . Na osnovu ovog rezultata može se tvrditi sa sigurnošću od 95% i rizikom od 5%, da postoji statistički značajna razlika u postignutim rezultatima studenata kontrolne i eksperimentalne grupe za I kolokvijum iz predmeta Java.

Statistička analiza rezultata II kolokvijuma iz Jave za akademsku 2009/2010. godinu data je u tabeli 7.6.

Tabela 7.6: Statistička analiza II kolokvijuma Java 2009/2010

Rezultati II kolokvijuma Java 2009/2010			
Kontrolna grupa		Eksperimentalna grupa	
n	43	n	40
Srednja vrednost	58.60	Srednja vrednost	68.9
Standardna greška	3.35	Standardna greška	3.72
Medijan	55	Medijan	70
Standardna devijacija	21.99	Standardna devijacija	23.55
t	-2.08		

Na osnovu rezultata datih u tabeli 7.6, vidi se da je srednja vrednost rezultata eksperimentalne grupe ( $\bar{X}=68.9$ ,  $SD=23.55$ ) veća za 10.3 poena od kontrolne grupe ( $\bar{X}=58.60$ ,  $SD=21.99$ ). Analiza podataka koja je rađena primenom t-testa dala je vrednost  $|t| = 2.08$ . Dobijena t-vrednost veća je od  $t_{95\%} = 1.96$ , a manja od  $t_{99\%} = 2.61$ . Na osnovu ovog rezultata može se tvrditi sa sigurnošću od 95% i rizikom od 5%, da postoji statistički značajna razlika u postignutim rezultatima studenata kontrolne i eksperimentalne grupe za II kolokvijum iz predmeta Java.

Akadske 2010/2011. godine, u istraživanju je učestvovao 91 student koji sluša predmet Objektno orijentisano programiranje u letnjem semestru I godine. Studenti su metodom slučajnog izbora bili podeljeni u kontrolnu grupu sa 46 studenata i eksperimentalnu grupu sa 45 studenata. Rezultati prvog kolokvijuma koji se radio sedme nedelje nastave letnjeg semestra dati su u tabeli 7.7.

Tabela 7.7: Statistička analiza I kolokvijuma OOP 2010/2011

Rezultati I kolokvijuma OOP 2010/2011			
Kontrolna grupa		Eksperimentalna grupa	
n	46	n	45
Srednja vrednost	35.78	Srednja vrednost	48.98
Standardna greška	4.47	Standardna greška	4.59
Medijan	25	Medijan	40
Standardna devijacija	29.99	Standardna devijacija	30.51
t	-2.0579		

Na osnovu rezultata datih u tabeli 7.7, može se videti da je srednja vrednost rezultata eksperimentalne grupe ( $\bar{X}=48.98$ ,  $SD=30.51$ ) veća za 13.2 poena od kontrolne grupe ( $\bar{X}=35.78$ ,  $SD=29.99$ ). Analiza podataka koja je rađena primenom t-testa dala je vrednost  $|t| = 2.06$ . Dobijena t-vrednost veća je od  $t_{95\%} = 1.96$ , a manja od  $t_{99\%} = 2.61$ . Na osnovu ovog rezultata može se tvrditi sa sigurnošću od 95% i rizikom od 5%, da postoji statistički značajna razlika u postignutim rezultatima studenata kontrolne i eksperimentalne grupe za I kolokvijum iz predmeta Objektno orijentisano programiranje.

Deskriptivna statistika drugog kolokvijuma iz predmeta Objektivno orijentisano programiranje koji se radio trinaeste nedelje nastave u letnjem semestru 2010/2011. godine, data je u tabeli 7.8.

Tabela 7.8: Statistička analiza II kolokvijuma OOP 2010/2011

Rezultati II kolokvijuma OOP 2010/2011			
Kontrolna grupa		Eksperimentalna grupa	
n	46	n	45
Srednja vrednost	41.51	Srednja vrednost	54.18
Standardna greška	5.02	Standardna greška	3.35
Medijan	28	Medijan	52
Standardna devijacija	33.67	Standardna devijacija	22.24
t	-2.0898		

Na osnovu rezultata datih u tabeli 7.8, može se videti da je srednja vrednost rezultata eksperimentalne grupe ( $\bar{X}=54.18$ ,  $SD=22.24$ ) veća za 12.67 poena od kontrolne grupe ( $\bar{X}=41.51$ ,  $SD=33.67$ ). Analiza podataka koja je rađena primenom t-testa dala je vrednost  $|t| = 2.09$ . Dobijena t-vrednost veća je od  $t_{95\%} = 1.96$ , a manja od  $t_{99\%} = 2.61$ . Na osnovu ovog rezultata može se tvrditi sa sigurnošću od 95% i rizikom od 5%, da postoji statistički značajna razlika u postignutim rezultatima studenata kontrolne i eksperimentalne grupe za II kolokvijum iz predmeta Objektivno orijentisano programiranje.

Akadske 2010/2011. godine, u istraživanju je učestvovalo 70 studenata II godine koji slušaju predmet Java u letnjem semestru. Studenti su metodom slučajnog izbora bili podeljeni u kontrolnu grupu i eksperimentalnu grupu sa po 35 studenata. Rezultati prvog kolokvijuma koji se radio sedme nedelje nastave dati su u tabeli 7.9.

Tabela 7.9: Statistička analiza I kolokvijuma Java 2010/2011

Rezultati I kolokvijuma Java 2010/2011			
Kontrolna grupa		Eksperimentalna grupa	
n	35	n	35
Srednja vrednost	55.59	Srednja vrednost	64.26
Standardna greška	3.25	Standardna greška	4.30
Medijan	55	Medijan	65
Standardna devijacija	18.98	Standardna devijacija	25.08
t	-1.60		

Na osnovu rezultata datih u tabeli 7.9, može se videti da je srednja vrednost rezultata eksperimentalne grupe ( $\bar{X}=64.24$ ,  $SD=25.08$ ) veća za 8.67 poena od kontrolne grupe ( $\bar{X}=55.59$ ,  $SD=18.98$ ). Analiza podataka koja je rađena primenom t-testa dala je vrednost  $|t| = 1.60$ . Dobijena t-vrednost manja je od  $t_{95\%} = 1.96$ . Na osnovu ovog rezultata može se tvrditi sa sigurnošću od 95% i rizikom od 5%, da ne postoji statistički

značajna razlika u postignutim rezultatima studenata kontrolne i eksperimentalne grupe za I kolokvijum iz predmeta Java.

Drugi kolokvijum iz predmeta Java radio se trinaeste nedelje nastave u letnjem semestru akademske 2010/2011. godine. Deskriptivna statistika rezultata data je u tabeli 7.10.

Tabela 7.10: Statistička analiza II kolokvijuma Java 2010/2011

Rezultati II kolokvijuma Java 2010/2011			
Kontrolna grupa		Eksperimentalna grupa	
<b>n</b>	35	<b>n</b>	35
<b>Srednja vrednost</b>	56.47	<b>Srednja vrednost</b>	68.09
<b>Standardna greška</b>	3.18	<b>Standardna greška</b>	3.98
<b>Medijan</b>	55	<b>Medijan</b>	70
<b>Standardna devijacija</b>	18.53	<b>Standardna devijacija</b>	23.26
<b>t</b>	<b>-2.28</b>		

Na osnovu rezultata datih u tabeli 7.10, može se videti da je srednja vrednost rezultata eksperimentalne grupe ( $\bar{X}=68.09$ ,  $SD=23.26$ ) veća za 11.62 poena od kontrolne grupe ( $\bar{X}=56.47$ ,  $SD=18.53$ ). Analiza podataka koja je rađena primenom t-testa dala je vrednost  $|t| = 2.28$ . Dobijena t-vrednost veća je od  $t_{95\%} = 1.96$ , a manja od  $t_{99\%} = 2.61$ . Na osnovu ovog rezultata može se tvrditi sa sigurnošću od 95% i rizikom od 5%, da postoji statistički značajna razlika u postignutim rezultatima studenata kontrolne i eksperimentalne grupe za II kolokvijum iz predmeta Java.

### 7.3.1. Diskusija rezultata

t-test koji je primenjen na dve grupe studenata (kontrolna i eksperimentalna) omogućava poređenje postignutih poena na kolokvijumima, koji predstavljaju obavezni deo polaganja ispita. Tabela 7.11 daje prikaz rezultata po akademskim godinama za I i II kolokvijum, za predmete Objektno orijentisano programiranje i Java.

Tabela 7.11: Statistička analiza I i II kolokvijuma 2009/2010 i 2010/2011

	OOP 2009/2010							
	K1				K2			
	n	$\bar{X}$	$\bar{X}_K - \bar{X}_E$	t	n	$\bar{X}$	$\bar{X}_K - \bar{X}_E$	t
Kontrolna grupa	60	46.33	6.23	-1.28	60	38.5	11.74	-2.66
Eksperimentalna grupa	46	52.56			46	50.24		

	Java 2009/2010							
	K1				K2			
	n	$\bar{X}$	$\bar{X}_K - \bar{X}_E$	t	n	$\bar{X}$	$\bar{X}_K - \bar{X}_E$	t
Kontrolna grupa	43	49.42	11.33	-2.01	43	58.60	10.3	-2.08
Eksperimentalna grupa	40	60.75			40	68.9		

	OOP 2010/2011							
	K1				K2			
	n	$\bar{X}$	$\bar{X}_K - \bar{X}_E$	t	n	$\bar{X}$	$\bar{X}_K - \bar{X}_E$	t
Kontrolna grupa	46	35.78	13.2	-2.06	43	41.51	12.67	-2.09
Eksperimentalna grupa	45	48.98			40	54.18		

	Java 2010/2011							
	K1				K2			
	n	$\bar{X}$	$\bar{X}_K - \bar{X}_E$	t	n	$\bar{X}$	$\bar{X}_K - \bar{X}_E$	t
Kontrolna grupa	35	55.59	8.67	-1.6	35	56.47	11.62	-2.28
Eksperimentalna grupa	35	64.26			35	68.09		

Značajnost razlika postignutih rezultata studenata kontrolne i eksperimentalne grupe data je u tabeli 7.12.

Tabela 7.12: Značajnost razlika kontrolne i eksperimentalne grupe

Kurs	Značajnost razlika rezultata dva uzorka	
OOP K1 2010	$ t  < t_{95\%}$	$1.28 < 1.98$
OOP K2 2010	$t_{99\%} <  t $	$2.61 < 2.66$
OOP K1 2011	$t_{95\%} <  t  < t_{99\%}$	$1.98 < 2.06 < 2.61$
OOP K2 2011	$t_{95\%} <  t  < t_{99\%}$	$1.98 < 2.09 < 2.61$
JAVA K1 2010	$t_{95\%} <  t  < t_{99\%}$	$1.98 < 2.01 < 2.61$
JAVA K2 2010	$t_{95\%} <  t  < t_{99\%}$	$1.98 < 2.08 < 2.61$
JAVA K1 2011	$ t  < t_{95\%}$	$1.6 < 1.98$
JAVA K2 2011	$t_{95\%} <  t  < t_{99\%}$	$1.98 < 2.28 < 2.61$



Na osnovu podataka iz tabele 7.12 može se zaključiti sledeće:

- u šest slučajeva od osam, može da se odbaci hipoteza i da se kaže sa sigurnošću od 95% i rizikom od 5% da postoji statistički značajna razlika između rezultata kontrolne i eksperimentalne grupe, tj. da su studenti koji su radili računarski adaptivni test postigli bolji rezultat u odnosu na studente koji su test radili na konvencionalan način.
- u dva slučaja ne postoji razlog na osnovu koga bi se odbacila glavna hipoteza da ne postoji statistički značajna razlika u rezultatima kontrolne i eksperimentalne grupe.

Pored glavne hipoteze, u istraživanju je definisana i pomoćna hipoteza koja glasi da računarski adaptivni test smanjuje vreme potrebno za proveru usvojenosti gradiva kod studenata. U teoriji, računarski adaptivni test može značajno da smanji vreme potrebno za testiranje uz zadržavanje kvaliteta samog procesa testiranja u poređenju sa standardnim testovima koji se rade na papiru [161]. Računarski adaptivni test može da smanji vreme administracije testa i više od 50% uz zadržavanje nivoa pouzdanosti [122].

Za svakog pojedinačnog studenta kontrolne grupe zabeleženo je koliko mu je vremena bilo potrebno za rešavanje testa. Za studente eksperimentalne grupe, sama aplikacija beležila je koliko je vremena bilo potrebno studentu da odgovori na svako pojedinačno pitanje, a na osnovu toga je određeno i zbirno vreme trajanja testa svakog studenta eksperimentalne grupe. Maksimalno vreme predviđeno za rešavanje testa za obe grupe bilo je 30 minuta. Kako uvek ima studenata koji izađu na ispit potpuno nespremni, sa željom da vide „kako izgledaju pitanja“, iz analize su izbačena vremena i kontrolne i eksperimentalne grupe koja su bila kraća od 10% vremena trajanja testa. Statistički podaci koji se odnose na vreme trajanja testa kontrolne i eksperimentalne grupe dati su u tabeli 7.13.

Tabela 7.13: Trajanje testa kontrolne i eksperimentalne grupe

Kurs	Prosečno vreme trajanja testa [min]					
	$\bar{t}_K$	$\bar{t}_E$	SDK	SDE	$ \bar{t}_K - \bar{t}_E $	t
OOP K1 2010	25.28	21.93	3.60	3.04	3.35	4.26**
OOP K2 2010	26.24	22.97	3.38	3.34	3.27	4.13**
OOP K1 2011	25.48	21.75	3.74	4.31	3.73	3.97**
OOP K2 2011	24.92	22.46	3.85	4.69	2.46	2.46
JAVA K1 2010	23.92	21.62	5.49	5.45	2.3	1.81*
JAVA K2 2010	24.45	21.92	4.82	5.68	2.53	2.06
JAVA K1 2011	26.37	22.62	3.41	3.88	3.75	4.40**
JAVA K2 2011	25.70	23.00	5.39	4.91	2.7	2.25

Nakon izvršene t-analize za slučaj velikog uzorka, na osnovu statističkih pokazatelja datih u tabeli 7.13, može se zaključiti sledeće:

- u sedam slučajeva od osam, postoji značajna statistička razlika u prosečnom vremenu trajanja testa u slučaju kontrolne i eksperimentalne grupe. Od ovih sedam slučajeva, u četiri slučaja (obeleženi sa \*\* u tabeli) može se tvrditi sa sigurnošću od 99% i rizikom od 1% da postoji statistički značajna razlika između prosečnog vremena trajanja testa kontrolne i eksperimentalne grupe, jer je  $t > t_{99\%}$ . U tri slučaja sa sigurnošću od 95% i rizikom od 5% može se tvrditi da postoji značajna statistička razlika između kontrolne i eksperimentalne grupe koja se odnosi na prosečno vreme trajanja testa.
- Samo u jednom slučaju (obeležen sa \* u tabeli) ne postoji statistički značajna razlika u prosečnom vremenu trajanja testa kontrolne i eksperimentalne grupe.
- Najveća razlika u vremenu trajanja testa kontrolne i eksperimentalne grupe je 3.75 minuta, odnosno 12.5% od ukupnog vremena koje je predviđeno za test.
- Najmanja razlika u prosečnom vremenu trajanja testa kontrolne i eksperimentalne grupe je 2.3 minuta, odnosno 7.67% od ukupnog vremena predviđenog za rešavanje testa.

Na osnovu ovih rezultata, može se reći da je potvrđena pomoćna hipoteza da računarski adaptivni test skraćuje vreme potrebno za proveru usvojenosti gradiva kod studenata.

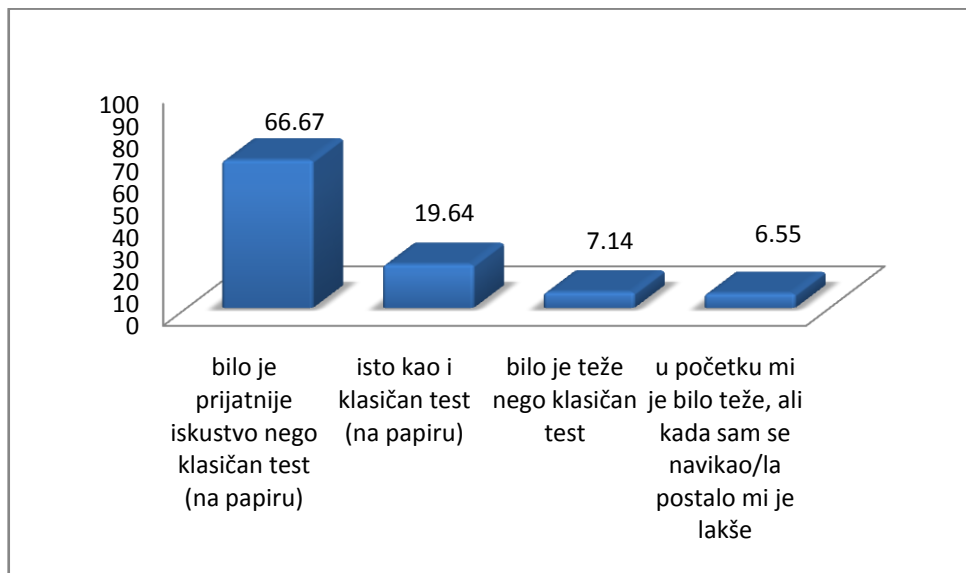
U studiji [23] autori kao razlog za kraće prosečno vreme trajanja testa na računaru navode to da je studentima potrebno više vremena da napišu odgovor koristeći olovku, nego što im je potrebno da kliknu na jedan od ponuđenih odgovora kada rade test na računaru. Međutim, u našem slučaju ne može se govoriti o tome, jer su studenti kontrolne grupe, kao i studenti eksperimentalne grupe, imali ponuđene odgovore za svako pitanje, te nije bilo potrebe za bilo kakvim pisanjem nego samo zaokruživanje tačnog odgovora, što ne traje mnogo duže od klika mišem. Kod rešavanja konvencionalnog testa, studenti jedan deo vremena potroše na proveravanje rešenja, kao i na promenu već datog odgovora. Kod računarski podržanog adaptivnog testa, studenti nemaju mogućnost da se vrate unazad i izmene neki odgovor, kao ni da pregledaju sva pitanja i date odgovore pre finalne „predaje“ testa na proveru od strane računara, te je to sigurno jedan od faktora koji utiče na kraće trajanje testa.

Prema rezultatima statističke analize može se reći da postoji značajna razlika u prosečnom vremenu trajanja testa, međutim teško je odgovoriti na pitanje da li ta razlika ima i praktičnog značaja. U svim posmatranim slučajevima, najveća razlika u vremenu bila je 3.75 minuta, odnosno 12.75% od ukupnog vremena trajanja testa. Kako sam test ne traje duže od 30 minuta, ova razlika od četiri minuta nije od velikog praktičnog značaja za test ove dužine. Međutim, ukoliko test traje duže ova razlika u vremenima trajanja testa, odnosno kraće trajanje računarskog adaptivnog testa za nešto više od 10% mogla bi biti od značaja. Rezultat dobijen ovim istraživanjem ne može se generalizovati za proizvoljno trajanje testa, nego su potrebna daljnja istraživanja u slučaju dužeg trajanja testa.

Svi studenti koji su činili eksperimentalnu grupu (ukupno 168 studenata) i koji su polagali računarski adaptivni test popunjavali su upitnik koji se odnosi na njihove stavove i mišljenje o adaptivnom testiranju. Zatraženo je od studenata eksperimentalne grupe da odaberu tvrdnju koja najbolje opisuje njihovo mišljenje o korišćenju računara za polaganje kolokvijuma, odnosno ispita. Dve trećine studenata izjasnilo se da je polaganje testa na računaru bilo mnogo prijatnije iskustvo nego konvencionalan test. Blizu 20% studenata smatra da ne postoji razlika u odnosu na konvencionalan test. Oko 7% studenata odgovorilo je da je polaganje testa na računaru teže od konvencionalnog testa, a jednak procenat studenata odgovorio je da im se u početku činilo teže dok se nisu navikli, a onda im je bilo lakše.

Komentar jednog od studenata bio je, da ga je u početku bio strah da će slučajno kliknuti na pogrešan odgovor, ali da je taj strah nestao nakon datih odgovora na nekoliko početnih pitanja i da mu se ovakav način polaganja testa čini mnogo boljim od klasičnog.

Jedan od studenata dao je i ovakav komentar: „Navikli smo da polazemo kolokvijume i ispite na papiru, svaka promena od uobičajenog načina polaganja loše utiče na mene. U početku me je bio strah i samo sam razmišljao kako ću uopšte uspeti da ostvarim nepohodan minimum poena potreban za polaganje kolokvijuma. Posle 5 minuta, kada sam video pitanja koja sam dobijao, uopšte više nisam razmišljao o tome da polazem test na računaru.“ Rezultati odgovora na ovo pitanje prikazani su na slici 7.5.



Slika 7.5: Rezultati mišljenja studenata eksperimentalne grupe o adaptivnom testiranju

Sledeće pitanje koje je postavljeno studentima, bilo je da procene kako su uradili test na računaru. Oko 80% studenata smatra da su test uradili najbolje što su mogli, dok nešto više od 13% studenata smatra da je test uradilo podjednako dobro kao i da su radili na konvencionalan način.

Neki od komentara studenata na ovo pitanje bili su da na njihov uspeh ne utiče to da li polažu test na računaru ili na papiru: „Znaš, ili ne znaš. Učiti moraš ako želiš da položiš, a da li polažeš ispit na računaru ili na papiru nema uticaja. Mada, meni se lično više sviđa da polažem test na računaru.“

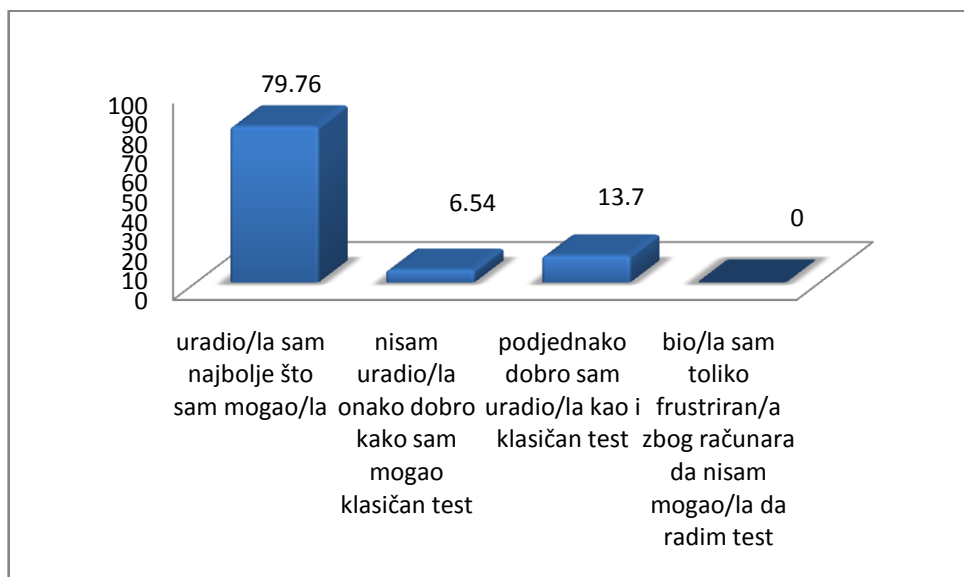
„Gradivo koje moraš da spremiš da bi izašao na kolokvijum isto je i kada polažeš ispit i na papiru i na računaru. Moja ocena ne bi bila ništa bolja od ove koju sam dobio kada sam polagao test na računaru.“

„Sviđa mi se ideja da se pitanja biraju prema mojim sposobnostima. Uradio sam test najbolje što sam mogao, ne bih bolje uradio ni da sam radio test na papiru.“

„Mislim da je mnogo bolje ovako, kada se ispit polaže na računaru. Pitanja se daju jedno po jedno i lakše mogu da se skoncentrišem kako bih odgovorio. Kada radim test na papiru pročitam sva pitanja i onda me uhvati panika da ne znam dovoljno da bih položio ispit. Često izađem a da ništa i ne uradim, predam prazan papir. Voleo bih i sledeći put da polažem kolokvijum na računaru, mnogo mi je bolje tako. Jednostavnije je i lakše.“

Jedan student je napisao kao komentar sledeće: „Ne mogu da se opredelim ni za jedan ni drugi način. Mada, kada bolje razmislim bolje je da polažem na papiru. Tako sam siguran da ne može doći do kvara računara, ili nekog drugog problema sa računarom pa da nestanu svi odgovori koje sam dao.“

Grafički prikaz raspodele odgovora na ovo pitanje prikazan je na slici 7.6.



Slika 7.6: Odgovori studenata na pitanje da procene kako su uradili test

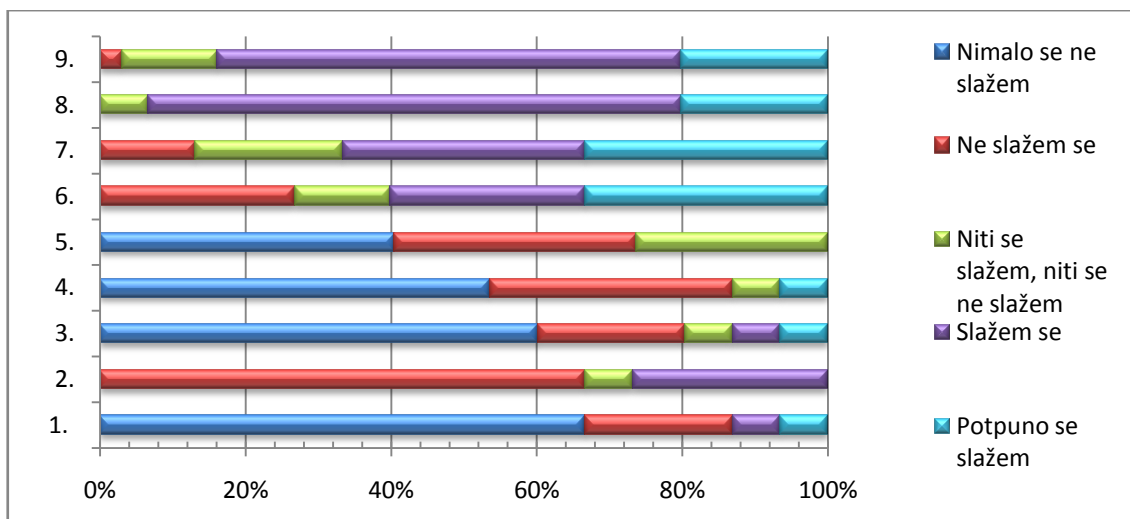
Jedno od pitanja postavljeno studentima, bilo je i da li im je smetalo što su za pojedina pitanja morali da koriste skrolbar kako bi pročitali postavljeno pitanje i našli odgovor na njega. Najveći procenat studenata, 46.43%, smatralo je da im je korišćenje skrolbara olakšalo pronalaženje tačnog odgovora, dok jedna trećina studenata smatra da je bilo potpuno jednako kao i u slučaju testa na papiru.

Studenti eksperimentalne grupe trebalo je i da popune upitnik sa nizom tvrdnji koje je trebalo da rangiraju koristeći Likertovu skalu od 1 (Nimalo se ne slažem) do 5 (Potpuno se slažem). Odgovori studenata na ovaj upitnik prikazani su u tabeli 7.14 i grafički prikazani na slici 7.7.

Tabela 7.14: Odgovori studenata na upitnik prema Likertovoj skali

1 Nimalo se ne slažem, 2 Ne slažem se, 3 Niti se slažem, niti se ne slažem, 4 Slažem se, 5 Potpuno se slažem

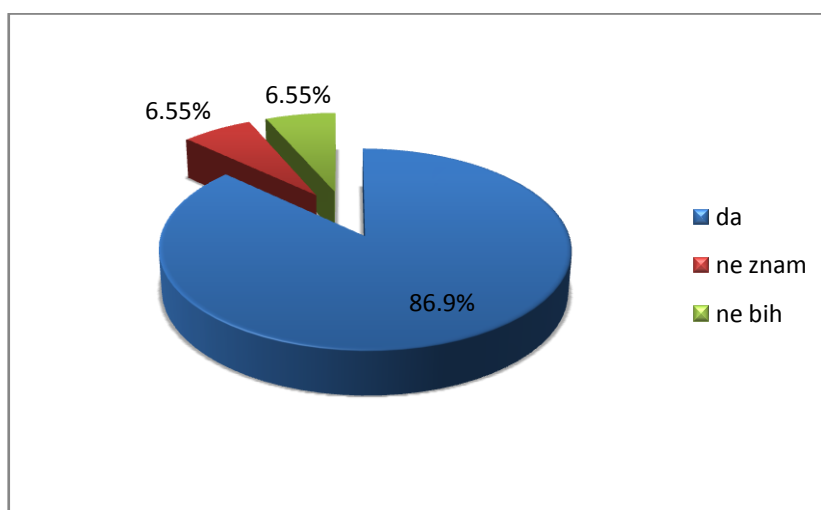
Tvrdnja	1	2	3	4	5
T1. Kada moram da polažem kolokvijum/ispit na računaru osećam se daleko neprijatnije nego kada isti zadatak radim na papiru	66.67%	20.24%		6.55%	6.55%
T2. Tokom testa koji sam radio/la na računaru osećao/la sam veliku tremu		66.67%	6.55%	26.78%	
T3. Tokom testa bio/bila sam toliko nervozan/a da sam zaboravio/a činjenice koje sam znao-la	60.12%	20.24%	6.55%	6.55%	6.55%
T4. Loše mi je i pri samoj pomisli da moram da radim neki zadatak na računaru	53.57%	33.33%	6.55%		6.55%
T5. Pre samog testa bio sam veoma zabrinut/a zbog činjenice da ću ga polagati na računaru	40.48%	33.33%	26.19%		
T6. Mnogo lakše polažem one ispite koji se polažu na računaru		26.78%	13.09%	26.78%	33.33%
T7. Polaganje kolokvijuma putem adaptivnog testa na računaru manje je stresno nego kada se radi test na papiru		13.09%	20.24%	33.33%	33.33%
T8. Interfejs je jednostavan za korišćenje			6.55%	73.21%	20.24%
T9. Moje znanje realno je ocenjeno		2.98%	13.1%	63.68%	20.24%



Slika 7.7: Odgovori studenata na tvrdnje T1-T9

Studenti su mogli i da daju svoje komentare, pa su tako imali zamerku na nemogućnost provere već datih odgovora i eventualne izmene, dok je bilo i kritika na račun izgleda grafičkog korisničkog interfejsa sa komentarom „da bi trebalo poraditi na njemu.“ Zamerki je bilo i na nemogućnost sagledavanja svih pitanja odjednom, kako bi se napravila najbolja strategija odgovaranja na pitanja, tj. odgovaranja na preskok a ne po redosledu kako se pitanja pojavljuju.

Na pitanje da li bi preporučili adaptivni način polaganja testa svojim kolegama, velika većina studenata odgovorila je potvrdno 86.90%, dok je podjednak broj studenata neopredeljen ili ne bi preporučio. Grafički prikaz odgovora dat je na slici 7.8.



Slika 7.8: Odgovor studenata na pitanje da li bi adaptivni test preporučili kolegama

Gledano u celini, na osnovu odgovora studenata na upitnik, vidi se da postoji pozitivan stav prema računarski podržanom adaptivnom testiranju. Više od 60% studenata

smatra da je ovakav način polaganja kolokvijuma manje stresan nego kada se provera znanja radi na papiru. Iz komentara koje su sami studenti napisali, može se videti da im se ovakav način polaganja testa više sviđa jer se prilagođava njihovim sposobnostima, te se time ujedno smanjuje frustracija kada su pitanja isuviše teška za pojedine studente, ili dosada koja se javlja kod dobrih studenata kada im postavljena pitanja ne predstavljaju izazov za njihovo znanje. I u jednom, i u drugom slučaju, može da dođe do demotivacije i nezadovoljstva, kao i izostanka želje za napretkom.

Uzimajući u razmatranje sve date odgovore studenata eksperimentalne grupe, kao i njihove komentare, možemo da kažemo da je potvrđena hipoteza da *računarski adaptivni test povećava motivisanost studenata tokom polaganja testa*. Korišćenje novih tehnologija stvara dodatni faktor motivacije kod studenata [80], ali prema [106] motivacija vremenom opada te je potrebno uložiti napore da do toga ne dođe.

Nakon završenog testiranja urađena je statistička evaluacija pitanja, te je prema statistici odgovora studenata izvršena preraspodela jednog broja pitanja po klasterima težine. Kod jednog broja pitanja studenti su ukazali na greške u ponuđenim distraktorima koji su izazivali nedoumicu, te su ta pitanja ispravljena kako bi bila jednoznačna.

## 8. Zaključak

Unazad nekoliko decenija primena računara u obrazovanju postala je veoma zastupljena, pa tako i njihova primena u proveru znanja. Najčešći oblik testova koji se rade na računaru su linearni testovi fiksne dužine koji za svakog ispitanika daju ista pitanja u određenom redosledu, a rezultat zavisi od broja pitanja na koje je ispitanik tačno odgovorio.

Za razliku od linearnih računarskih testova, računarski adaptivni testovi dinamički prilagođavaju test prema individualnim sposobnostima pojedinca. Adaptivno testiranje zasnovano je na činjenici da konvencionalan ispit (neadaptivan test) može da zataji u proceni napretka studenta. Na teška pitanja na testu studenti obično ne daju odgovor zbog nedostatka određenih veština i znanja. Za razliku od teških pitanja, na lakim pitanjima ponekad dolazi do greške, jer studenti ne pročitaju dovoljno pažljivo tekst pitanja. Stoga, postavljanje pitanja pogrešnog nivoa težine može da rezultira pogrešnom procenom napretka studenta.

Adaptivna teorija testiranja daje rešenje ovog problema uvođenjem fleksibilnosti u proces testiranja, u kome se pitanja biraju uzimajući u obzir psihološki profil studenta i njegov nivo znanja. Kod adaptivnog testa, redosled prezentacije pitanja nije unapred određen kao u slučaju konvencionalnog testa, već se bazira na algoritmu koji vodi računa o težini pitanja na koje student odgovara, kao i o tačnosti datog odgovora. Ukoliko student odgovori tačno na postavljeno pitanje, prvo sledeće pitanje je za jedan nivo teže od prethodnog pitanja. Ako je student odgovori netačno, prvo sledeće pitanje je za jedan nivo težine lakše. Ovaj proces odabira odgovora iz baze pitanja prema sposobnostima i znanju studenta, nastavlja se sve dok se ne odgovori na predviđen broj pitanja (ukoliko je test fiksne dužine), ili dok se ne postigne unapred određen nivo pouzdanosti određivanja nivoa znanja studenta.

Istraživanje koje je predmet ove doktorske disertacije imalo je za cilj da potvrdi hipotezu da *ne postoji statistički značajna razlika u postignutim rezultatima studenata koji polažu test primenom računarski adaptivnog testa i koji polažu test na konvencionalan način (papir i olovka)*. Pored glavne hipoteze, postavljene su i dve pomoćne hipoteze a koje se odnose na skraćivanje vremena potrebnog za rešavanje testa, kao i na povećanje motivisanosti studenata tokom adaptivnog testiranja. Ukupan broj studenata koji je učestvovao u istraživanju bio je 352, što čini 48.35% ukupnog broja studenata Visoke tehničke škole strukovnih studija u Subotici.

Na osnovu rezultata istraživanja sa sigurnošću od 95% i rizikom od 5%, može se reći da u šest od osam ispitivanih slučajeva, postoji statistički značajna razlika između rezultata kontrolne i eksperimentalne grupe, tj. da su studenti koji su radili računarski adaptivni test postigli bolji rezultat u odnosu na studente koji su test radili na konvencionalan način. Prosečna razlika u postignutim poenima je oko 10 poena u korist studenata eksperimentalne grupe. U dva slučaja ne postoji razlog na osnovu koga bi se odbacila



glavna hipoteza da ne postoji statistički značajna razlika u rezultatima kontrolne i eksperimentalne grupe.

Statistička analiza podataka pokazuje da postoji statistički značajna razlika u rezultatima kontrolne i eksperimentalne grupe, te se može odbaciti glavna hipoteza da *ne postoji statistički značajna razlika u postignutim rezultatima studenata koji polažu test primenom računarski adaptivnog testa i koji polažu test na konvencionalan način (papir i olovka).*

Razmatranjem rezultata istraživanja koja se odnose na skraćanje vremena trajanja testa potrebnog za proveru znanja studenata u slučaju adaptivnog testiranja može se konstatovati da postoji značajna razlika u prosečnom vremenu trajanja testa, međutim teško je odgovoriti na pitanje da li ta razlika ima i praktičan značaj. U svim posmatranim slučajevima, najveća razlika u vremenu bila je 3.75 minuta, odnosno 12.75% od ukupnog vremena trajanja testa. Kako sam test ne traje duže od 30 minuta, ova razlika nije od velikog praktičnog značaja za test ove dužine trajanja. Međutim, ukoliko test traje duže, ova razlika u vremenima trajanja testa, odnosno kraće trajanje računarskog adaptivnog testa za nešto više od 10% mogla bi biti od značaja. Rezultat dobijen ovim istraživanjem ne može se generalizovati za proizvoljno trajanje testa, nego su potrebna daljna istraživanja u slučaju dužeg trajanja testa. Na osnovu ovih rezultata, možemo da kažemo da je *potvrđena pomoćna hipoteza da računarski adaptivni test skraćuje vreme potrebno za proveru usvojenosti gradiva kod studenata.*

Gledano u celini, na osnovu odgovora studenata na upitnik koji se odnosi na njihovo mišljenje o adaptivnom testu, vidi se da postoji pozitivan stav prema računarski podržanom adaptivnom testiranju. Na pitanje da li bi računarski adaptivni test preporučili kolegama, 86.90% studenata eksperimentalne grupe odgovorilo je potvrdno, dok je podjednak broj studenata neopredeljen ili ga ne bi preporučio. Više od 60% studenata smatra da je ovakav način polaganja kolokvijuma manje stresan nego kada se proveru znanja radi na papiru. Iz komentara koje su studenti napisali, može se videti da im se ovakav način polaganja testa više sviđa jer se prilagođava njihovim sposobnostima, te se time ujedno smanjuje frustracija kada su pitanja isuviše teška za pojedine studente, ili dosada koja se javlja kod dobrih studenata kada im postavljena pitanja ne predstavljaju izazov. I u jednom, i u drugom slučaju, može da dođe do demotivacije i nezadovoljstva, kao i do izostanka želje za napretkom. Uzimajući u razmatranje sve date odgovore studenata eksperimentalne grupe, kao i njihove komentare, možemo da kažemo da je *potvrđena pomoćna hipoteza da računarski adaptivni test povećava motivisanost studenata tokom polaganja testa.*

Ovom disertacijom postignut je osnovni cilj istraživanja da se proverom znanja na računaru primenom adaptivnih testova može preciznije proceniti znanje svakog pojedinca i postići veće zadovoljstvo i motivisanost ispitanika.

Najznačajniji naučni doprinos ove doktorske disertacije je primena informaciono komunikacionih tehnologija za realizaciju podrške tradicionalnog obrazovanja, kao i

dokaz boljih kvantitativnih i kvalitativnih rezultata u savladavanju gradiva. Disertacija predstavlja zaokruženi proces istraživanja koji je obuhvatio dizajn i uvođenje novog načina evaluacije u obrazovanju inženjera iz specifične grupe predmeta. Nakon uočavanja novih zahteva i potreba u ovom segmentu obrazovanja, kreiran je i razvijen sistem baziran na savremenim informaciono-komunikacionim tehnologijama i sprovedena je formativna evaluacija u svim fazama razvoja. Proverene su postavljene istraživačke hipoteze i sagledani su pravci daljih istraživanja i primene razvijenog sistema.

Originalni doprinosi ovog rada su:

- sagledavanje problema u tradicionalnom obrazovnom procesu nastave programiranja u cilju razvoja programa za adaptivno testiranje primenom računara,
- razvoj konceptualnog modela za izradu računarskog adaptivnog testa,
- razvoj algoritma za računarski adaptivni test za kvalitetniju evaluaciju znanja,
- izbor zadataka i dodavanje u prethodno kreiranu bazu pitanja,
- primena adaptivnog testa u realnim uslovima nad reprezentativnim uzorkom studenata Visoke tehničke škole strukovnih studija u Subotici i prikaz rezultata primene.

Rezultati disertacije pružaju osnove za dalja istraživanja koja bi bila usmerena ka unapređenju aplikacije dodavanjem mogućnosti da se uz pitanje koriste i multimedijalni elementi (slika, zvuk, video). Takođe, unapređenje aplikacije treba da ide i u pravcu realizacije povratne sprege, koja bi studenta, pored indikacije da li je odgovor na postavljeno pitanje tačan ili ne, usmerila na deo gradiva u kome se nalazi odgovor na postavljeno pitanje. Inicijalna istraživanja za potrebe disertacije rađena su na Visokoj tehničkoj školi u Subotici, a u toku je i realizacija primene računarskog adaptivnog testa na Univerzitetu Obuda iz Budimpešte na Informatičkom fakultetu Nojman Janoš (mađarski, Óbudai Egyetem Neumann János Informatikai Kar).

Net generacija (engl. *net generation*) koja se rodila i raste okružena tehnologijom, zahteva i savremene načine provere znanja, koji će uključiti i upotrebu tehnologija koje su bliske studentima 21. veka. Računarski podržani adaptivni testovi predstavljaju težnju da se uskladi način procene znanja sa inovacijama koje tehnologija omogućava i da se napravi sistem procene znanja pogodan za sve studente. Kako je upotreba mobilnih uređaja, kao što su mobilni telefoni i PDA (engl. *Personal Digital Assistant*), u značajnom porastu u toku poslednje decenije, pažnju bi trebalo usmeriti i u pravcu upotrebe ovih uređaja u procesu učenja i provere znanja. Zbog svojih dimenzija i značajnih tehničkih mogućnosti, mobilni uređaji mogu biti pogodna platforma za razvoj računarski baziranih testova, pa samim tim i adaptivnih testova. Primenom tehnologije koja je bliska studentima može se povećati njihova motivisanost za izvršenje zadataka koji se traže od njih, a koji su neophodni za polaganje ispita.

Rezultati rada na ovoj doktorskoj disertaciji objavljeni su u više radova u časopisima i saopšteni na nekoliko naučnih skupova u zemlji i inostranstvu.

## Prilog

### A. Anketa

1. Pol: a. muško b. žensko
2. Godine starosti:
3. Završena \_\_\_\_\_ (upisati ime škole i mesto),  
smer: \_\_\_\_\_  
TroGodišnja          b. četvorogodišnja škola
4. Prosek ocena:
5. Smer upisan na VTŠ: \_\_\_\_\_
6. Broj poena na prijemnom: \_\_\_\_\_
7. Broj položenih ispita do sada: \_\_\_\_\_
8. Broj nepoloženih ispita: \_\_\_\_\_
9. Ukupan broj kredita do sada: \_\_\_\_\_
10. Prosek ocena tokom studija: \_\_\_\_\_
11. Da li ste imali nastavu programiranja u srednjoj školi:          a. da          b. ne

Ako je odgovor **Da**, napišite koliko časova nedeljno ste imali programiranje: \_\_\_\_\_

12. Koji programski jezik ste učili u srednjoj školi:

- a. Basic
- b. Pascal
- c. C#
- d. C++
- e. Delphi
- g. nismo učili ni jedan
- h. neki drugi (koji?)C

13. Koliko dugo imate računar?

- a. Nemam ga
- b. Do 3 meseca
- c. Do 6 meseci
- d. Do 1 god
- e. Do 2 god
- f. Do 3 god
- g. Do 4 god
- h. Do 5 god
- i. 5 do 10 god
- j. 10 do 15 god

14. Koliko dnevno vremena provodite za računarom?

- a. Do 1h

- b. Do 2h
  - c. Do 3h
  - d. Do 4h
  - e. Do 5h
  - f. Do 6h
  - g. Do 7h
  - h. Do 8h
  - i. 8h do 12h
  - j. Preko 12h
15. Koliko vremena dnevno provodite na Internetu?
- a. Do 30 minuta
  - b. Do 1h
  - c. Do 2h
  - d. Do 3h
  - e. Do 4h
  - f. Do 5h
  - g. Do 6h
  - h. Do 7h
  - i. Do 8h
  - j. 8h do 12h
  - k. Preko 12h
16. Koji servis na Internetu najviše koristite?
- a. e-mail
  - b. Traženje dodatnog materijala za učenje
  - c. youtube
  - d. facebook
  - e. nešto drugo (napisati šta)
17. Koja tvrdnja najbolje opisuje Vaš stav o korišćenju računara?
- a. Osećam da najbolje mogu da radim na računaru
  - b. Osećam se ugodno kada koristim računar
  - c. Korišćenje računara mi se ne dopada
  - d. Frustriran/a sam kada koristim računar
18. Koja tvrdnja najbolje opisuje Vaše mišljenje o korišćenju računara za polaganje kolokvijuma/ispita:
- a. Bilo je prijatnije iskustvo nego klasičan test (na papiru)
  - b. Isto kao i klasičan test (na papiru)
  - c. Bilo je teže nego klasičan test (na papiru)
  - d. U početku mi je bilo teže, ali kada sam se navikao/la postalo mi je lakše

19. Kada sam radio/la test na računaru:
- Uradio/la sam najbolje što sam mogao/la
  - Nisam uradio/la onako dobro kako sam mogao/la da je klasičan test
  - Podjednako dobro sam uradio/la kao i klasičan test
  - Bio/la sam toliko frustriran/a zbog računara da nisam mogao/la da radim test
20. Vreme predviđeno za test na računaru:
- Bilo je kraće nego vreme za klasičan test
  - Bilo je duže nego za klasičan test
  - Jednako kao za klasičan test
  - Nisam primetio/la koliko je trajao test na računaru
21. Primena skrolbara da bi ste pročitali postavljeno pitanje i našli odgovor na njega:
- Činilo je lakše pronalaženje tačnog odgovora
  - Bilo je teže naći odgovor zbog skrolovanja nego da je test na papiru
  - Jednako kao i kada radim test na papiru
  - Nisam koristio skrolbar da bih ponovo pročitao pitanje i našao odgovor

Molim Vas da pažljivo pročitate svaku tvrdnju. **Označite stepen Vašeg slaganja ili neslaganja**, služeći se skalom od 1 do 5, pri čemu brojevi znače sledeće:

1 - Nimalo se ne slažem, 2 - Ne slažem se, 3 - Niti se slažem, niti se ne slažem, 4 - Slažem se, 5 - Potpuno se slažem

a. Kada moram da polažem kolokvijum/ispit na računaru osećam se daleko neprijatnije nego kada isti zadatak radim na papiru	1	2	3	4	5
b. Tokom testa koji sam radio/la na računaru osećao/la sam veliku tremu	1	2	3	4	5
c. Tokom testa bio/bila sam toliko nervozan/a da sam zaboravio/a činjenice koje sam znao-la	1	2	3	4	5
d. Loše mi je i pri samoj pomisli da moram da radim neki zadatak na računaru	1	2	3	4	5
e. Pre samog testa bio sam veoma zabrinut/a zbog činjenice da ću ga polagati na računaru	1	2	3	4	5
f. Mnogo lakše polažem one ispite koji se polažu na računaru	1	2	3	4	5
g. Polaganje kolokvijuma putem testa na računaru manje je stresno nego kada se isti radi na papiru	1	2	3	4	5

## B. Programski kod

```
function test

clc
clear all
svetlo_plava=[.81 .86 .99];
figura = figure('NumberTitle','off',...
    'Name','Test',...
    'Resize','off',...
    'MenuBar','none',...
    'Position',[320 173 800 555]);
panel_dod= uipanel('Parent',figura,...
    'Position',[0 0 1 1],...
    'BackgroundColor','white');
uputstvo = {'UPUTSTVO';...
    ''};...

    'Korisnik daje svoj odgovor tako sto selektuje opition
button pored jednog od 5';...
    'ponudjenih odgovora a zatim pritisne Enter ili klikne
na dugme -Sledece pitanje-';...
    ''};...
    'Nakon poslednjeg odgovorenogpitanja prikazuju se
konacni rezultati kao i konacan';...
    'rezultat. Ove rezultate je moguće sacuvati u obliku
teksta tako sto se iz menija';...
    'izabere opcija -Sacuvaj rezultate rada-';...
    ''};...
    'Maksimalno dozvoljeno vreme da se uradi test je 30
min';...
    ''};...
    'S V E N A J B O L J E !';...
    ''};
directions = uicontrol(panel_dod,...
    'Style','text',...
    'String',uputstvo,...
    'BackgroundColor','white',...
    'HorizontalAlignment','left',...
    'FontName','arial',...
    'FontWeight','bold',...
    'FontSize',12,...
    'Position',[70,65,650,440]);
pocetak_testiranja = uicontrol(panel_dod,...
    'Style','pushbutton',...
    'Tag','pocetak_testiranja',...
    'String','Pocetak testa',...
    'FontName','arial',...
    'FontSize',12,...
    'FontWeight','bold',...
    'Position',[580,40,130,35],...
    'Callback',{@start},...
    'KeyPressFcn',{@start},...
    'KeyPressFcn',{@keypress});
br_pitanja = uicontrol(panel_dod,...
    'Style','text',...
    'String','',...
    'FontName','arial',...
    'FontWeight','bold',...
    'HorizontalAlignment','left',...
    'FontSize',12,...
```

```

        'BackgroundColor', 'white', ...
        'Position', [25 555 200 20]);
postavka_pitanja = uicontrol(panel_dod, ...
    'Tag', 'postavka_pitanja', ...
    'Style', 'edit', ...
    'String', '', ...
    'Position', [260, 185, 370, 370], ...
    'FontName', 'arial', ...
    'FontSize', 12, ...
    'FontWeight', 'bold', ...
    'Max', 35, 'Min', 0, ...
    'BackgroundColor', 'white', ...
    'HorizontalAlignment', 'left', ...
    'Visible', 'off');
submit = uicontrol(panel_dod, ...
    'Tag', 'submit', ...
    'Style', 'pushbutton', ...
    'String', 'Sledece pitanje', ...
    'FontName', 'arial', ...
    'FontSize', 12, ...
    'FontWeight', 'bold', ...
    'Position', [25, 185, 130, 28], ...
    'Visible', 'off', ...
    'Callback', {@sub}, ...
    'KeyPressFcn', @sub_key);
meni = uimenu(figura, 'Label', 'Menu');
ex1 = uimenu(meni, 'Tag', 'Unos_podataka', 'Label', 'Unesi svoje
podatke', 'Callback', {@unos_podataka_callback});
ex2 = uimenu(meni, 'Tag', 'Cuvanje_rezultata', 'Label', 'Sacuvaj
rezultate rada', 'Callback', {@cuvanje_podataka_callback});
ex =
uimenu(meni, 'Tag', 'izlaz', 'Label', 'Izlaz', 'Separator', 'on', 'UserData',
1, 'Callback', {@izlaz_callback});

for redni_broj=1:5
option_button(redni_broj) = uicontrol(panel_dod, ...
    'Tag', num2str(redni_broj), ...
    'Style', 'radiobutton', ...
    'String', 'neki string', ...
    'FontName', 'arial', ...
    'FontWeight', 'bold', ...
    'FontSize', 12, ...
    'BackgroundColor', 'white', ...
    'Value', 0, ...
    'Visible', 'off', ...
    'Position', [25, 30, 800, 25], ...
    'Callback', {@option});
end

global dat_odgovor
dat_odgovor=0;
global izlaz
izlaz=0;
global userdt
userdt=1;
global taster korisnikov_odgovor grupa_pitanja tezina_pitanja
korisnikov_odgovor='0';
global ime prezime br_indeksa a b total_marks
try
[ime prezime br_indeksa]=unos_podataka;

```

```

        set(figura, 'Name', ['Student: ', ime, ' ', prezime, ',
Br.indeksa:', br_indeksa])
    end
    uicontrol(pocetak_testiranja)

    function cuvanje_podataka_callback(source, eventdata)

sacuvaj_rezultate(ime, prezime, br_indeksa, a, tezina_pitanja, b, total_marks);
    end

    function unos_podataka_callback(source, eventdata)
        try
            [ime prezime br_indeksa]=unos_podataka;
            set(figura, 'Name', ['Student: ', ime, ' ', prezime, ',
Br.indeksa:', br_indeksa])
        end
        uicontrol(pocetak_testiranja)
    end

    function izlaz_callback(source, eventdata)
        set(ex, 'UserData', 0);
        delete(figura)
    end

    function sub_key(source, eventdata)
        keypressed=eventdata.Key;
        if strcmp(keypressed, 'return')
            taster=str2num(korisnikov_odgovor);
            dat_odgovor=1;
        end
    end

    function sub(source, eventdata)
        dat_odgovor=1;
        taster=str2num(korisnikov_odgovor);
    end

    function option(source, eventdata)
        korisnikov_odgovor=get(source, 'Tag');
        switch korisnikov_odgovor
            case '1'
                set(option_button(2), 'Value', 0)
                set(option_button(3), 'Value', 0)
                set(option_button(4), 'Value', 0)
                set(option_button(5), 'Value', 0)
            case '2'
                set(option_button(1), 'Value', 0)
                set(option_button(3), 'Value', 0)
                set(option_button(4), 'Value', 0)
                set(option_button(5), 'Value', 0)
            case '3'
                set(option_button(1), 'Value', 0)
                set(option_button(2), 'Value', 0)
                set(option_button(4), 'Value', 0)
                set(option_button(5), 'Value', 0)
            case '4'
                set(option_button(1), 'Value', 0)
                set(option_button(2), 'Value', 0)
                set(option_button(3), 'Value', 0)
        end
    end

```



```

        set(option_button(5), 'Value', 0)
    case '5'
        set(option_button(1), 'Value', 0)
        set(option_button(2), 'Value', 0)
        set(option_button(3), 'Value', 0)
        set(option_button(4), 'Value', 0)
    end
    uicontrol(submit)
end

function keypress(source, eventdata)
    keypressed=eventdata.Key;
    if strcmp(keypressed, 'return')
        grupa_pitanja=ones(1,20);
        pocetak_testa
    end
end

function start(source, eventdata)
    grupa_pitanja=ones(1,20);
    pocetak_testa
end

ose0 = axes ('Parent', figura, ...
            'Position', [0 0 1 .03], ...
            'Visible', 'on');
axis off
xdata0 = [0;0;4;4];
ydata0 = [0;1;1;0];
zdata0 = ones(4,1);

okvir=patch(xdata0,ydata0,zdata0, 'w', 'EdgeColor', 'w', 'FaceColor', 'white');

x=800/(20*800);
ose = axes ('Parent', figura, ...
            'Position', [0 0 x .03], ...
            'Visible', 'on');
axis off

xdata = [0;0;4;4];
ydata = [0;1;1;0];
zdata = ones(4,1);

progres=patch(xdata,ydata,zdata, 'w', 'EdgeColor', 'w', 'FaceColor', 'w');

function pocetak_testa
    set(figura, 'Position', [320 150 840 600])
    set(ex1, 'Enable', 'off')
    set(directions, 'Visible', 'off')
    set(pocetak_testiranja, 'Visible', 'off')
    set(postavka_pitanja, 'Visible', 'on')
    set(submit, 'Visible', 'on')

set(option_button(1), 'Visible', 'on', 'Position', [25,150,800,25])

set(option_button(2), 'Visible', 'on', 'Position', [25,120,800,25])
set(option_button(3), 'Visible', 'on', 'Position', [25,90,800,25])
set(option_button(4), 'Visible', 'on', 'Position', [25,60,800,25])
set(option_button(5), 'Visible', 'on', 'Value', 1)

```

```

set(okvir, 'EdgeColor', 'b', 'FaceColor', 'w');
set(progres, 'EdgeColor', 'b', 'FaceColor', svetlo_plava);

[a b]=final_test;

try
set(br_pitanja, 'Visible', 'off')
set(postavka_pitanja, 'Visible', 'off')
set(submit, 'Visible', 'off')
set(option_button(1), 'Visible', 'off')
set(option_button(2), 'Visible', 'off')
set(option_button(3), 'Visible', 'off')
set(option_button(4), 'Visible', 'off')
set(option_button(5), 'Visible', 'off')
set(ose0, 'Visible', 'off')
set(ose, 'Visible', 'off')
delete(progres)
delete(okvir)
set(figura, 'Position', [470 200 550 555])

total_marks=int2str(totaling(a));

Text1 = uicontrol(panel_dod, 'Style', ...
    'text', 'String', ['Rezultati za svako pitanje:'], ...
    'BackgroundColor', 'white', ...
    'HorizontalAlignment', 'left', ...
    'FontName', 'arial', ...
    'FontWeight', 'bold', ...
    'FontSize', 12, ...
    'Visible', 'on', ...
    'Position', [35, 510, 250, 25]);

columnname1 = {'Tezina
pitanja', 'Tacno/Netacno', 'sec/pitanju'};
t1 = uitable('Units', 'normalized', 'Position', ...
    [0 0.001 0.58 0.9], ...
    'ColumnName', columnname1, ...
    'ColumnWidth', {90.5}, ...
    'Visible', 'on'); % [0 0.001 0.4035 0.9]
dat=[tezina_pitanja' a' b'];
set(t1, 'Data', dat)

final_score=strvcat('Krajnji rezultat:', [ ' ', total_marks
'/100']);
Text3 = uicontrol(panel_dod, 'Style', ...
    'text', 'String', final_score, ...
    'BackgroundColor', 'white', ...
    'HorizontalAlignment', 'left', ...
    'FontName', 'arial', ...
    'FontWeight', 'bold', ...
    'HorizontalAlignment', 'center', ...
    'FontSize', 14, ...
    'Visible', 'on', ...
    'Position', [330, 100, 200, 50]);
end
end

function [ans_array, time_arr]=final_test
[a b c t1]=ask_qn(1, 1, grupa_pitanja, ones(1, 3), 1);

```

```

ans_array = b;
time_arr = t1;

end

function
[next_question_set,ans_array,question_status_new,qn_time]=ask_qn(question_set,ask_1,com_arr,question_status,ques_num_start)

    qn_time=[];
    ans_array=[];
    tezina_pitanja=[];

    for(i=1:length(com_arr))
        try
            set(option_button(1),'Value',0)
            set(option_button(2),'Value',0)
            set(option_button(3),'Value',0)
            set(option_button(4),'Value',0)
            set(option_button(5),'Value',0)

            deciding_factor=ask_1;

question_set=normalize_qno(question_set,deciding_factor,1,3);
            tezina_pitanja=[tezina_pitanja question_set];

            ask_1_char=int2str(question_set);
            ask_1_char_type=question_type(com_arr(i));
            ques_char=['PITANJE BR. ' int2str(ques_num_start) '/'
num2str(length(grupa_pitanja) ':');
            set(br_pitanja,'String',ques_char);
            ask_1_char_txt=[ask_1_char_type '_' ask_1_char '.txt'];

            x=ques_num_start*(100/(length(grupa_pitanja)*100));
set(ose,'Position',[0 0 x .03]);
            axis off

            [ask_1
q_time]=ask(ask_1_char_txt,question_status(com_arr(i),question_set));
            qn_time=[qn_time q_time];
            clc;
            ans_array=[ans_array ask_1];
question_status(com_arr(i),question_set)=question_status(com_arr(i),question_set)+35

            ques_num_start=ques_num_start+1;

            if (ques_num_start==30)
                set(submit,'String','Prikazi rezultat')
            end
        end
    end
    next_question_set=question_set;
    question_status_new=question_status;
end

```

```

function
[output_check,q_time]=ask(input_text_file_name,start_line_number)
try
q_time=[];
y=input_text_file_name;

fid=fopen(y);
lin_count=0;
output_check=0;
pitanja = '';

    for(k=1:35+start_line_number-1) %
        tline = fgetl(fid);
        if strcmp(tline,'')
            pitanja=strvcat(pitanja,' ');
        else
            pitanja=strvcat(pitanja,tline);
        end
        lin_count=lin_count+1;
    end

colwidth = 74;

set(postavka_pitanja,'Units','characters')

string={pitanja(start_line_number:(start_line_number+28),:)};
outstring1 = textwrap(postavka_pitanja,string,colwidth);
newpos1 = get(postavka_pitanja,'Position');

newpos1(3) = colwidth;
newpos1(4) = length(outstring1)+1;
set(postavka_pitanja,'String',outstring1,'Position',newpos1)

set(option_button(1),'String',pitanja((start_line_number+29),:))%
set(option_button(2),'String',pitanja((start_line_number+30),:))%
set(option_button(3),'String',pitanja((start_line_number+31),:))%
set(option_button(4),'String',pitanja((start_line_number+32),:))%
set(option_button(5),'String',pitanja((start_line_number+33),:))%

    if(rem(lin_count,35)==0)%
        num_tline=tline-48;
        tic
while(dat_odgovor==0)
        pause(0.02)
        userdt=get(ex,'UserData');
        if (userdt==0)
            delete(figura)
            return
        end
    end
    dat_odgovor=0;
    q_time=[q_time toc];
    if(taster==num_tline)
        output_check=1;
    end
end

```

```

                end
            end
        end

end

function [ime prezime br_indeksa]=unos_podataka

    try
        answer={' ',' ',' '};
        dijalog = inputdlg({'Ime:', 'Prezime:', 'Br.Indeksa:'}, 'STUDENT:', [1
28; 1 28;1 28],answer);

        ime = dijalog{1};
        prezime = dijalog{2};
        br_indeksa = dijalog{3};
    end
end

function question_str=question_type(number)

if(number==1)
    question_str='pitanja';
end

function output=normalize_qno(present_qno,deciding_factor,min,max)

if(deciding_factor==1)
    output=present_qno+1;
else
    output=present_qno-1;
end

if(output<min)
    output=min;
end
if(output>max)
    output=max;
end

function
sacuvaj_rezultate(ime,prezime,br_indeksa,a,tezina_pitanja,b,total_marks)

    try
        [filename, pathname, filterindex] = uiputfile( ...
            {'*.txt', 'Text File (*.txt)';...
            '*.doc', 'Word Document (*.doc)';...
            '*.*', 'All Files (*.*)'},...
            'Save as');

        fid=fopen(filename, 'a');

        rb=[];
        for(i=1:length(a))
            rb=[rb i];
        end
    end
end

```

```

matrica=[rb' tezina_pitanja' a' round(b)'];

fprintf(fid, '\n-----\n');
fprintf(fid, ['\nSTUDENT: ', ime, ' ', prezime, ', Br.indeksa: ',
br_indeksa, '\n']);
fprintf(fid, ['\nKonacni rezultat: ' [ total_marks '/100']]);

fprintf(fid, ['\n\n\nREZULTATI ZA SVAKO PITANJE:']);
fprintf(fid, '\n-----\n');
fprintf(fid, '\n-- r.broj--      -- tezina pitanja -- --
odgovori-- -- vreme(s/pitanju) --\n');
fprintf(fid, '\n-----\n');

for i=1:9
    fprintf(fid, ' ');
    for j=1:4
        fprintf(fid, '%d', matrica(i,j));
        fprintf(fid, ' ');
    end
    fprintf(fid, '\n-----\n');
    fprintf(fid, '\n');
end

for i=10:30
    fprintf(fid, ' ');
    for j=1:4
        fprintf(fid, '%d', matrica(i,j));
        fprintf(fid, ' ');
    end
    fprintf(fid, '\n-----\n');
    fprintf(fid, '\n');
end

fclose(fid);
end
end

function total_mark=totaling(a)

mark_arr=[30 30 30 30 30 25 25 25 25 25 20 20 20 20 20 15 15 15 15 15
10 10 10 10 10 5 5 5 5 5];
wrong_arr=-[15 15 15 15 15 12 12 12 12 12 10 10 10 10 10 8 8 8 8 8 6 6
6 6 6 4 4 4 4 4];

for(i=1:length(a))
    if(a(i)==0)
        mark_arr(i)=wrong_arr(i);
    end
end
marks_scored_correct=sum(mark_arr);
%-----
-----

```

```
b=1:length(a);
c=[];
for(i=1:length(a))
    if(a(i)==1)
        c=[c b(i)];
    end
end
c=[0 c length(a)+1];
for(i=1:length(c)-1)
    d(i)=c(i+1)-c(i)-1;
end
d_nesortirano=d;
d=fliplr(sort(d));
e=[];
for(i=1:length(d))
    if(d(i)>=3)
        e(i)=d(i);
    end
end
e=e*(-20);
marks_cut_repeat=sum(e);
%-----
-----
total_mark=marks_scored_correct+marks_cut_repeat+425;
if(total_mark<200)
    total_mark=200;
end
if(total_mark>800)
    total_mark=800;
end
total_mark=round(total_mark/10)*10;
%-----
-----
```

## Literatura

1. Ahoniemi, T., Lahtinen, E., Erkkola, T. (2008). Fighting the Student Dropout Rate with an Incremental Programming Assignment, *Conferences in Research and Practice in Information Technology*. <http://crpit.com/confpapers/CRPITV88Ahoniemi2.pdf>
2. Al-A'ali, M. (2007). Implementation of an Improved Adaptive Testing Theory. *Educational Technology & Society*, 10 (4), pp. 80-94.
3. Al-Amri, S. (2008). Computer-based testing vs. paper-based testing: a comprehensive approach to examining the comparability of testing modes, *Essex Graduate Student Papers in Language & Linguistics* 10, pp. 22-44.
4. Ala-Mutka, K. (2005). A survey of automated assessment approaches for programming assignments. *Computer Science Education* 15, pp. 83-102
5. Alderson, J. C. (2000). Technology in Testing: the Present and the Future. *System*, Vol. 28, No. 4, pp. 593-603.
6. Al-Smadi, M. and Gütl, C. (2008). Past, Present and Future of e-Assessment: Towards a Flexible e-Assessment System, *Conference ICL2008, Villach, Austria*, pp. 1-8
7. Anđelić, S., Čekerevac, Z., Anđelić, D. (2011). Testing of students using specialized CAT model, *YUINFO 2011, Kopaonik*.
8. Anđelković, J. (2004). Procena znanja primenom kompjuterskog adaptivnog testiranja, IX Međunarodni simpozijum SymOrg 2004 „Menadžment - ključni faktor uspeha“, *Zbornik apstrakata str 239, Zbornik radova - (elektronsko izdanje) sekcija: Menadžment ljudskih resursa, Zlatibor*.
9. Anđelković, Z., Stević, Z. (2007). Informacione tehnologije u obrazovanju i proceni znanja, *INFOTEH-JAHORINA Vol. 6, Ref. E-IV-1*, p. 464-468.
10. Armstrong, S. J. (2000). The influence of individual cognitive style on performance in Management Education. *Educational Psychology*, 20 (3), pp 323-340.
11. Barker, T (2009) An automated feedback system based on adaptive testing: extending the model, *Proceedings of the Interactive Computer Aided Learning Conference, ICL2009, 23 - 25 September 2009, Villach/Austria*
12. Barker, T., (2008). Computer adaptive testing in higher education: the validity and reliability of the approach. IN: Khandia, F. (ed.). *12th CAA International Computer Assisted Assessment Conference : Proceedings of the Conference on 8th and 9th July 2008 at Loughborough University. Loughborough : Loughborough University*, pp. 25-40
13. Basu, A., Cheng, I., Prasad. M., Rao, G. (2007). Multimedia Adaptive Computer based Testing: An Overview, *ICME 2007 Special Session, Beijing*, pp.1850-1853.
14. Bayroff, A. G. (1964). Feasibility of a programmed testing machine (U.S. Army Personnel Research Office Research Study 6403). Washington, DC: U.S. Army Behavioral Science Research Laboratory.



15. Bayroff, A. G., Thomas, J. J., and Anderson, A. A. (1960). Construction of an experimental sequential item test (Research Memorandum 60-1). Washington, DC: Department of the Army, Personnel Research Branch.
16. Bergstrom, B. A., & Lunz, M. E. (1999). CAT for certification and licensure. In F. Drasgow & J. B. Olson-Bunchanan (Eds.), *Innovations in computerized assessment*, pp. 67-91. Mahwah, NJ: Lawrence Erlbaum Associates.
17. Binet, A., & Simon, Th. A. (1905). Méthode nouvelle pour le diagnostic du niveau intellectuel des anormaux. *L'Année Psychologique*, 11, 191-244.
18. Birnbaum, A. (1968). Some latent trait models and their use in inferring an examinee's ability. In F. M. Lord and M. R. Novick (Eds.), *Statistical theories of mental test scores* (chap.17-20). Reading, MA: Addison-Wesley.
19. Bjekić D., Papić Ž. (2005). *Ocenjivanje, priručnik za ocenjivanje u srednjem stručnom obrazovanju*, Ministarstvo prosvete i sporta Republike Srbije, Beograd.  
<http://www.vetserbia.edu.rs/Zbirka%20doc/Ocenjivanje.pdf>
20. Björnsson, J. K. (2008). *The PISA Computer Based Assessment of Science: What did we learn?* Educational Testing Institute. Pristupljeno 15. 11. 2010.  
[http://crell.jrc.ec.europa.eu/Presentations\\_Iceland%202008/Bj%C3%B6rnsson%20-%20CBAS.pdf](http://crell.jrc.ec.europa.eu/Presentations_Iceland%202008/Bj%C3%B6rnsson%20-%20CBAS.pdf).
21. Bliss, L.B. (1980). A test of Lord's assumption regarding examinee guessing behavior on multiple-choice tests using elementary school students, *Journal of Educational Measurement*, 17(2), pp.147–152.
22. Bloom, B.S. (Ed.) (1956). *Taxonomy of educational objectives: The classification of educational goals: Handbook I, Cognitive domain*. New York;Toronto: Longmans, Green.
23. Bocij, P. and Greasley, A. (1999). Can computer-based Testing Achieve Quality and Efficiency in Assessment? *International Journal of Educational Technology*, 1(1), pp. 1-18.
24. Bock, R.D., & Mislevy, R.J. (1982). Adaptive EAP estimation of ability in a microcomputer environment. *Applied Psychological Measurement*, 6, pp. 431-444
25. Brusilovsky, P. (2004). Knowledge Tree: A Distributed Architecture for Adaptive E-Learning, *Proceedings of the 13th World Wide Web Conference*, May 17-22, New York, New York, USA, pp. 104-113.
26. Carlson, R. D. (1994). Computer-Adaptive Testing: a Shift in the Evaluation Paradigm, *Journal of Educational Technology Systems*, 22(3), pp 213-224.
27. Chang, S. W., Ansley, T. N., Lin, S. H. (2000). Performance of item exposure control methods in computerized adaptive testing: Further explorations. Paper presented at the annual meeting of the American Educational Research Association, New Orleans. pp. 1-44. Pristupljeno 07.03.2011. <http://www.eric.ed.gov/PDFS/ED442837.pdf>.
28. Chatzopoulou, D.I., Economides A.A., (2010). Adaptive assessment of student's knowledge in programming courses, *Journal of Computer Assisted Learning*, Vol. 26, No.4, pp. 258–269
29. Cheng, I. and Basu, A. (2006). Improving multimedia innovative item types for computer based testing, *IEEE Symposium on Multimedia*, San Diego, pp. 557–566. doi: 10.1109/ISM.2006.92.

30. Conejo, R., Guzmán, E., Millán, E., Trella, M., Pérez-De-La-Cruz, J. L. & Ríos, A. (2004). SIETTE: A Web-Based Tool for Adaptive Testing, *International Journal of Artificial Intelligence in Education* 14, pp. 1 -33.
31. Conejo, R.; Millán, E.; Pérez-de-la-Cruz, J. L.; Trella, M. (2000) An Empirical Approach to On-Line Learning in SIETTE, *Lecture Notes in Computer Science*, 1839, pp. 605-614.
32. Coster, W., Haley, S., Ni P, Dumas, H., Fragala Pinkham MA (2008). Assessing self-care and social function using a computer adaptive testing version of the pediatric evaluation of disability inventory *Arch Phys Med Rehabil*, 89(4), pp.622-629.
33. Dave, O., D.&Dobele, T. (2007). First year courses in IT: a bloom rating. *Journal of Information Technology Education* 6, 347–359.
34. Davey, T. and Nering, M. (2002). Controlling Item Exposure and Maintaining Item Security. In *Computer-Based Testing: Building the Foundation for Future Assessments*, Edited by Craig N. Milles, Maria T. Potenza, John J. Fremer, and William C. Ward. Lawrence Erlbaum Associates, Publishers, Mahwah, New Jersey.
35. Davis, L. L. & Dodd, B.G. (2003). Item exposure constraints for testlets in the verbal reasoning section of the MCAT. *Applied Psychological Measurement*, 27(5), pp. 335-356.
36. Davis, L.L. (2004). Strategies for controlling item exposure in computerized adaptive testing with the generalized partial credit model. *Applied Psychological Measurement*, 28(3), pp. 165-185.
37. Drasgow, F. (2002). The work ahead: A psychometric infrastructure for computerized adaptive tests. In C.N. Mills, M.T. Potenza, J.J. Fremer, & W.C. Ward (Eds.), *Computer-based testing: Building the foundation for future assessments*, pp. 67–88. Hillsdale, NJ: Lawrence Erlbaum.
38. Egberink, I. J. L. & Veldkamp, B. P. (2007). The development of a computerized adaptive test for integrity. In D. J. Weiss (Ed.), *Proceedings of the 2007 GMAC Conference on Computerized Adaptive Testing*. [www.psych.umn.edu/psylabs/CATCentral/](http://www.psych.umn.edu/psylabs/CATCentral/)
39. Eggen, T. J. H. M. (2007). Choices in CAT models in the context of educational testing. In D. J. Weiss (Ed.), *Proceedings of the 2007 GMAC Conference on Computerized Adaptive Testing*. [www.psych.umn.edu/psylabs/CATCentral/](http://www.psych.umn.edu/psylabs/CATCentral/) [15.06.2011.]
40. Elhan, A. H., Öztuna, D., Kutlay, Ş, Küçükdeveci, A. A, and Tennant, A. (2008). An initial application of computerized adaptive testing (CAT) for measuring disability in patients with low back pain. *BMC Musculoskeletal Disorders* 2008, 9:166. doi:10.1186/1471-2474-9-166
41. Embretson, S. E. & Reise, S. P. (2000). *Item response theory for psychologists*. Mahwah, NJ: Lawrence Erlbaum Associates, Inc.
42. Fajgelj, S., Janičić, B. (2008). Kakaobejz: Makro za ajtem analizu dihotomno skorovanih ajtema-teorija ajtemskih odgovora, *Primenjena psihologija*, Vol. 1 (3-4), UDC 159.9.072.59:519.2, str. 223-241.
43. Fetzer, M., Dainis, A., Lambert, S., Meade, A. (2011). *Computer Adaptive Testing (CAT) in an Employment Context*, SHLPreVisor White Paper 2011, pp. 1-14
44. Flaugher, R. (2000). Item Pools. In Wainer, H. (Ed.) *Computerized adaptive testing: A Primer*. Mahwah, NJ: Erlbaum.

45. Fox, J. and Andersen, R., (2005). Using the R Statistical Computing Environment to Teach Social Statistics Courses, <http://www.unt.edu/rss/Teaching-with-R.pdf>
46. Fuerstner, I., Anisic, Z. (2010). Adaptive involvement of customers as co-creators in mass customization, In I. Fuerstner (Ed.), *Products and Services; from R&D to Final Solutions*, Sciyo, Rijeka, Hrvatska, ISBN: 978-953-307-211-1, pp. 179-198.
47. Gafni, N., Cohen, Y., Roded, K., Baumer, M., & Moshinsky, A. (2009). Applications of CAT in admissions to higher education in Israel: Twenty-two years of experience. In D. J. Weiss (Ed.), *Proceedings of the 2009 GMAC Conference on Computerized Adaptive Testing*. [www.psych.umn.edu/psylabs/CATCentral/](http://www.psych.umn.edu/psylabs/CATCentral/)
48. Galatanu, C.,-D., Barbieru, E. (2003). Docimological principles applied to the e-learning tests, *RoEduNet International Conference*, Iasi, Romania, pp.77-81.
49. Gibby, R., & McCloy, R. A. (2011). Computerized adaptive testing. In N. Tippins and S. Adler (Eds.), *Technology enhanced assessments*. San Francisco: Wiley.
50. Giesinger et al. (2011). Cross-cultural development of an item list for computer-adaptive testing of fatigue in oncological patients. *Health and Quality of Life Outcomes* 2011 9:19. doi:10.1186/1477-7525-9-19
51. Goldik, Z. (2008). Abandoning negative marking, *European Journal of Anaesthesiology*, 25(5), pp. 349-351.
52. Guyer, R. D. and Weiss, D. J. (2009). Effect of early misfit in computerized adaptive testing on the recovery of theta. In D. J. Weiss (Ed.), *Proceedings of the 2009 GMAC Conference on Computerized Adaptive Testing*. [www.psych.umn.edu/psylabs/CATCentral/](http://www.psych.umn.edu/psylabs/CATCentral/)
53. Hambleton, R. K., Swaminathan, H., & Rogers, H. J. (1991). *Fundamentals of Item Response Theory*. Newbury Park, CA: Sage Press.
54. Han, K. T. (2010). SimulCAT: Windows application that simulates computerized adaptive test administration. <http://www.hantest.net/simulcat> [23. 03. 2010.]
55. Harris, D. (1989). Comparison of 1-, 2-, and 3-parameter IRT models. *Educational Measurement: Issues and Practice*; pp. 35-41. <http://www.learnlab.org/research/wiki/images/2/22/Harris-article.pdf>
56. Harta, D. L., Cook, K. F., Mioduski, J. E., Teal, C., R., Crane, P. K. (2006). Simulated computerized adaptive test for patients with shoulder impairments was efficient and produced valid measures of function. *Journal of Clinical Epidemiology*. Volume 59, Issue 3, pp. 290-298.
57. Hartley, J., & Nicholls, L. (2008). Time of day, exam performance and new technology. *British Journal of Educational Technology*, 39, pp. 555-558. doi: 10.1111/j.1467-8535.2007.00768.x
58. Hatzilygeroudis, I., Koutsojannis, C., Papavlasopoulos, C., & Prentzas, J. (2006). Knowledge-based adaptive assessment in a Web-based intelligent educational system. Paper presented at the Sixth International Conference on Advanced Learning Technologies, July 5-7, 2006, Kerkrade, The Netherlands, pp. 1-5. <http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=8D51E147358389700ABBEE3199610F31?doi=10.1.1.143.3075&rep=rep1&type=pdf> [08. 02. 2011.]

59. Hollingsworth, J. (1960). Automatic graders for programming classes. *Communications of the ACM*, vol. 3, no. 10, pp. 528– 529.
60. Huang, S. X. (1996). A Content-Balanced Adaptive Testing Algorithm for Computer-Based Training Systems, *Lecture Notes in Computer Science*, 1086, pp. 306-314.
61. Hulin, C. L., Drasgow, F., Parsons, C. K. (1983). *Item response theory*. Dow Jones-Irwin, Homewood, IL.
62. Ismail, M. N., Ngah N. A., Umar I. N. (2010). Instructional Straregy in the Teaching of Computer Programming:a Need Assessment Analyses, *TOJET: The Turkish Online Journal of Educational Technology*, Volume 9, Issue 2, pp. 125-131
63. Jacobsen, J., Ackermann, R., Egüez, J., Ganguli, D., Rickard, P., Taylor, L. (2011). Design of a Computer-Adaptive Test to Measure English Literacy and Numeracy in the Singapore Workforce: Considerations, Benefits, and Implications, *Journal of Applied Testing Technology*, Volume 12, Special Issue on Adaptive Testing. pp. 1-26
64. Jekić, R. (2010). GUI za kompjuterizovano adaptivno testiranje izrađen u Matlabu, seminarski rad, VTŠ.
65. Jettmar, E. and Nass, C. (2002). Adaptive testing: effects on user performance, *Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves*. Minneapolis, Minnesota, USA, pp. 129-134.
66. Kantrowitz, T. M., Dawson, C. R., Fetzer, M. S. (2011). Computer Adaptive Testing (CAT): A Faster, Smarter, and More Secure Approach to Pre-Employment Testing, *J Bus Psychol* (2011) 26:227–232. doi: 10.1007/s10869-011-9228-3
67. Kingsbury, G. G., & Zara, A. R. (1989). Procedures for selecting items for computerized adaptive tests. *Applied Psychological Measurement*, 2, pp. 359-375.
68. Kingston, N., & Dorans, N. (1985). The analysis of item-ability regressions: an exploratory IRT model fit tool. *Applied Psychological Measurement*, 9, pp. 281-288.
69. Lahtinen, E., AlaMutka, K., Järvinen, H.-M. (2005). A Study of the Difficulties of Novice Programmers. *ITiCSE'05*, June 27–29, pp. 14-18. [http://student.brighton.ac.uk/mod\\_docs/cmisis/past%20papers/ci\\_modules/level%20\\_2/2006\\_07/ci215\\_cs2\\_2006.pdf](http://student.brighton.ac.uk/mod_docs/cmisis/past%20papers/ci_modules/level%20_2/2006_07/ci215_cs2_2006.pdf)
70. Lalović, R. (2009). O testovima znanja-Traganje za kvalitetnim modelom ocjenjivanja, Prosvjetni rad, List prosvjetnih, kulturnih i naučnih radnika Republike Crne Gore, broj 15-16, novembar 2009. Pristupljeno 19. 05. 2011. <http://www.prosvjetnirad.co.me/>
71. Lilley, M. & Barker, T. (2007). Students perceived usefulness of formative feedback for a computer-adaptive test, *Electronic Journal of e-Learning (EJEL)*, 5(1), Special Issue (ECEL 2006), pp. 31-38. <http://www.ejel.org/Volume-5/v5-i1/v5-i1-art-5.htm>
72. Lilley, M. (2007). The Development and Application of Computer Adaptive Testing in a Higher Education Environment, PhD thesis. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.100.7459&rep=rep1&type=pdf>
73. Lilley, M.; Barker, T. & Britton, C. (2005a). Automated feedback for a computer-adaptive test: A case study, *Proceedings of the 9th Computer-Assisted Assessment Conference*, Loughborough University, United Kingdom.
74. Lilley, M.; Barker, T. & Britton, C. (2005b). Learners perceived level of difficulty of a computer-adaptive test: A case study, *Proceedings of the 10th International Conference*

- on Human-Computer Interaction, Lecture Notes in Computer Science, 3585, pp. 1026-1029.
75. Lilley, M. and Barker, T. (2004). A Computer-Adaptive Test That Facilitates the Modification of Previously Entered Responses: An Empirical Study, J.C. Lester et al. (Eds.): ITS 2004, LNCS 3220, pp. 22–33. © Springer-Verlag Berlin Heidelberg.
  76. Lilley, M. and Barker, T. (2003). An Evaluation of a Computer Adaptive Test in a UK University Context, 7th CAA Conference 2003, pp. 171-182
  77. Lilley, M.; Barker, T. & Maia, M. (2003) Do Cognitive Styles of Learning Affect Student Performance in Computer-Adaptive Testing?, Proceedings of the 6th Simpósio de Administração da Produção, Logística e Operações Internacionais (SIMPO I), Fundação Getulio Vargas Escola de Administração de Empresas de São Paulo, Brazil.
  78. Linacre, J.M. (2000). Computer-adaptive testing: a methodology whose time has come, MESA memorandum no. 69. Published in Sunhee Chae, Unson Kang, Eunhwa Jeon and J. M. Linacre. Development of computerised middle school achievement test, Seoul, South Korea, pp. 1-58.
  79. Lister R. (2005). One small step toward a culture of peer review and multi-institutional sharing of educational resources: a multiple choice exam for first semester students. In Proceedings of the 7th Australasian Conference on Computing Education, Vol. 42 (eds A. Young & D. Talhurst), pp. 155– 164. ACM, Newcastle, NSW.
  80. Lister, R., Leaney, J. (2003). First year programming: let all the flowers bloom. In Proceedings of the Fifth Australasian Conference on Computing Education, Vol. 20, pp. 221– 230. ACM, Adelaide, SA.
  81. Lord, F. M. & Novick, M. R. (1968). Statistical theories of mental test scores. Reading MA: Addison-Wesley Publishing Company
  82. Lord, F. M. (1971a). Tailored testing, an approximation of stochastic approximation, Journal of the American Statistical Association, 66, pp. 707–711.
  83. Lord, F. M. (1971b). A theoretical study of the measurement effectiveness of flexilevel tests, Educational and Psychological Measurement, 31, pp. 805–813.
  84. Lord, F. M. (1977). A broad-range tailored test of verbal ability. Applied Psychological Measurement, 1, pp. 95-100.
  85. Lord, F. M. (1980). Applications of item response theory to practical problems. Hillsdale, NJ: Lawrence Erlbaum Associates.
  86. Lord, F.M. (1952). A theory of test scores. Psychometric Monograph, 7. Chicago, IL: Psychometric Society.
  87. Lord, F.M. (1970). Some test theory for tailored testing. In W.H. Holtzman (Ed.), Computer-assisted instruction, testing and guidance. New York: Harper and Row.
  88. Ljubotina, D. Teorija odgovora na zadatke (Item response Theory-IRT). Konceptualizacija i mogućnost primene, Sveučilište u Zagrebu, Filozofski fakultet, Odsek za psihologiji. [www.ffzg.hr/psih/phm/nastava/phm/toz\\_predavanje.ppt](http://www.ffzg.hr/psih/phm/nastava/phm/toz_predavanje.ppt) [20.06.2011.]
  89. Maravić Čisar, S., Čisar, P., Pinter, R. (2009). True/false Questions Analysis Using Computerized Certainty-based Marking Tests. 7th International Symposium on Intelligent Systems and Informatics SISY 2009, Subotica, Serbia. Proceedings CD ROM,

- IEEE Catalog Number: CFP0984C-CDR, ISBN: 978-1-4244-5349-8, Library of Congress: 2009909575
90. Maravić Čisar, S., Radosav, D., Markoski, B., Pinter, R., Čisar, P. (2010). Computer Adaptive Testing of Student Knowledge, *Acta Polytechnica Hungarica*, Vol.7, No.4, ISSN 1785-8860, pp. 139-152.
  91. Maravić Čisar, S., Radosav, D., Markoski, B., Pinter, R., Čisar, P. (2010). New Possibilities for Assessment through the Use of Computer Based Testing, 8th International Symposium on Intelligent Systems and Informatics SISY 2010, Subotica, Serbia. Proceedings CD ROM, IEEE Catalog Number: CFP1084C-CDR, ISBN: 978-1-4244-7395-3, pp. 149-152
  92. Maravić Čisar, S., Radosav, D., Markoski, B., Pinter, R., Čisar, P. (2010). Računarski adaptivni test realizovan u MATLAB-u, Zbornik radova II međunarodnog naučno-stručnog skupa Informacione tehnologije za e-obrazovanje, ITEO 2010, Banja Luka, pp. 157-163, ISBN 978-99955-49-48-0
  93. Maravić Čisar, S., Radosav, D., Markoski, B., Pinter, R., Čisar, P. (2010). Computer Adaptive Testing for Student's Knowledge in C++ Exam, Proceedings of 11th International Symposium on Computational Intelligence and Informatics, pp. 263-266, Budapest, IEEE Catalog Number: CFP1024M-PRT, ISBN: 978-4244-9278-7
  94. Maravić Čisar, S., Radosav, D., Pinter, R., Čisar, P. (2011). Effectiveness of Program Visualization: A Case Study with the Jeliot 3, *International Journal of Computers, Communications & Control*, ISSN 1841-9836, E-ISSN 1841-9844, Vol. 6, No. 4. Accepted for publication.
  95. Marković, S., Dedić, V. (2011). Poređenje klasičnog i adaptivnog računarski podržanog testiranja, *INFOTEH-JAHORINA* Vol. 10, Ref. E-V-12, pp. 804-808.
  96. Marković, V. (2002). Osnove Authorwarea, [www.cet.rs/cetcitaliste/CitalisteTekstovi/429.pdf](http://www.cet.rs/cetcitaliste/CitalisteTekstovi/429.pdf). Pristupljeno 2002 [20.06.2011.]
  97. Marković, V. (2003a). Osnove Authorwarea, [www.cet.rs/cetcitaliste/CitalisteTekstovi/429.pdf](http://www.cet.rs/cetcitaliste/CitalisteTekstovi/429.pdf). Pristupljeno [20.06.2011.]
  98. Marković, V. (2003b). Osnove Authorwarea, [www.cet.rs/cetcitaliste/CitalisteTekstovi/443.pdf](http://www.cet.rs/cetcitaliste/CitalisteTekstovi/443.pdf). Pristupljeno [20.06.2011.]
  99. McBride, J. R. (2001) Technical Perspective, in W. A. Sands, B. K. Waters, & J. R. McBride (Eds.) (2001), *Computerized adaptive testing: From inquiry to operation*, Washington DC: American Psychological Association.
  100. Mellenbergh, G. J. (1996). Measurement precision in test score and item response models. *Psychological Methods*, Vol 1(3), pp. 293-299.
  101. Mendes, A. J. and Marcelino, M. J. (2006). Tools to support initial programming learning. *International Conference on Computer Systems and Technologies - CompSysTech'06*. pp. 1-6. <http://ecet.ecs.ru.acad.bg/cst06/Docs/cp/siv/IV.16.pdf>
  102. Mendoza Díaz et al. (2011). Computer Adaptive Testing and the Networked Model of Curriculum, *Proceedings of the Research in Engineering Education Symposium 2011 – Madrid*, pp. 1-6.
  103. Mills, C. N. (1999). Development and introduction of a computer adaptive Graduate Record Examination general test. In F. Drasgow & J. B. Olson-Bunuchanan (Eds.),

- Innovations in computerized assessment (pp. 117-135). Mahwah, NJ: Lawrence Erlbaum Associates.
104. Mills, C. N., & Stocking, M. L. (1995). Practical issues in large-scale high-stakes computerized adaptive testing. (Research Report 95-23). Princeton, NJ: Educational Testing Service.
  105. Milne, I., Rowe, G. (2002). Difficulties in Learning and Teaching Programming—Views of Students and Tutors. *Education and Information Technologies* 7:1, 55–66, 2002. <http://course.zjnu.cn/kcjx/uploadfile/20081127212811932.pdf>
  106. Mitrovic, A. & Martin, B. (2004). Evaluating Adaptive Problem Selection, *Lecture Notes in Computer Science*, 3137, pp. 185–194.
  107. Naglieri, J. A., Drasgow, F., Schmit, M., Handler, L., Prifitera, A., Margolis, A., & Velasquez, R. (2004). Psychological testing on the internet: New problems, old issues. *American Psychologist*, 59, pp. 150-162.
  108. Olea, J., Abad, F. J., Ponsoda, V., Aguado, D., Díaz, J. (2011). Development, psychometric properties and new validity evidences of the web-based computerized adaptive test of English eCat, *Revista Electrónica de Metodología Aplicada* 2011, Vol. 16, n 1, pp. 50-65. <http://www.iic.uam.es/pdfs/eCatPDF.pdf> [10. 06. 2011.]
  109. Paek, P. (2005). Recent trends in comparability studies (PEM Research Report 05-05). [http://www.pearsonedmeasurement.com/downloads/research/RR\\_05\\_05.pdf](http://www.pearsonedmeasurement.com/downloads/research/RR_05_05.pdf). [10. 12. 2010.]
  110. Parshall, C.G., Spray, J.A., Kalohn, J.C., & Davey, T. (2002). Practical considerations in computer-based testing. New York: Springer-Verlag.
  111. Partchev, I. (2004). A visual guide to item response theory, Friedrich-Schiller-Universität Jena, [www.uni-jena.de/svw/metheval/irt/VisualIRT.pdf](http://www.uni-jena.de/svw/metheval/irt/VisualIRT.pdf)
  112. Pendergast, Mark O. (2006). Teaching Introductory Programming to IS Students: Java Problems and Pitfalls, *Journal of Information Technology Education*, Volume 5, pp. 491-515
  113. Pérez, D. & Alfonseca, E. (2004) Adapting the Automatic Assessment of Free-Text Answers to the Students, *Proceedings for 8th Computer-Assisted Assessment Conference 2004*, Available: [http://www.caaconference.com/pastConferences/2005/proceedings/PerezD\\_AlfonsecaE.pdf](http://www.caaconference.com/pastConferences/2005/proceedings/PerezD_AlfonsecaE.pdf)
  114. Pinter, R. (2008). On the road to Adaptive and Intelligent Webbased Educational Systems, 6th International Symposium on Intelligent Systems and Informatics, pp. 318-320, Subotica, Serbia. IEEE Catalog Number: CFP0884C-CDR, ISBN: 978-1-4244-2407-8, Library of Congress: 2008903275
  115. Pinter, R., Radosav, D. and Maravić Čisar, S. (2010). Interactive Animation in Developing e-Learning Contents, *Proceedings of 33rd International Convention MIPRO 2010, Computers in Education*, Vol. IV, May 24-28, 2010, Opatija, Croatia, ISSN 1847-3938, ISBN 978-953-233-054-0, pp. 251-254
  116. Radosav, D., Karuovic, D. (2009). Knowledge Checking when Applying Multimedia in Web Authorized Systems, 5th Conference of Information and Communication Technology, *Proceeding INFORMATIKA*, pp.41-45, ISSN 1986-5295, Knjiga 7, Mostar.

117. Radosav, D., Marušić, T. (2007). Verification of Knowledge within Interactive Learning with Authoring Systems, *Mostariensia, Časopis za humanističke znanosti, Sveučilište u Mostaru*, broj 26, pp.139-152, ISSN 1023-8638, Mostar.
118. Rasch, G. (1960). Probabilistic models for some intelligence and attainment tests. Copenhagen: Danish Institute for Educational Research
119. Reckase, M.D. (1989). Adaptive testing: The evolution of a good idea. *Educational Measurement Issues and Practice*, 8, pp. 11-15.
120. Revuelta, J. & Ponsoda, V. (1998). A comparison of item exposure control methods in computerized adaptive testing. *Journal of Educational Measurement*, 35(4), pp. 311-328.
121. Rhodes A., Bower A. & Bancroft P. (2004) Managing large class assessment. In *Proceedings of the Sixth Conference on Australasian Computing Education*, Vol. 30 (eds R. Lister & A. Young), pp. 285–289. ACM, Dunedin.)
122. Rudner, L. M. (1998). An On-line, Interactive, Computer Adaptive Testing Tutorial. <http://echo.edres.org:8080/scripts/cat/catdemo.htm> [28. 06. 2011.]
123. Rudner, L. M. (2009). An examination of decision-theory adaptive testing procedures. In D. J. Weiss (Ed.), *Proceedings of the 2009 GMAC Conference on Computerized Adaptive Testing*. [www.psych.umn.edu/psylabs/CATCentral/](http://www.psych.umn.edu/psylabs/CATCentral/) [28. 06. 2011.]
124. Rulison, K. L. & Loken, E. (2009). I've fallen and I can't get up: Can high-ability students recover from early mistakes in CAT? *Applied Psychological Measurement*, 33, pp. 83-101.
125. Savić, M. (2008). Praktikum za definisanje ishoda učenja, Arhitektonski fakultet Univerziteta u Beogradu, Beograd. [http://www.arh.bg.ac.rs/upload/dokumenta/AF\\_Akreditacija/Praktikum%20za%20definisiranje%20ishoda%20ucenja%20MS.pdf](http://www.arh.bg.ac.rs/upload/dokumenta/AF_Akreditacija/Praktikum%20za%20definisiranje%20ishoda%20ucenja%20MS.pdf)
126. Schaeffer, G. A., Steffen, M. Golub-Smith, M. L., Mills, C. N., & Durso, R. (1995). The introduction and comparability of the computer-adaptive GRE General Test (Research Rep. No. 95-20). Princeton NJ: Educational Testing Service.
127. Scheuermann, F. and Guimarães Pereira A. (2008). Towards a research agenda on computer-based assessment, Challenges and needs for European Educational Measurement, European Commission, Joint Research Centre, Institute for the Protection and Security of the Citizen, ISSN 1018-5593, <http://crell.jrc.ec.europa.eu/CBA/EU-Report-CBA.pdf>
128. Segall, D. O., and Moreno, K. E. (1999). Development of the computerized adaptive testing version of the Armed Service Vocational Aptitude Battery. In F. Drasgow & J. B. Olson-Bunuchanan (Eds.), *Innovations in computerized assessment*, pp. 35-65. Mahwah, NJ: Lawrence Erlbaum Associates.
129. Sivasakthi, M., and Rajendran, R. (2011). Learning difficulties of 'object-oriented programming paradigm using Java': students' perspective, *Indian Journal of Science and Technology*, Vol. 4 No. 8. ISSN: 0974- 6846. pp. 983-985
130. Tao, Y.-H., Wu, Y.-L., & Chang, H.-Y. (2008). A Practical Computer Adaptive Testing Model for Small-Scale Scenarios. *Educational Technology & Society*, 11(3), pp. 259–274.
131. Taradi, M., Kukolja Taradi, S., Marijanović, M. V., Radić, K. (2007). Samoprocjena i procjena znanja u e-obrazovanju. <http://www.carnet.hr/referalni/obrazovni/spzit/pismeni/teorija/bloom.html> [20. 05. 2011.]



132. Thissen, D. & Orlando, M. (2001). Item response theory for items scored in two categories. In D. Thissen & Wainer, H. (Eds.), *Test Scoring*, pp. 73-140. Mahwah, NJ: Lawrence Erlbaum Associates, Inc.
133. Thissen, D., & Mislevy, R. J. (2000). Testing algorithms. In H. Wainer, N. Dorans, D. Eignor, R. Flaugher, B. Green, R. Mislevy, L. Steinberg, & D. Thissen (Eds.), *Computerized Adaptive Testing: A Primer* (2nd ed), pp. 101-133. Hillsdale, NJ: Erlbaum
134. Thompson, Nathan A., & Weiss, David A. (2011). A Framework for the Development of Computerized Adaptive Tests. *Practical Assessment, Research & Evaluation*, 16(1). Available online: <http://pareonline.net/getvn.asp?v=16&n=1>.
135. Tian Jian-quan, Miao Dan-min, Zhu Xia, Gong Jing-jing. (2007). An Introduction to the Computerized Adaptive Testing. *US-China Education Review*, ISSN1548-6613, USA, , Volume 4, No.1 (Serial No.26), pp.72-81.
136. Tighe et al. (2011). Use of Computer Adaptive Testing in the Development of Machine Learning Algorithms, *Pain Medicine* 2011; 12: 1450–1452, Wiley Periodicals, Inc. <http://anest.ufl.edu/files/2011/09/Tighe-MLT-Editorial.pdf>
137. Traynor, D. , Gibson, J.P. (2005.) Synthesis and analysis of automatic assessment methods in CS1. *ACM SIGCSE Bulletin* 37, 495–499.
138. Triantafillou, E., Georgiadou, E., Economides, A. A. (2010). The Implementation of a Server-based Computerized Adaptive Testing on Mobile Devices (CAT-MD), *Novel Developments in Web-Based Learning Technologies: Tools for Modern Teaching*. doi: 10.4018/978-1-60566-938-0.ch020
139. Tzanavari, A.; S. & Pastellis, P. (2004). Giving More Adaptation Flexibility to Authors of Adaptive Assessments, *Adaptive Hypermedia and Adaptive Web-Based Systems 2004*, *Lecture Notes in Computer Science*, 3137/2004, pp. 340–343.
140. Van der Linden, W. J. (2010). Constrained adaptive testing with shadow tests. In W. J. van der Linden & C. A. W. Glas (Eds.), *Elements of adaptive testing*, pp. 31-55. New York: Springer.
141. Van der Linden, W. J. and Glas, C. A. W. (2000). *Computerized adaptive testing: Theory and practice*, Boston: Kluwer.
142. Van der Linden, W. J., Pashley, P. J. (2010). Item Selection and Ability Estimation in Adaptive Testing, *Elements of Adaptive Testing*, *Statistics for Social and Behavioral Sciences*, 2010, Part 1, pp. 3-30. doi: 10.1007/978-0-387-85461-8\_1
143. Van der Linden, W.J. & Glas, C.A.W. (2003). Preface. In van der Linden, W.J., Glas, C.A.W (Eds). *Computerised Adaptive Testing: Theory and Practice*. Dordrecht, Boston, London: Kluwer Academic Publishers, VI-XII.
144. Vispoel, W. P., Rocklin, T. R., & Wang, T. (1994). Individual differences and test administration procedures: A comparison of fixed-item, computerized-adaptive, and self-adapted testing. *Applied Measurement in Education*, 53, pp. 53-79.
145. Vispoel, W. P., Wang, T., & Bleiler, T. (1997). Computerized adaptive and fixed-item testing of music listening skill: A comparison of efficiency, precision, and concurrent validity. *Journal of Educational Measurement*, 34, pp. 43–63.

146. Wainer, H. & Mislevy, R. J. (2000) Item Response Theory, Item Calibration, and Proficiency Estimation, in H. Wainer (2000), Computerized Adaptive Testing: A Primer, Lawrence Erlbaum Associates Inc.
147. Wainer, H. & Thissen, D. (1987). Estimating ability with the wrong model. *Journal of Educational Statistics*, 12, pp. 339-368.
148. Wainer, H. (1989). The future of item analysis. *Journal of Educational Measurement*, 26, pp. 191-208.
149. Wainer, H. (2000). *Computerized Adaptive Testing (A Primer)*, 2nd Edition. New Jersey: Lawrence Erlbaum Associates.
150. Wang, F.L., Wong, T.L. (2008). Designing programming exercises with computer assisted instruction. In *Proceedings of the First International Conference on Hybrid Learning and Education (ICHL 2008)* (eds J. Fong, R. Kwan & F.L. Wang), pp. 283–293. Springer, Berlin/Heidelberg. August 13–15, 2008, LNCS 5169.
151. Wang, P., Dai, H., Ding, S. (2010). Computerized Adaptive Testing in Chinese Mainland: A Review. *International Journal of Digital Content Technology and its Applications* Volume 4, Number 6, September 2010.
152. Wang, S., Jiao, H., Young, M. J., Brooks, T. E., & Olson, J. (2008). Comparability of computer-based and paper-and-pencil testing in K-12 assessment: A meta-analysis of testing mode effects. *Educational and Psychological Measurement*, 68, pp. 5-24.
153. Wang, S., Jiao, H., Young, M. J., Brooks, T. E., & Olson, J. (2007). A meta-analysis of testing mode effects in Grade K–12 mathematics tests. *Educational and Psychological Measurement*, 67, pp. 219-238.
154. Wang, T., & Kolen, M. J. (2001). Evaluating comparability in computerized adaptive testing: Issues, criteria and an example. *Journal of Educational Measurement*, 38, pp. 19–49.
155. Ware, J.E., Kosinski, M., Bjorner, J.B., Bayliss, M.S., Batenhorst, A., Dahlöf, C.G.H., Teper, S., Dowson, A. (2003). Applications of computerized adaptive testing (CAT) to the assessment of headache impact. *Quality of Life Research*, Volume 12, Number 8, pp. 935-952, doi: 10.1023/A:1026115230284
156. Way, W. D. (1998). Protecting the integrity of computerized testing item pools. *Educational Measurement: Issues and Practice*, 17, pp. 17-27.
157. Weiss, D. J. (1973). *The stratified adaptive computerized ability test* (Research Report 73-3). Minneapolis: University of Minnesota, Department of Psychology, Psychometric Methods Program, Computerized Adaptive Testing Laboratory.
158. Weiss, D. J., *Computerized Adaptive Testing, Research and Applications*, <http://www.psych.umn.edu/psylabs/catcentral/> [15.06.2011.]
159. Whalley, J.L., Lister, R., Thompson, E., Clear, T., Robbins, P., Kumar, P.K.A.&Prasad, C. (2006). An Australasian study of reading and comprehension skills in novice programmers, using the Bloom and SOLO taxonomies. In *Proceedings of the 8th Australian Conference on Computing Education*, Vol. 52 (eds D. Tolhurst & S. Mann), pp. 243–252. ACM, Hobart, Tas.
160. Wheadon, C. & He, Q. (2006). An Investigation of the Response Time for Maths Items in a Computer Adaptive Test. *Proceedings of the 10th Computer-Assisted Assessment*

- Conference 2006. [http://www.caaconference.com/pastConferences/2006/proceedings/Wheadon\\_C\\_He\\_Q\\_j3](http://www.caaconference.com/pastConferences/2006/proceedings/Wheadon_C_He_Q_j3). [10. 06. 2010.]
161. Wise, S. L., & Kingsbury, G. G. (2000). Practical issues in developing and maintaining a computerized adaptive testing program. *Psicológica*, 21, pp. 135-155.
162. Yong, C. F. & Higgins, C. (2004). Self-assessing with Adaptive Exercises, Proceedings for 8th Computer-Assisted Assessment Conference 2004, Available: <http://www.caaconference.com/pastConferences/2004/proceedings/Yong.pdf>
163. Zhou, J., Gierl, M. J., & Cui, Y. (2007). Computerized attribute-adaptive testing: A new computerized adaptive testing approach incorporating cognitive psychology. In D. J. Weiss (Ed.), Proceedings of the 2007 GMAC Conference on Computerized Adaptive Testing. [www.psych.umn.edu/psylabs/CATCentral/](http://www.psych.umn.edu/psylabs/CATCentral/)
164. Zickar, M. J. and Drasgow, F. (1996). Detecting faking on a personality instrument using appropriateness measurement, *Applied Psychological Measurement*, Vol. 20, pp. 71–87.
165. Zickar, M.J., Overton, R., Taylor, L., & Harms, H. (1999). Development of an adaptive test for computer programmers. In Drasgow, F. & Olson-Buchanan, J. B. (Eds), *Innovations in computerized assessment*, pp. 7-33. Lawrence Erlbaum: Mahwah, NJ.

Korišćeni web sajtovi:

166. [www.assess.com/xcart/product.php?productid=270&cat=0&page=1](http://www.assess.com/xcart/product.php?productid=270&cat=0&page=1). Assessment Systems Corporation XCALIBRE - Marginal Maximum-Likelihood Estimation; Assessment Systems Corporation. [16. 04. 2011.]
167. [www.cpa-exam.org](http://www.cpa-exam.org), American Institute of Certified Public Accountants (AICPA), [16. 04. 2011.]
168. [www.ilrs.mdx.ac.uk/lang/ells/cate.htm](http://www.ilrs.mdx.ac.uk/lang/ells/cate.htm), English Language and Learning Support of the Information and Learning Resource Services at Middlesex University in the United Kingdom [16. 04. 2011.]
169. [www.collegedegreeguide.com/articles-fr/gre.htm](http://www.collegedegreeguide.com/articles-fr/gre.htm). CollegeDegreeGuide [05. 04. 2011.]
170. [www.edutools.info](http://www.edutools.info). EduTools, the Western Cooperative for Educational Telecommunications. [05. 04. 2011.]
171. [www.ets.org/toefl](http://www.ets.org/toefl). Educational Testing Service. [05. 04. 2011.]
172. [www.official-asvab.com](http://www.official-asvab.com). The Armed Services Vocational Aptitude Battery (ASVAB) [15. 06. 2011.]
173. [www.schuhfried.co.at](http://www.schuhfried.co.at). Schuhfried company, computer-based psychological assessment. [15. 06. 2011.]
174. [www.bopsycho.rwth-aachen.de](http://www.bopsycho.rwth-aachen.de). Institut für Psychologie der RWTH Aachen [28. 06. 2011.]
175. [www.carla.umn.edu/assessment/CAT.html](http://www.carla.umn.edu/assessment/CAT.html). Center for Advanced Research on Language Acquisition (CARLA) [28. 06. 2011.]
176. [www.ec.europa.eu/growthandjobs/](http://www.ec.europa.eu/growthandjobs/). European Commission the Lisbon strategy [28. 06. 2011.]
177. [www.dodaj.rs/f/2k/L6/3yjaTYg7/8-psihologija-nastavnika.ppt](http://www.dodaj.rs/f/2k/L6/3yjaTYg7/8-psihologija-nastavnika.ppt). Smiljanić, M. Psihologija. [05. 04. 2011.]

178. <http://www.psychometrics.cam.ac.uk>. The Psychometrics Centre, University of Cambridge [12.05.2011]
179. [www.schuhfried.com/vienna-test-system-vts](http://www.schuhfried.com/vienna-test-system-vts). Vienna Test System [10. 06. 2011.]
180. [www.j-cat.org/en/](http://www.j-cat.org/en/). J-CAT Japanese Language Test [10. 06. 2011.]
181. [www.mathworks.com](http://www.mathworks.com). MATLAB [10. 06. 2011.]
182. [www.mathworks.com/matlabcentral/GREpattern](http://www.mathworks.com/matlabcentral/GREpattern). GRE adaptivni test [10. 06. 2011.]
183. [www.microsoft.com/learning/en/us/certification/exam.aspx](http://www.microsoft.com/learning/en/us/certification/exam.aspx). Microsoft Certification Exams. [10. 06. 2011.]
184. [www.ascp.org/bor/certification/procedures/scoring.asp](http://www.ascp.org/bor/certification/procedures/scoring.asp). American Society for Clinical Pathology [16. 04. 2011.]
185. <http://www.assess.com>. Assessment Systems Corporation [16. 04. 2011.]
186. [www.4roi.com](http://www.4roi.com). 4ROI organization for the development and administration of internet-based assessments and accredited certification tools. [10. 06. 2011.]
187. <http://www.fasttestweb.com>. *FastTest Web* system for online testing and assessment. [10. 06. 2011.]
188. [http://www.assess.com/xcart/skin1/downloads/FastTEST2\\_Manual.pdf](http://www.assess.com/xcart/skin1/downloads/FastTEST2_Manual.pdf). Weiss, D.J. (2010). *User's Manual for FastTEST 2.0*. St. Paul MN: Assessment Systems Corporation. [06. 05. 2011.]