

Izrada aplikacije za Android operativni sustav

Javor, Božidar

Undergraduate thesis / Završni rad

2018

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:200:887960>

Rights / Prava: [In copyright / Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-05-20**

Repository / Repozitorij:

[Faculty of Electrical Engineering, Computer Science
and Information Technology Osijek](#)



SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I
INFORMACIJSKIH TEHNOLOGIJA OSIJEK**

Sveučilišni studij

IZRADA APLIKACIJE ZA ANDROID OPERATIVNI SUSTAV

ZAVRŠNI RAD

Božidar Javor

Osijek, 2018.

SADRŽAJ

1.	UVOD	1
1.1	Zadatak	2
2.	ANDROID	2
2.1	Android verzije	4
2.2	Android uređaji.....	5
3.	IZRADA ANDROID APLIKACIJE.....	7
3.1	Tehnologije izrade	9
3.1.1	Android studio.....	9
3.1.2	Android SDK.....	12
3.2	Algoritam sivog predviđanja.....	12
3.3	Izrada Android aplikacije	16
3.4	Slike iz funkcionalne aplikacije u nekoliko koraka.....	20
4.	ZAKLJUČAK.....	22
5.	SAŽETAK.....	23
6.	ABSTRACT	24
	LITERATURA.....	25
	ŽIVOTOPIS.....	26
	PRILOG 1:IZVORNI DIO APLIKACIJE.....	27

1. UVOD

Ovaj završni rad predstavlja opis sivih sustava, odnosno primjer predviđanja na temelju sivih sustava i izradu android aplikacije koja prikazuje sivo predviđanje. Sivo predviđanje je relativno nova metoda poznata pod nazivom (eng. *grey forecasting*) koja se temelji na teoriji sivih sustava (eng. *grey system theory*). Teorija sivih sustava vrlo je dugo poznata, od 1982. godine, te joj od tada raste popularnost a time i primjena. Danas se koristi u rješavanju raznih problema u različitim područjima.

Sivi sustavi su sustavi za koje nisu poznate sve informacije, odnosno sivi sustav se dijeli na sivi prostor stanja što je zapravo nepoznati dio sustava, te bijeli prostor stanja što čini poznati dio sustava. Cilj je kao što je ranije navedeno iz poznatog dijela informacija saznati ne poznati dio informacija, odnosno iz bijelog dijela sustava saznati sivi dio sustava. Sve je potkrijepljeno matematičkim aparatom.

Kako bi se izradila Android aplikacija potrebno je proći kroz pojedine teorijske dijelove od povijesti Androida i pojašnjenja pojedinih verzija, pa do Android uređaja i primjene. U radu su predstavljeni alati i tehnologije za izradu android aplikacije. Za izradu Android aplikacije potrebne su tehnologije koje su opće poznate svim programerima, a to je program „Android studio“ s svojim dodacima koji su neophodni prilikom izrade Android aplikacije.

Android prvenstveno predstavlja operativni sustav koji je namijenjen za mobilne uređaje, s naznakom na približavanje računalnim funkcionalnostima na mobilne uređaje. Ta činjenica proizlazi iz potrebe da se operacije koje se obavljaju na računalima približe korisnicima na mobilnim uređajima.

Rad je predstavljen kroz teorijski dio pod nazivom „Android“ koji obuhvaća definiciju Androida te verzije Android sustava ali i uređaje na kojima je osposobljen operativni sustav Android. Nastavak rada predstavljen je kroz izradu Android aplikacije koja započinje tehnologijama prilikom izrade, što obuhvaća programsku podršku prilikom izrade te se nastavlja s opisivanjem algoritma za sivo predviđanje i samom izradom.

1.1 Zadatak

Zadatak završnoga rada je izraditi Android aplikaciju za sivo predviđanje na osnovu modela sivih sustava. Prvenstveno je potrebno opisati Android operativni sustav te načine i tehnologije potrebne za izradu Android aplikacije. Po tome je potrebno na temelju algoritma za sivo predviđanje izraditi Android aplikaciju za sivo predviđanje, odnosno predviđanje na osnovu modela sivih sustava.

2. ANDROID

Android je operativni sustav kojeg koriste mobilni uređaji. Nastao je prvenstveno iz razloga da se funkcionalnost osobnog računala što je moguće brže približi mobilnim uređajima, odnosno sve što radimo na računalu, prilagodimo i na mobilnim aplikacijama.

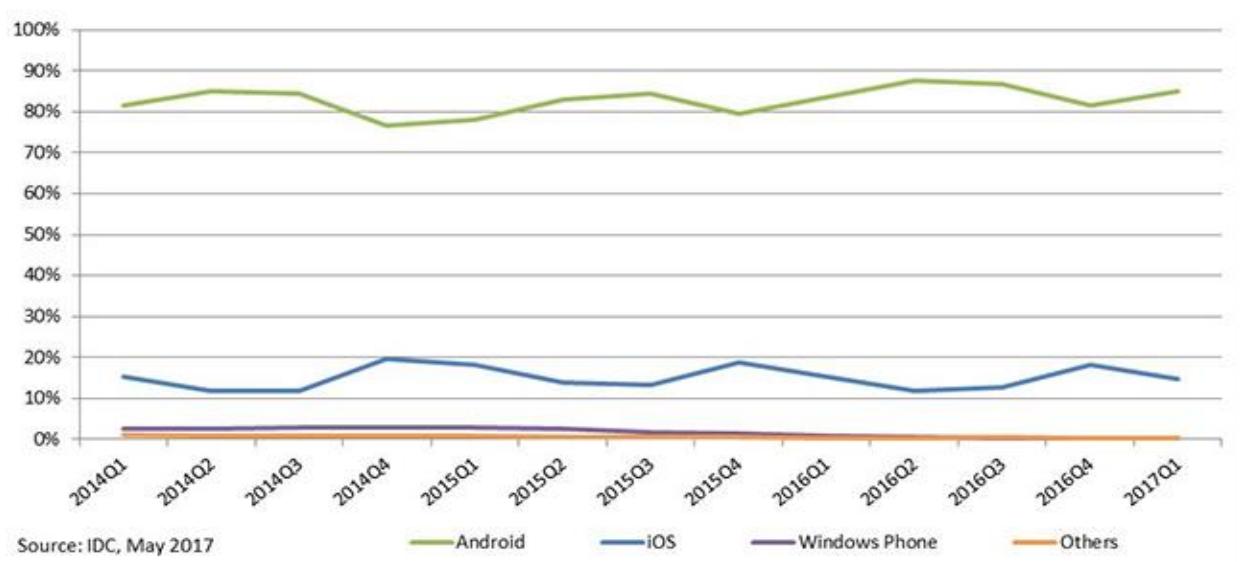


Slika 1. Prikaz Android Logo-a

Karakteristike Google Androida prema OHA(*eng. Open Handset Alliance*) su:

- otvorenost - programerima aplikacija omogućena je sloboda u smislu razvoja aplikacije, a proizvođačima je omogućeno besplatno korištenje.
- ravnopravnost aplikacija – nema razlike između aplikacije koju korisnik instalira i osnovne jezgrene aplikacije. Aplikaciji je omogućen pristup sredstvima i to omogućuje izbor korisniku da prilagodi uređaj svojim potrebama.
- mogućnost da se aplikacija pokrene i izvede, a da pri tome korisnik ne mora brinuti o tome je li druga aplikacija pokrenuta prije nego što je prethodna ugašena,
- omogućeno je brže razvijanje aplikacije pomoću baza programskih biblioteka i drugih alata za izradu,
- omogućen je bolji prikaz slike i zvuka te podrška za audio i video formate.

Prema mnogima, Androidi su mnogo primamljiviji zbog svoje jednostavnosti, ali i pristupačnosti, jer je u većini slučajeva Android uređaj povoljniji od npr. Apple uređaja. Slika 2 predstavlja omjer prodanih uređaja po operativnim sustavima. Iz slike je jasno vidljivo da na svim tržištima većinom prednjače Android uređaji, a Europa je najjača u tome s preko 70% [1].



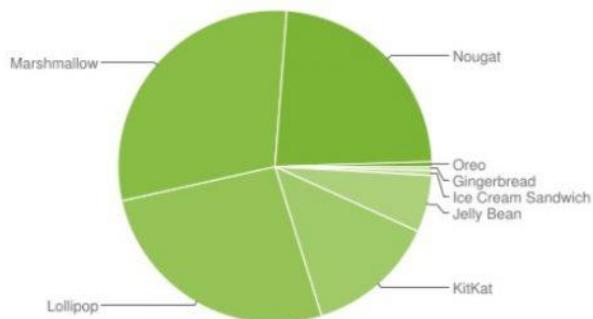
Slika 2. Prikaz konkurentnosti Android uređaja na svjetskom tržištu [6].

2.1 Android verzije

Android verzije poznate su prema tome što izlaze prema nazivima poslastica i slatkiša, ali i prema brojevima.

Posljednja verzija androida 8.0/8.1 Oreo predstavljena je 21.kolovoza.2017 godine. Glavna karakteristika Oreo Android verzije je „PROJECT TREBLE“, sustav koji omogućuje olakšanu i ubrzenu isporuku nadogradnje Android sustava. Neke karakteristike koje ga definiraju su: novi sustav brzih postavki, jednostavno korištenje ikona, duplo brže dizanje sustava te preuzimanje fontova i veći raspon boja za aplikacije [2]. Slika 3 predstavlja trenutni poredak u postotku (%) korištenje Android operativnog sustava prema verzijama.

Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	0.4%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.5%
4.1.x	Jelly Bean	16	2.0%
4.2.x	Jelly Bean	17	3.0%
4.3		18	0.9%
4.4	KitKat	19	13.4%
5.0	Lollipop	21	6.1%
5.1	Lollipop	22	20.2%
6.0		23	29.7%
7.0	Nougat	24	19.3%
7.1	Nougat	25	4.0%
8.0		26	0.5%



Slika 3. Prikaz Android operativnih sustava prema verzijama i trenutnom korištenju [7].

2.2 Android uređaji

Kod android uređaja, prvenstveno treba naglasiti da u to spadaju pametni telefoni koji čine jezgru kako na tržištu tako i na samoj razvojnoj platformi. No, u Android uređaje trenutno možemo svrstati :

- pametne telefone
- tablete
- hibride (tablet/laptop)
- laptop
- internet TV
- smart Watch (pametni satovi).



Slika 4. Prikaz Samsung Chromebook-a, laptop s Android operativnim sustavom [8].

Ovo su kategorije prema kojima Android odnosno Google izdaje svoje operativne sustave te iza čega stoji vrhunska podrška. Treba naglasiti da kada se radi o laptopima Google ima svoju verziju pod nazivom „Chromebook“ koja je rađena od različitih proizvođača prema raznim modelima.

Ovo nije prvi pokušaj da Google ujedinjuje razne proizvođače. Postoji stari projekt „Nexus“ koji godinama izdaje nove Android uređaje s tzv. „Stock Androidom“ odnosno čistim Android operativnim sustavom. Taj sustav nema sučelje zadalog proizvođača, iako svake godine kada izlazi proizvođač bude drugi. Google dodjeljuje zadatom proizvođaču mogućnost izrade mobilnog uređaja koji će biti novi Nexus. Tako do sada obitelj Nexus čine:

- Samsung
- Htc
- Lg
- Motorola
- Huawei
- Asus.

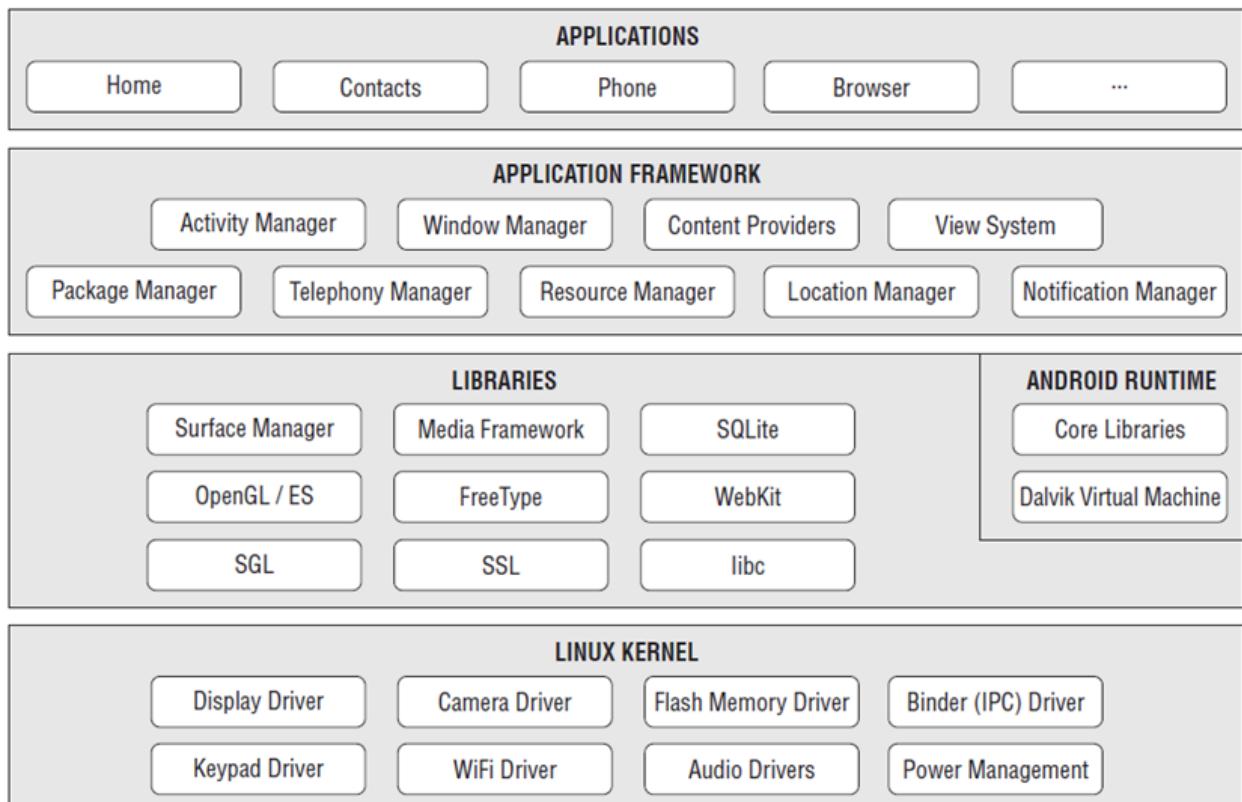


Slika 5. Android Google uređaj Nexus [9].

3. IZRADA ANDROID APLIKACIJE

Android operacijski sustav sastoji se od nekoliko dijelova i slojeva. Svaki pojedini sloj ima svoje karakteristike i ulogu, ali slojevi nisu jasno odvojeni već se najčešće preklapaju jedni s drugima. Android operacijski sustav je baziran na Linux Kernel. Linux je operacijski sustav otvorenog koda, pa aplikacije pomoću posrednika (eng. *middleware*) mogu komunicirati i pokretati druge aplikacije kao što je slanje SMS poruka , e-pošte, slikanja i primanja poziva.

Za sustav temeljen na Linuxu nije potrebno se potrebno brinuti o hardverskim karakteristikama pojedinih uređaja, pa se time omogućuje implementacija Androida na relativno velik raspon uređaja. Najveći broj aplikacija se piše u programskom jeziku Javi uz pomoć android SDK alata.



Slika 6. Prikaz Android arhitekture (Linux Kernel) [3].

Postupkom pisanja aplikacija u C++ programskom jeziku uz korištenje Android SDK alata omogućeno je jednostavnije iskoriščavanje sredstava samog uređaja i korištenje biblioteka

te je povećana brzina aplikacija i do 10 puta. Mana ovakvog pristupa je mnogo složenije pisanje koda.

Ukratko su nabrojane biblioteke koje se nalaze iznad jezgre i služe za najvažnije funkcije:

- Surface Manager – omogućuje nadziranje za grafičko sučelje
- Media Framework – omogućuje korištenje audio i video formata
- SQLite – koristi se za rad s bazama podataka
- OpenGL | ES – omogućuje rad s 3D prikazom
- FreeType – omogućuje rad s fontovima
- WebKit – koristi se kod web preglednika
- SGL – prisutna kod većine aplikacija
- SSL (Secure Sockets Layer) – osigurava komunikaciju na internetu
- libc – omogućuje rad u linuxu

Java čini glavni programski jezik prilikom izrade. Sun Micro systems tvrtka i njeni inženjeri objavili su krajem 1995. godine programski jezik pod nazivom Java. Osnovna prednost Java u odnosu na druge programske jezike je u tome što nije potrebna prilagodba za operacijske sustave koji podržavaju JVM (Java virtual machine) dok se obični programi pisani npr. u C-u moraju prilagoditi operacijskim sustavima u kojem se izvršavaju [3].

Java programski jezik zamišljen je na način da se ne prave gotovo nikakve preinake prilikom korištenja aplikacija na operacijskim sustavima kao što su Windows, Linux itd.

Ideja Java kao programskog jezika zamišljena je na način da bude pristupačna zajednici, samim time je zajednica pridonijela velikom broju alata koji omogućuju lakše korištenje Java. Alati nužni za razvoj aplikacija dostupni su putem Java development kit-a JDK koji je besplatan. Najbitnije u razvoju aplikacije je testiranje aplikacije tijekom izrađivanja, a za programere je najvažniji tzv.unit test [3,4].

3.1 Tehnologije izrade

Prilikom izrade završnog rada potrebno je naglasiti korištenje programa „Android studio“ koji služi kao glavni program i glavno okruženje tijekom izrade Android aplikacije. Prilikom prvoga korištenja potrebno je učitati sve bitne dodatke kako bi se omogućio rad putem virtualne mašine koja će simulirati događanja tijekom same izrade.

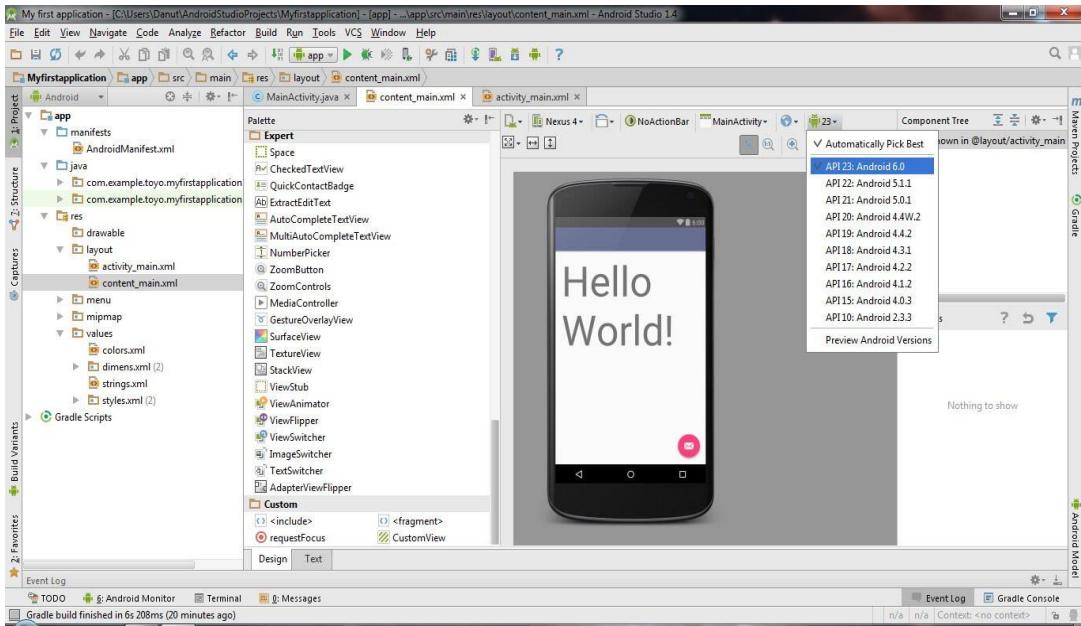
3.1.1 Android studio

Android studio predstavlja službeni i najzastupljeniji alat za razvijanje android aplikacija. Dostupan je za Windows, macOS i Linux operacijske sustave. Prva verzija Android studia 1.0 krenula je s radom u prosincu 2014 godine, a trenutna verzija 3.0 je objavljena u listopadu 2017 godine.



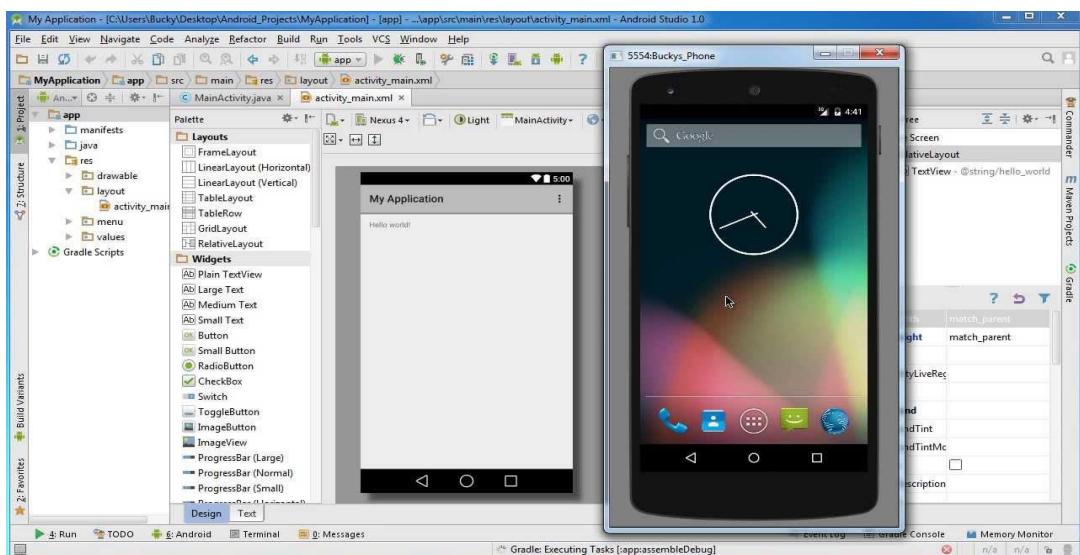
Slika 7. Android studio Logo.

Slika 8. prikazuje izgled rada u Android studiju. Vrlo je teško pričati samo o Android studiju, a ne spomenuti ostale programe i nadogradnje kao i razvojne alate. Prilikom rada odnosno kreacije aplikacije stvara se tzv. Android paket koji obuhvaća kompajlirani java kod sa svim skupovima podataka te datotekama resursa koje su potrebne za razvoj.



Slika 8. Prikaz Android studio okruženja.

Android studio je poprilično snažan alat za sve vrste izrade aplikacija, ali ponajprije za Android upravo zbog velike podrške prilikom korištenja Java. Slika 9. prikazuje pokrenuti Android projekt u Android studiu. Kada se započne nove projekt u Android studiu ponuđen je i odabir konkretnog uređaja odnosno u ponudi s ostalim razvojnim alatima dolaze većinom Nexus uređaji kao okosnica tome što se odmah dobije željena rezolucija prilikom čega može započeti razvoj.



Slika 9. Prikaz započetog projekta u Android studio okruženju s pokrenutom konzolom

Mnogi smatraju vrlo složenim izradu Android aplikacija upravo zbog toga što se koristi Java. Početkom izrade dolazi do početnog prozora koji je predstavljen kao na mobilnom uređaju, odnosno u gornjem kutu prozorčića može se primjetiti da je ikona baterije ili pojedinih drugih svojstava, ali je omogućeno i samo konzolno upravljanje, odnosno pojedine funkcije kao na pravom uređaju što je prikazano na slici 10.



Slika 10. Android konzola tijekom izrade, SDK.

Iz navedenoga se vidi da je konzola u potpunosti virtualno omogućen prikaz Android operativnog sustava, što znači da će kao i na mobilnom uređaju, tijekom izrade biti omogućena aplikacija, te simulirati sva događanja koja se događaju i na Android uređajima. Znači svaka promjena na konzoli prikazat će stvarnu promjenu kao na uređaju.

Svaka aplikacija je svojevrsni unikat jer svaka nova aplikacija koja se izrađuje imat će svoj izgled i svoj dizajn, upravo je to ono što aplikacije čini privlačnima i posebnima. Mnogo je dodatnih alata u Android studiju razvojnog okruženju koje svojevrsno pomažu tijekom izrade, pa tako postoji i niz dodataka(*eng plugins*) za samo kreiranje i olakšano kreiranje aplikacija.

Prvobitni dio koda tzv. „import“ dio koda, učitava zadane biblioteke koje će omogućiti razvoj aplikacije i time omogućiti pristup potrebnim parametrima koji su potrebni za programiranje aplikacije. Kako bi sve bilo omogućeno, odnosno kako bi konzola bila funkcionalna potrebno je ono najvažnije a to je Android SDK (*eng. Software Development Kit*).

3.1.2 Android SDK

Pilikom izrade android aplikacija u Javi najvažniji alat za programiranje je Android SDK. Android SDK sadrži mnoštvo alata koji omogućavaju lakše uklanjanje pogrešaka, simulator za testiranje aplikacije te android aplikacije koje služe kao ogledni primjer. Android SDK može raditi na svim operacijskim sustavima. Za izrađivanje Android aplikacija najviše se upotrebljava program Android studio. Osim Android studija pisanje koda aplikacije za Android moguće je izvršiti u bilo kojem programu za uređivanje teksta.

3.2 Algoritam sivog predviđanja

Ova android aplikacija mora služiti u svrhu predviđanja stanja sustava sivim modelom, što znači da je potrebno integrirati algoritam te prema tome izraditi adekvatno sučelje koje će moći na temelju prethodnih podataka ali i trenutnih podataka previdjeti nadolazeće stanje. Ovo predviđanje vrlo je zastupljeno na temelju pitanja prilikom održavanja tehničkih sustava.

Postoje razni postupci i metode koje se mogu primijeniti, ali svaka ta metoda ima određene prednosti i nedostatke.

Sivo predviđanje je relativno nova metoda poznata pod nazivom (eng. *grey forecasting*) koja se temelji na teoriji sivih sustava (eng. *grey system theory*). Teorija svih sustava vrlo je dugo poznata, od 1982. godine, te joj od tada raste popularnost a time i primjena. Danas se koristi u rješavanju raznih problema u različitim područjima.

Sivi sustavi su sustavi za koje nisu poznate sve informacije, odnosno sivi sustav se dijeli na sivi prostor stanja što je zapravo nepoznati dio sustava, te bijeli prostor stanja što čini poznati dio sustava. Cilj je kao što je ranije navedeno iz poznatog dijela informacija sazнати ne poznati dio informacija, odnosno iz bijelog dijela sustava sazнати sivi dio sustava. Sve je potkrijepljeno matematičkim aparatom.

Kako bi došli do sivog predviđanja potrebno je koristiti model sivog sustava koji se predstavlja diferencijalnom jednadžbom prvoga reda s jednom nepoznanicom. Cilj je kroz algoritam odnosno matematički model načiniti aplikaciju, preslikavanjem matematičkog dijela u programski kod, kojom bi se vrsilo sivo predviđanje za pojedine parametre unutar nekog sustava.

Kako bi sve bilo moguće potreban je algoritam za sivo predviđanje. Kada se radi o praksi ponekada je teško doći do podataka koji su prikupljeni u jednakim intervalima, tj. problem nastaje prilikom prikupljanja podataka koji u velikom broju slučaja nisu jednaki.

Kada se radi o izvršavanju algoritma sivog predviđanja nad podacima s nejednakim intervalima prikupljanja podataka, stvara se potreba za prilagođavanjem ulaznih parametara odnosno podataka kako bi se sveli na jednakе vremenske intervale.

Zadatak je sada kreirati aplikaciju na temelju algoritma za sivo predviđanje te prikazati rezultate koji će pokazati točnost zadanog algoritma kojega je potrebno prikazati kroz Java programski jezik kako bi ga bilo moguće implementirati unutar aplikacije za Android operativni sustav. Za primjer su uzeti podaci o količini otpada u Slavonskom Brodu te prema tim podacima i formulama za GM(1,1) model napravljeni su sljedeći proračuni:

Primjer. Sivo predviđanje količine otpada u Slavonskom Brodu.

U Slavonskom Brodu 2013-te godine količina otpada iznosila je 26406 tona, prema tome količina otpada po stanovniku iznosila je 1.22 kg/st./dan.

$$\text{Količina otpada po stanovniku} = 26406 \times 1000 / (59141 \times 365) = 1.22 \text{ kg/st./dan}$$

Pretpostavlja se da količina otpada koji proizvede jedan stanovnik Slavonskog Broda iznosi 0.85 – 0.9 kg/st./dan.

Tablica 1 Količina otpada na području Grada Savanskog Broda „Vijuš Jug“ od 2009. do 2013. godine.

Godina	2009.	2010.	2011.	2012.	2013.
Komunalni otpad (tona)	35537.8	29339.1	27723.2	26024.2	26406.8

U navedenom primjeru provedeno je predviđanje za podatke od 2009. do 2013. godine kako bi se predvidjela količina otpada u 2014. godini.

1. Izvorni niz: $x^{(0)} = (35537.8, 29339.1, 27723.2, 26024.2, 26406.8)$

2. AGO niz: $x^{(1)} = (35537.8, 64876.9, 92600.1, 118624.3)$

3. Srednji niz: $z^{(1)} = (50207.35, 78738.5, 105612.2)$

4. Matrica Y: $Y = \begin{bmatrix} 29339.1 \\ 27723.2 \\ 26024.2 \end{bmatrix}$

5. Matrica B: $\begin{bmatrix} -50207.35 & 1 \\ -78738.5 & 1 \\ -105612.2 & 1 \end{bmatrix}$

6. Koeficijenti a, b pomoću jednadžbe $\hat{a} = (B^T B)^{-1} B^T Y$ gdje je $\hat{a} = \begin{bmatrix} a \\ b \end{bmatrix}$.

$$B^T = \begin{bmatrix} -50207.35 & -78738.5 & -105612.2 \\ 1 & 1 & 1 \end{bmatrix}$$

$$B^T B = \begin{bmatrix} 19874466165.11 & -234558.05 \\ -234558.05 & 3 \end{bmatrix}$$

$$(B^T B)^{-1} = \begin{bmatrix} 0 & 0 \\ 0 & 4.31 \end{bmatrix}$$

$$\hat{a} = \begin{bmatrix} 0.06 \\ 32370.84 \end{bmatrix}$$

Dakle, $a = 0.06, b = 32370.84$.

7. Jednadžba predviđanja.

$$\hat{x}^{(0)}(k+1) = (1 - e^a) \left(x^{(0)}(1) - \frac{b}{a} \right) e^{-ak}$$

$$\hat{x}^{(0)}(k+1) = (1 - e^{0.06}) \left(35537.8 - \frac{32370.84}{0.06} \right) e^{-0.06k}, k = 1, 2, 3, 4$$

Tablica 2 Izračunata predviđena vrijednost.

Godina	2009.	2010.	2011.	2012.	2013.	2014.
Prava vrijednost	35537.8	29339.1	27723.2	26024.2	26406.8	
Predviđena vrijednost	35537.8	29349.3	27640.1	26030.48	24514.6	23086.97

8. Računanje greške:

Koristeći formulu za grešku GM(1,1) modela $e(k) = \left| \frac{x^{(0)}(k) - \hat{x}^{(0)}(k)}{x^{(0)}(k)} \right| \times 100\%$ dobivamo greške:

Tablica 3 Izračunata pogreška predviđanja.

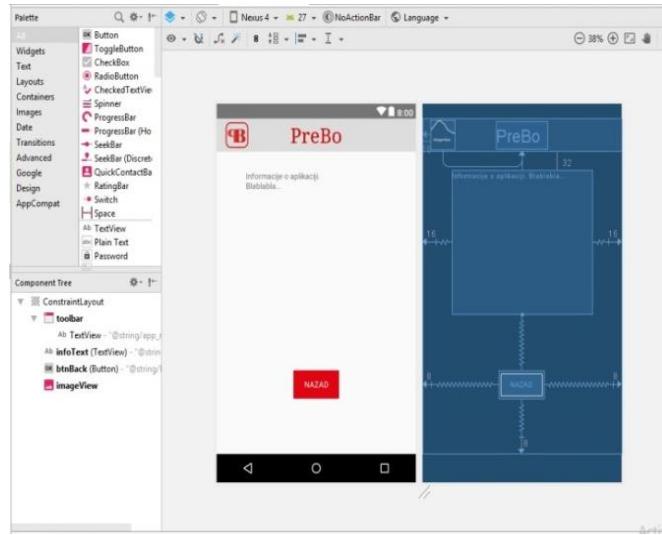
Godina	2009.	2010.	2011.	2012.	2013.
Pogreška	0%	0.188%	0.3%	0.02%	7.17%

9. Izračunata prosječna pogreška: $\frac{0\%+0.188\%+0.3\%+0.02\%+7.17\%}{5} = 1.54\%$

3.3 Izrada Android aplikacije

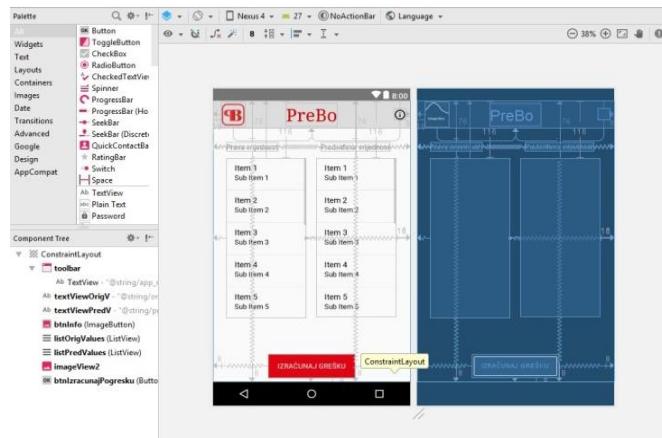
Prikaz izrade aplikacije

Slika 11 prikazuje proces izrade front-end dijela aplikacije koji se odnosi na informacije o aplikaciji i njezinoj upotrebi.



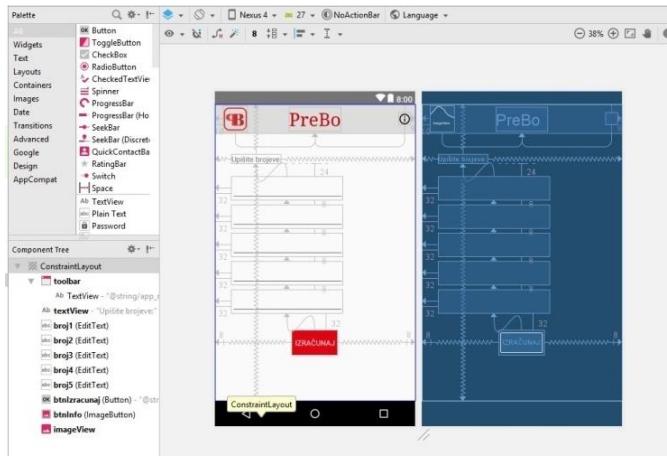
Slika 11. Prikaz izrade info dijela aplikacije.

Slika 12 prikazuje proces izrade front-end dijela aplikacije koji se odnosi na izračun aplikacije.



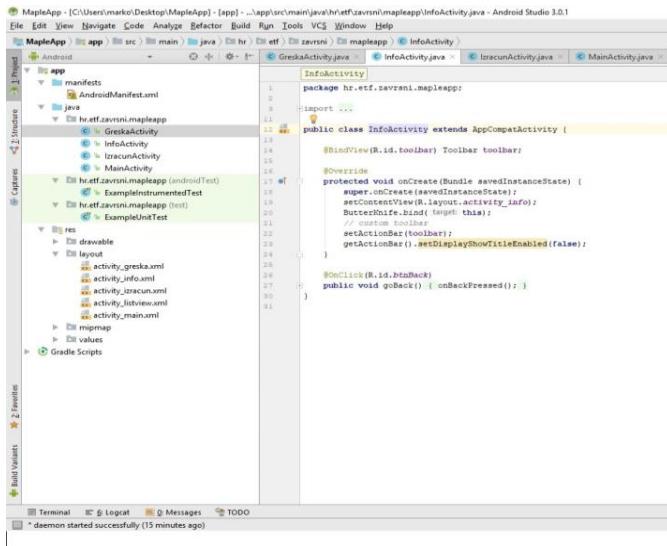
Slika 12. Prikaz izračuna u front-end dijelu aplikacije

Slika 13 prikazuje front-end dio unošenja 5 brojeva kako bi se mogao izvršiti daljnji postupak računanja.



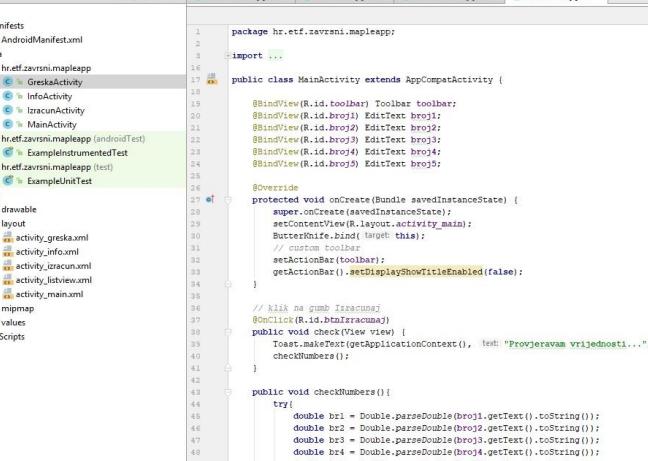
Slika 13. Prikaz unošenja brojeva u front-end dijelu aplikacije

Slika 14 prikazuje Back-end dio procesa izrade aplikacije prilikom koje sam kodirao info dio koji se odnosi na opis aplikacije i njehe svrhe.



Slika 14. Back-end dio info dijela aplikacije

Slika 15 prikazuje back-end dio prilikom kojeg sam kodirao početni zaslon koji se prikazuje otvaranjem aplikacije te omogućuje unos brojeva kako bi se izračun mogao izvršiti.



The screenshot shows the Android Studio interface with the project 'MapleApp' open. The left sidebar displays the project structure, including packages like 'app' and 'hr.efz.zavrsni.mapleapp', and files such as 'AndroidManifest.xml', 'MainActivity.java', and various XML layout files. The main editor window shows the Java code for 'MainActivity.java'. The code implements an AppCompatActivity and contains methods for initializing a toolbar and displaying a toast message. It also includes a placeholder for a checkNumbers() method.

```
1 package hr.efz.zavrsni.mapleapp;
2
3 import ...
4
5 public class MainActivity extends AppCompatActivity {
6
7     @BindView(R.id.toolbar) Toolbar toolbar;
8     @BindView(R.id.broj1) EditText broj1;
9     @BindView(R.id.broj2) EditText broj2;
10    @BindView(R.id.broj3) EditText broj3;
11    @BindView(R.id.broj4) EditText broj4;
12    @BindView(R.id.broj5) EditText broj5;
13
14    @Override
15    protected void onCreate(Bundle savedInstanceState) {
16        super.onCreate(savedInstanceState);
17        setContentView(R.layout.activity_main);
18        ButterKnife.bind(this);
19        setSupportActionBar(toolbar);
20        getSupportActionBar().setDisplayHomeAsUpEnabled(false);
21
22        // Klik na tombon Provjerava vrijednost
23        @OnClick(R.id.buttonProvjeri)
24        public void checkProvjeri(View view) {
25            Toast.makeText(getApplicationContext(), text: "Provjeravam vrijednost...", Toast.LENGTH_SHORT);
26            checkNumbers();
27        }
28
29        public void checkNumbers() {
30            try {
31                double br1 = Double.parseDouble(broj1.getText().toString());
32                double br2 = Double.parseDouble(broj2.getText().toString());
33                double br3 = Double.parseDouble(broj3.getText().toString());
34                double br4 = Double.parseDouble(broj4.getText().toString());
35                double br5 = Double.parseDouble(broj5.getText().toString());
36                double[] brojevi = {br1, br2, br3, br4, br5};
37
38                Intent intent = new Intent(getApplicationContext(), MainActivity.this, IrcunActivity.class);
39                intent.putExtra("brojevi", brojevi);
40                startActivity(intent);
41            } catch (Exception e) {
42                e.printStackTrace();
43            }
44        }
45    }
46
47
48
49
50
51
52
53
54
55
56
57
58
59
59
60
61
62
63
64
65
66
67
68
69
69
70
71
72
73
74
75
76
77
78
79
79
80
81
82
83
84
85
86
87
88
89
89
90
91
92
93
94
95
96
97
98
99
99
100
101
102
103
104
105
106
107
108
109
109
110
111
112
113
114
115
116
117
118
119
119
120
121
122
123
124
125
126
127
128
129
129
130
131
132
133
134
135
136
137
137
138
139
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
159
160
161
162
163
164
165
166
167
168
169
169
170
171
172
173
174
175
176
177
178
179
179
180
181
182
183
184
185
186
187
187
188
189
189
190
191
192
193
194
195
196
197
197
198
199
199
200
201
202
203
204
205
206
207
207
208
209
209
210
211
212
213
214
215
216
216
217
218
218
219
219
220
221
222
223
224
225
225
226
227
227
228
228
229
229
230
231
232
233
233
234
235
235
236
236
237
237
238
238
239
239
240
241
241
242
242
243
243
244
244
245
245
246
246
247
247
248
248
249
249
250
251
251
252
252
253
253
254
254
255
255
256
256
257
257
258
258
259
259
260
260
261
261
262
262
263
263
264
264
265
265
266
266
267
267
268
268
269
269
270
270
271
271
272
272
273
273
274
274
275
275
276
276
277
277
278
278
279
279
280
280
281
281
282
282
283
283
284
284
285
285
286
286
287
287
288
288
289
289
290
290
291
291
292
292
293
293
294
294
295
295
296
296
297
297
298
298
299
299
300
300
301
301
302
302
303
303
304
304
305
305
306
306
307
307
308
308
309
309
310
310
311
311
312
312
313
313
314
314
315
315
316
316
317
317
318
318
319
319
320
320
321
321
322
322
323
323
324
324
325
325
326
326
327
327
328
328
329
329
330
330
331
331
332
332
333
333
334
334
335
335
336
336
337
337
338
338
339
339
340
340
341
341
342
342
343
343
344
344
345
345
346
346
347
347
348
348
349
349
350
350
351
351
352
352
353
353
354
354
355
355
356
356
357
357
358
358
359
359
360
360
361
361
362
362
363
363
364
364
365
365
366
366
367
367
368
368
369
369
370
370
371
371
372
372
373
373
374
374
375
375
376
376
377
377
378
378
379
379
380
380
381
381
382
382
383
383
384
384
385
385
386
386
387
387
388
388
389
389
390
390
391
391
392
392
393
393
394
394
395
395
396
396
397
397
398
398
399
399
400
400
401
401
402
402
403
403
404
404
405
405
406
406
407
407
408
408
409
409
410
410
411
411
412
412
413
413
414
414
415
415
416
416
417
417
418
418
419
419
420
420
421
421
422
422
423
423
424
424
425
425
426
426
427
427
428
428
429
429
430
430
431
431
432
432
433
433
434
434
435
435
436
436
437
437
438
438
439
439
440
440
441
441
442
442
443
443
444
444
445
445
446
446
447
447
448
448
449
449
450
450
451
451
452
452
453
453
454
454
455
455
456
456
457
457
458
458
459
459
460
460
461
461
462
462
463
463
464
464
465
465
466
466
467
467
468
468
469
469
470
470
471
471
472
472
473
473
474
474
475
475
476
476
477
477
478
478
479
479
480
480
481
481
482
482
483
483
484
484
485
485
486
486
487
487
488
488
489
489
490
490
491
491
492
492
493
493
494
494
495
495
496
496
497
497
498
498
499
499
500
500
501
501
502
502
503
503
504
504
505
505
506
506
507
507
508
508
509
509
510
510
511
511
512
512
513
513
514
514
515
515
516
516
517
517
518
518
519
519
520
520
521
521
522
522
523
523
524
524
525
525
526
526
527
527
528
528
529
529
530
530
531
531
532
532
533
533
534
534
535
535
536
536
537
537
538
538
539
539
540
540
541
541
542
542
543
543
544
544
545
545
546
546
547
547
548
548
549
549
550
550
551
551
552
552
553
553
554
554
555
555
556
556
557
557
558
558
559
559
560
560
561
561
562
562
563
563
564
564
565
565
566
566
567
567
568
568
569
569
570
570
571
571
572
572
573
573
574
574
575
575
576
576
577
577
578
578
579
579
580
580
581
581
582
582
583
583
584
584
585
585
586
586
587
587
588
588
589
589
590
590
591
591
592
592
593
593
594
594
595
595
596
596
597
597
598
598
599
599
600
600
601
601
602
602
603
603
604
604
605
605
606
606
607
607
608
608
609
609
610
610
611
611
612
612
613
613
614
614
615
615
616
616
617
617
618
618
619
619
620
620
621
621
622
622
623
623
624
624
625
625
626
626
627
627
628
628
629
629
630
630
631
631
632
632
633
633
634
634
635
635
636
636
637
637
638
638
639
639
640
640
641
641
642
642
643
643
644
644
645
645
646
646
647
647
648
648
649
649
650
650
651
651
652
652
653
653
654
654
655
655
656
656
657
657
658
658
659
659
660
660
661
661
662
662
663
663
664
664
665
665
666
666
667
667
668
668
669
669
670
670
671
671
672
672
673
673
674
674
675
675
676
676
677
677
678
678
679
679
680
680
681
681
682
682
683
683
684
684
685
685
686
686
687
687
688
688
689
689
690
690
691
691
692
692
693
693
694
694
695
695
696
696
697
697
698
698
699
699
700
700
701
701
702
702
703
703
704
704
705
705
706
706
707
707
708
708
709
709
710
710
711
711
712
712
713
713
714
714
715
715
716
716
717
717
718
718
719
719
720
720
721
721
722
722
723
723
724
724
725
725
726
726
727
727
728
728
729
729
730
730
731
731
732
732
733
733
734
734
735
735
736
736
737
737
738
738
739
739
740
740
741
741
742
742
743
743
744
744
745
745
746
746
747
747
748
748
749
749
750
750
751
751
752
752
753
753
754
754
755
755
756
756
757
757
758
758
759
759
760
760
761
761
762
762
763
763
764
764
765
765
766
766
767
767
768
768
769
769
770
770
771
771
772
772
773
773
774
774
775
775
776
776
777
777
778
778
779
779
780
780
781
781
782
782
783
783
784
784
785
785
786
786
787
787
788
788
789
789
790
790
791
791
792
792
793
793
794
794
795
795
796
796
797
797
798
798
799
799
800
800
801
801
802
802
803
803
804
804
805
805
806
806
807
807
808
808
809
809
810
810
811
811
812
812
813
813
814
814
815
815
816
816
817
817
818
818
819
819
820
820
821
821
822
822
823
823
824
824
825
825
826
826
827
827
828
828
829
829
830
830
831
831
832
832
833
833
834
834
835
835
836
836
837
837
838
838
839
839
840
840
841
841
842
842
843
843
844
844
845
845
846
846
847
847
848
848
849
849
850
850
851
851
852
852
853
853
854
854
855
855
856
856
857
857
858
858
859
859
860
860
861
861
862
862
863
863
864
864
865
865
866
866
867
867
868
868
869
869
870
870
871
871
872
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
880
881
881
882
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
890
891
891
892
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
900
901
901
902
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
910
911
911
912
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1400
1401
1401
1402
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1410
1411
1411
1412
1412
1413
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1420
1421
1421
1422
1422
1423
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1430
1431
1431
1432
1432
1433
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1440
1441
1441
1442
1442
1443
1443
1444
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1450
1451
1451
1452
1452
1453
1453
1454
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1460
1461
1461
1462
1462
1463
1463
1464
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1470
1471
1471
1472
1472
1473
1473
1474
1474
1475
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
1480
1481
1481
1482
1482
1483
14
```

Slika 15. Back-end dio početnog zaslona aplikacije

Slika 16 prikazuje proces izrade back-end dijela aplikacije koji služi za izračunavanje prave vrijednosti i predviđene vrijednosti. Nakon unošenja brojeva otvara se mogućnost izračuna pogreške koji će biti prikazan na slici 17.

The screenshot shows the Android Studio interface with the project structure on the left and the code editor on the right. The code editor displays the `IzracunActivity` class, which extends `AppCompatActivity` . The class contains methods for handling the toolbar, setting up lists for origin and destination values, and displaying a toast message. It also includes an `onOptionsItemSelected` method and a `prevtoriDouble` helper method.

```
IzracunActivity | onCreate()
1 package hr.ett.zavrsmi.mapleapp;
2
3 import ...
4
5 public class IzracunActivity extends AppCompatActivity {
6
7     @BindView(R.id.toolbar) Toolbar toolbar;
8     @BindView(R.id.listViewOrigenValues) ListView origlista;
9     @BindView(R.id.listViewPredValues) ListView predlista;
10    double[] origVrijednosti;
11    double[] predVrijednosti;
12
13    @Override
14    protected void onCreate(Bundle savedInstanceState) {
15        super.onCreate(savedInstanceState);
16        setContentView(R.layout.activity_izracun);
17        ButterKnife.bind(this);
18        // custom toolbar
19        setSupportActionBar(toolbar);
20        getSupportActionBar().setDisplayShowTitleEnabled(false);
21
22        Intent intent = getIntent();
23        origVrijednosti = intent.getDoubleArrayExtra("brojevi");
24
25        ArrayAdapter adapter = new ArrayAdapter(context, this,
26                R.layout.activity_listview, prevtoriDouble(origVrijednosti));
27        origlista.setAdapter(adapter);
28
29        Toast.makeText(getApplicationContext(), "Vidoviti Milan...", Toast.LENGTH_SHORT).show();
30        pokreniAlgoritam(origVrijednosti);
31
32    }
33
34    @OnClick(R.id.btmInfor)
35    public void openInfo() {
36        Intent intent = new Intent(packageContext: IzracunActivity.this, InfoActivity.class);
37        startActivity(intent);
38    }
39
40    public Double[] prevtoriDouble(double[] lista) {
41        Double[] uredjenaLista = new Double[lista.length];
42        ...
43    }
44
45
46    @Override
47    public void onBackPressed() {
48        Intent intent = new Intent(packageContext: IzracunActivity.this, InfoActivity.class);
49        startActivity(intent);
50    }
51
52
53 }
```

Slika 16. Back-end dio izračuna prave vrijednosti i predviđene vrijednosti

Slika 17 prikazuje back-end dio aplikacije koja opisuje proces prilikom koje se računa greška. Nakon pritiska na gumb izračunaj grešku otvara se novi zaslon koji ispisuje prosječnu grešku te predviđenu slijedeću vrijednost.

The screenshot shows the Android Studio interface. On the left, the project structure is visible under the 'app' folder, including 'manifests', 'java' (with 'GreskaActivity' selected), 'res', and 'Gradle Scripts'. The right side displays the code for 'GreskaActivity.java'.

```

    package hr.etf.zavrsni.mapleapp;
    import ...
    public class GreskaActivity extends AppCompatActivity {
        @BindView(R.id.toolbar) Toolbar toolbar;
        @BindView(R.id.listViewPredValues) ListView predLista;
        @BindView(R.id.listViewMistakeValues) ListView mistakesLista;
        @BindView(R.id.textViewAvg) TextView avgBrojTextView;
        @BindView(R.id.textViewNext) TextView nextBrojTextView;

        @Override
        protected void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.activity_greska);
            ButterKnife.bind(this);
            setSupportActionBar(toolbar);
            getSupportActionBar().setDisplayHomeAsUpEnabled(false);

            Intent intent = getIntent();
            double[] origVrijednosti = intent.getDoubleArrayExtra("origVrijednosti");
            double[] predVrijednosti = intent.getDoubleArrayExtra("predVrijednosti");

            Double[] origDouble = pretvoriuDouble(origVrijednosti);

            Double[] predVDouble = pretvoriuDouble(predVrijednosti);
            Double[] pV = new Double[5];
            // kako bi prikazao prvih 5
            for(int i = 0; i<5; i++){
                pV[i] = predVDouble[i];
            }
            ArrayAdapter adapter = new ArrayAdapter(context: this, R.layout.activity_listview, formatirajRezultat(pV));
            predLista.setAdapter(adapter);

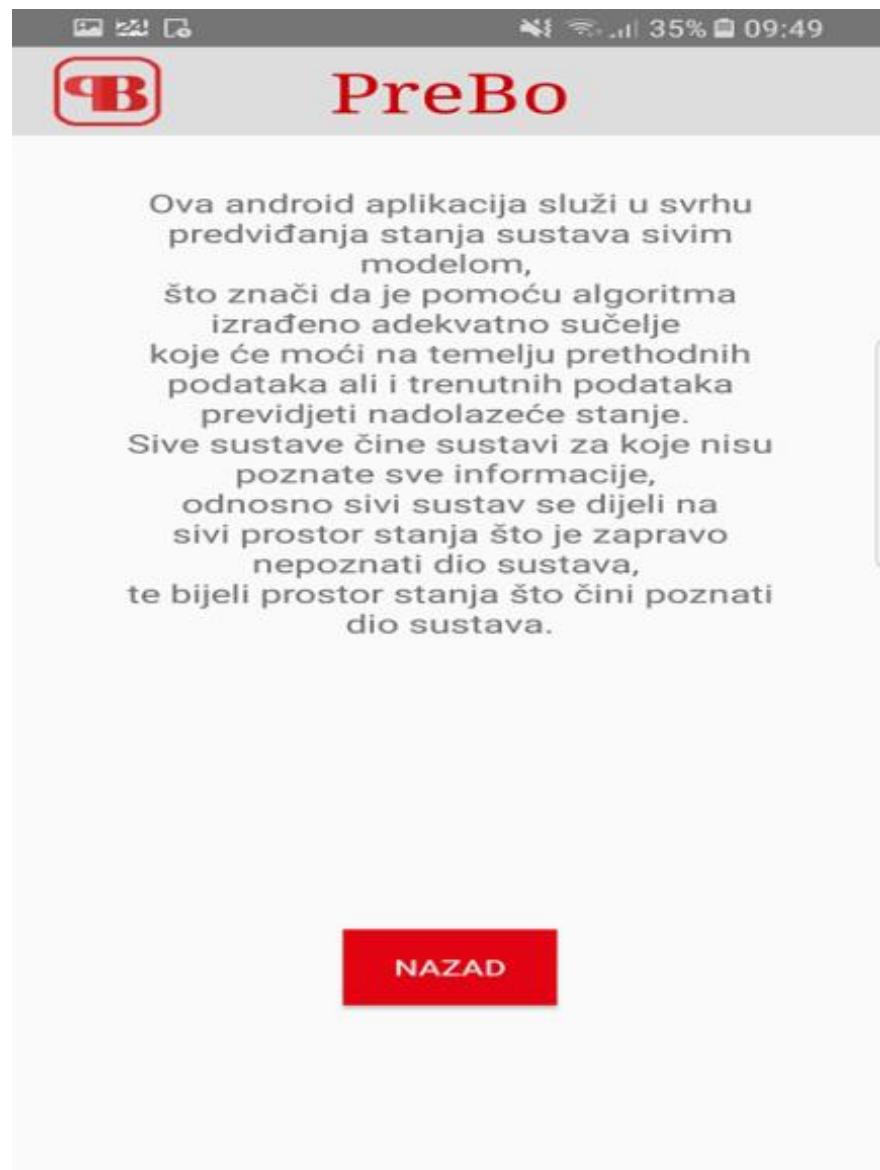
            Toast.makeText(getApplicationContext(), text: "Računanje grešaka...", Toast.LENGTH_SHORT).show();
            Double[] greske = izracunaGreske(origDouble, pV);
            ArrayAdapter adapter2 = new ArrayAdapter(context: this, R.layout.activity_listview, formatirajRezultat(greske));
            mistakesLista.setAdapter(adapter2);
        }
        Double[] avgGreska = null;
    }

```

At the bottom of the code editor, there are tabs for 'Terminal', 'Logcat', 'Messages', and 'TODO'.

Slika 17. Back-end dio izračunavanja prosječne pogreške i predviđene sljedeće vrijednosti

3.4 Slike iz funkcionalne aplikacije u nekoliko koraka



Slika 18. Info aplikacije

The screenshot shows a mobile application interface for PreBo. At the top, there is a header with the PreBo logo and a red 'B' icon. Below the header, there is a text input field with the placeholder 'Upišite brojeve:' (Enter numbers). There are four horizontal lines for input. At the bottom of the input area is a red button labeled 'IZRAČUNAJ' (Calculate).

Slika 19. Unos brojeva

The screenshot shows the results of a calculation. It displays two columns: 'Prava vrijednost' (True value) and 'Predviđena vrijednost' (Predicted value). The data is as follows:

Prava vrijednost	Predviđena vrijednost
35537.8	35537.8
29339.1	29359.211
27723.2	27655.058
26024.2	26049.822
26406.8	24537.762

At the bottom is a red button labeled 'IZRAČUNAJ GREŠKU' (Calculate error).

Slika 20. Unešeni brojevi

The screenshot shows the results of a calculation with error percentages. It displays two columns: 'Predviđena vrijednost' (Predicted value) and 'Pogreška (%)' (Error (%)). The data is as follows:

Predviđena vrijednost	Pogreška (%)
35537.8	0
29359.211	0.069
27655.058	0.246
26049.822	0.099
24537.762	7.078

At the bottom, there is a note: 'Prosječna greška: 1.50%' (Average error: 1.50%) and 'Predviđena sljedeća vrijednost: 23113.47' (Predicted next value: 23113.47).

Slika 21. Izračun pogreške i predviđena vrijednost

4. ZAKLJUČAK

Ovaj rad predstavlja upute za izradu Android aplikacije s primjenom za sivo predviđanje. Teorijski dio dokumenta prikazuje pojedina načela koja su bitna kako bi se općenito predstavio Android operativni sustav. Navedeni su pojedini modeli sivog predviđanja te opisana funkcionalnost ali i oprez kod pojedinih aspekata poput nejednakog vremenskog intervala prilikom uzimanja ulaznih podataka.

Aplikacija je obavljena uz pomoć alata "Android studio" koji uz dodatne alate poput "Android SDK" omogućavaju korisnicima izradu aplikacija za Android operativni sustav.

Izrada aplikacije prikazana je kroz praktični dio od samoga izgleda odnosno dizajna do implementacije algoritma ali i jednostavnih stvari poput definiranja biblioteka i sl.

5. SAŽETAK

U ovom radu prikazani su alati "Android studio" i "SDK" kao temelj za izradu pravovaljane aplikacije. Opisan je postupak računanja sivog predviđanja. Prikazana je izrada Android aplikacije u navedenim alatima. Prikazan je svaki korak tijekom izrade kao i prilikom implementacije java programa, odnosno algoritma za sivo predviđanje. Funkcionalnost aplikacije prikazana je na primjeru izračuna za sivo predviđanje otpada u Slavonskom Brodu.

Ključne riječi: Algoritam, Android, Android studio, Java, SDK

6. ABSTRACT

Title: Creating Android operating system application

This paper presents the “Android studio” and “SDK” tools as the basics for creating a legitimate application. The gray prediciton calculation procedure is described. Creating of Android application is shown using the listed tools. Each step during the design is presented as well as implementing the Java program, or the gray prediction algorithm. The functionality of the application is shown in the calculation example for gray prediction of garbage in Slavonski Brod.

Keywords : Algorithm,Android,Android studio,Java,SDK

LITERATURA

[1] Logo Android operacijskog sustava, lipanj, 2016.

[http://hr.wikipedia.org/wiki/Android_\(operacijski_sustav\)](http://hr.wikipedia.org/wiki/Android_(operacijski_sustav))

[2] Android verzije, lipanj, 2016.

<https://developer.android.com>

[3] Prikaz android arhitekture, lipanj, 2016.

http://www.tutorialspoint.com/android/android_architecture.html

[4] Predviđanje održavanja tehničkog sustava procjenom stanja, Predviđanje stanja sustava sivim predviđanjem(pdf document).

[5] Grey systems: Modeling and prediction, Kun-li Wen, 2004. godine by Yangsky.

[6] Prikaz konkurentnosti Android uređaja na svjetskom tržištu, travanj, 2018.

<https://www.idc.com/promo/smartphone-market-share/os>

[7] Prikaz Android operativnih sustava prema verzijama i trenutnom korištenju, travanj 2018.

<http://mobiclass.csc.ncsu.edu/2012/12/find-indystate-android-os-ice-cream.html>

[8] Prikaz Samsung Chromebook-a, laptop s Android operativnim sustavom, travanj 2018.

<http://www.upgrademag.com/web/2014/03/21/samsung-pldt-enable-smes-collaborate-work-cloud/>

[9] Android Google uređaj Nexus, travanj 2018.

<https://www.livemint.com/Leisure/iJdAuCEgy6xkZTTGpRo8cN/Are-Google-Nexus-and-Samsung-Galaxy-devices-the-only-safe-An.html>

[10] Prikaz Android arhitekture, travanj 2018.

<http://shubhamjain.space/wp-content/uploads/2016/09/android-architecture.png>

ŽIVOTOPIS

Božidar Javor rođen je 16.srpnja.1991.godine . Pohađao je osnovnu školu Antun Mihanović u Slavonskom Brodu u vremenu od 1998-2006. Srednju Tehničku školu u Slavonskom Brodu upisuje 2006. godine s naznakom smjera elektrotehničar. Srednju Tehničku školu završava 2010. godine te upisuje Elektrotehnički fakultet u Osijeku s naznakom smjera informatika. Krajem 2015-te godine odlazi u inozemstvo i zapošljava se u informatičkoj firmi kao tehničar koji održava rad na računalima.

PRILOG 1:IZVORNI DIO APLIKACIJE

```
package hr.etf.zavrsni.mapleapp;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;
import android.widget.Toolbar;

import java.math.RoundingMode;
import java.text.DecimalFormat;

import butterknife.BindView;
import butterknife.ButterKnife;
import butterknife.OnClick;

public class GreskaActivity extends AppCompatActivity {

    @BindView(R.id.toolbar) Toolbar toolbar;
    @BindView(R.id.listViewPredValues) ListView predLista;
    @BindView(R.id.listViewMistakeValues) ListView mistakesLista;
    @BindView(R.id.textViewAvg) TextView avgBrojTextView;
    @BindView(R.id.textViewNext) TextView nextBrojTextView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_greska);
        ButterKnife.bind(this);
        setActionBar(toolbar);
        getActionBar().setDisplayShowTitleEnabled(false);

        Intent intent = getIntent();
        double[] origVrijednosti =
intent.getDoubleArrayExtra("origVrijednosti");
        double[] predVrijednosti =
intent.getDoubleArrayExtra("predVrijednosti");

        Double[] origVDouble = pretvoriuDouble(origVrijednosti);
        Double[] predVDouble = pretvoriuDouble(predVrijednosti);
        Double[] pv = new Double[5];
        // kako bi prikazao prvih 5
        for(int i = 0; i<5; i++){
            pv[i] = predVDouble[i];
        }
        ArrayAdapter adapter = new ArrayAdapter(this,
R.layout.activity_listview, formatirajRezultat(pv));
        predLista.setAdapter(adapter);

        Toast.makeText(getApplicationContext(), "Računanje grešaka...", Toast.LENGTH_SHORT).show();
        Double[] greske = izracunajGreske(origVDouble, pv);
        ArrayAdapter adapter2 = new ArrayAdapter(this,
R.layout.activity_listview, formatirajRezultat(greske));
        mistakesLista.setAdapter(adapter2);

        Double avgGreska = 0.0;
```

```

        Double nextValue = 0.0;
        for(int i = 0; i<greske.length; i++){
            avgGreska += greske[i];
        }
        avgGreska = avgGreska / 5;
        nextValue = predvDouble[5];

        avgBrojTextView.setText(avgBrojTextView.getText() +
String.format("%.2f", avgGreska) + "%");
        nextBrojTextView.setText(nextBrojTextView.getText() + ""
+String.format("%.2f", nextValue));
    }

    private Double[] izracunajGreske(Double[] origv,Double[] predv) {
        Double[] greske = new Double[5];
        for(int i = 0; i < 5; i++){
            // formula za greške
            greske[i] = Math.abs((origv[i] - predv[i])/origv[i]) * 100;
        }
        return greske;
    }

    private String[] formatirajRezultat(Double[] predvDouble) {
        DecimalFormat dcf = new DecimalFormat("#.###");
        dcf.setRoundingMode(RoundingMode.CEILING);
        String[] formatirano = new String[predvDouble.length];
        for (int i =0; i < predvDouble.length; i++) {
            formatirano[i] = dcf.format(predvDouble[i]);
        }
        return formatirano;
    }

    public Double[] pretvoriuDouble(double[] lista){
        Double[] uredjenaLista = new Double[lista.length];
        try{
            for(int i = 0; i<lista.length; i++){
                uredjenaLista[i] = lista[i];
            }
        }
        catch (Exception e){
            Toast.makeText(getApplicationContext(), "Dogodila se pogreška",
Toast.LENGTH_SHORT).show();
        }
        return uredjenaLista;
    }

    @OnClick(R.id.btnAdd)
    public void openInfo(){
        Intent intent = new Intent(GreskaActivity.this, InfoActivity.class);
        startActivity(intent);
    }
}

```

```

package hr.etf.zavrsni.mapleapp;

import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.Toast;
import android.widget.Toolbar;

```

```

import java.math.RoundingMode;
import java.text.DecimalFormat;

import butterknife.BindView;
import butterknife.ButterKnife;
import butterknife.OnClick;

public class IzracunActivity extends AppCompatActivity {

    @BindView(R.id.toolbar) Toolbar toolbar;
    @BindView(R.id.listOrigValues) ListView origLista;
    @BindView(R.id.listPredValues) ListView predLista;
    double[] origVrijednosti;
    double[] predVrijednosti;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_izracun);
        ButterKnife.bind(this);
        // custom toolbar
        setSupportActionBar(toolbar);
        getSupportActionBar().setDisplayHomeAsUpEnabled(false);

        Intent intent = getIntent();
        origVrijednosti = intent.getDoubleArrayExtra("brojevi");

        ArrayAdapter adapter = new ArrayAdapter(this,
                R.layout.activity_listview,
                pretvoriuDouble(origVrijednosti));
        origLista.setAdapter(adapter);

        Toast.makeText(getApplicationContext(), "Vidoviti Milan...", Toast.LENGTH_SHORT).show();
        pokreniAlgoritam(origVrijednosti);
    }

    @OnClick(R.id.btnInfo)
    public void openInfo(){
        Intent intent = new Intent(IzracunActivity.this, InfoActivity.class);
        startActivity(intent);
    }

    public Double[] pretvoriuDouble(double[] lista){
        Double[] uredjenaLista = new Double[lista.length];
        try{
            for(int i = 0; i<lista.length; i++){
                uredjenaLista[i] = lista[i];
            }
        } catch (Exception e){
            Toast.makeText(getApplicationContext(), "Dogodila se pogreška",
                    Toast.LENGTH_SHORT).show();
        }
        return uredjenaLista;
    }

    public void pokreniAlgoritam(double[] origVrijednosti){
        // origVrijednosti uvijek ima 5
        // agoNiz - 4 broja
        double[] agoNiz = {origVrijednosti[0],
                           origVrijednosti[0] + origVrijednosti[1],
                           origVrijednosti[0] + origVrijednosti[1] +
                           origVrijednosti[2],
                           origVrijednosti[1] + origVrijednosti[2],
                           origVrijednosti[2] + origVrijednosti[3],
                           origVrijednosti[3] + origVrijednosti[4]};
    }
}

```

```

        origVrijednosti[0] + origVrijednosti[1] +
origVrijednosti[2] + origVrijednosti[3]};

// srednjiNiz - 3 broja
double[] srednjiNiz = {(agoNiz[0] + agoNiz[1])/2,
(agoNiz[1] + agoNiz[2])/2,
(agoNiz[2] + agoNiz[3])/2};

double[][] matricaY = {{origVrijednosti[1]}, {origVrijednosti[2]},
{origVrijednosti[3]}};

double[][] matricaB = {{-srednjiNiz[0], 1}, {-srednjiNiz[1], 1}, {-srednjiNiz[2], 1}};

double[][] matricaTransB = transponirajMatricu(matricaB);
double[][] tempMatrica1 = pomnoziMatrice(matricaTransB, matricaB);
double[][] tempMatrica2 = inverzMatrice(tempMatrica1);
double[][] tempMatrica3 = pomnoziMatrice(tempMatrica2,
matricaTransB);
double[][] tempMatrica4 = pomnoziMatrice(tempMatrica3, matricaY);
// tempMatrica4 se sastoji od 2 reda i 1 stupca (a/b)
// a,b - koeficijenti
double a = tempMatrica4[0][0];
double b = tempMatrica4[1][0];

predVrijednosti = predvidivrijednosti(origVrijednosti, a, b);

Double[] predVDouble = pretvoriuDouble(predVrijednosti);
Double[] pV = new Double[5];
// kako bi prikazao prvih 5
for(int i = 0; i<5; i++){
    pV[i] = predVDouble[i];
}
ArrayAdapter adapter = new ArrayAdapter(this,
    R.layout.activity_listview, formatirajRezultat(pV));
predLista.setAdapter(adapter);
}

private String[] formatirajRezultat(Double[] predVDouble) {
DecimalFormat dcf = new DecimalFormat("#.###");
dcf.setRoundingMode(RoundingMode.CEILING);
String[] formatirano = new String[predVDouble.length];
for (int i =0; i < predVDouble.length; i++) {
    formatirano[i] = dcf.format(predVDouble[i]);
}
return formatirano;
}

private double[][] inverzMatrice(double[][] m) {
// kvadratna matrica - jednak broj stupaca i redaka
double sk = 1/((m[0][0] * m[1][1]) - (m[0][1] * m[1][0]));
double[][] tmpMatrica = {{m[1][1], -m[0][1]},
{-m[1][0], m[0][0]}};
double[][] rezMatrica = {{sk * tmpMatrica[0][0], sk *
tmpMatrica[0][1]}, {sk * tmpMatrica[1][0], sk *
tmpMatrica[1][1]}];
return rezMatrica;
}

private double[][] pomnoziMatrice(double[][] matricaTransB, double[][] matricaB) {

int tranRows = matricaTransB.length;
int tranColumns = matricaTransB[0].length;

```

```

        int bRows = matricaB.length;
        int bColumns = matricaB[0].length;

        if (tranColumns != bRows) {
            throw new IllegalArgumentException("Broj redaka transponirane: "
+ tranColumns +
                " ne odgovara broju stupaca B matrice: " + bRows + ".");
        }
        double[][] rezMatrica = new double[tranRows][bColumns];
        for (int i = 0; i < tranRows; i++) {
            for (int j = 0; j < bColumns; j++) {
                for (int k = 0; k < tranColumns; k++) {
                    rezMatrica[i][j] += matricaTransB[i][k] * matricaB[k][j];
                }
            }
        }
        return rezMatrica;
    }

    private double[][] transponirajMatricu(double[][] matricaB) {
        int m = matricaB.length;
        int n = matricaB[0].length;
        double[][] transB = new double[n][m];
        for (int i = 0; i < m; i++)
            for (int j = 0; j < n; j++)
                transB[j][i] = matricaB[i][j];
        return transB;
    }

    private double[] predvidivrijednosti(double[] origVrijednosti, double a,
double b){
        double[] predVrijednosti = new double[6];
        // prva vrijednost se prepisuje
        predVrijednosti[0] = origVrijednosti[0];
        for(int k = 1; k <= origVrijednosti.length; k++){
            predVrijednosti[k] = (1 - Math.exp(a))*(origVrijednosti[0] -
b/a)*Math.exp(-a * k);
        }
        return predVrijednosti;
    }

    @OnClick(R.id.btnIzracunajPogresku)
    public void otvoriGreskaActivity(){
        Intent intent = new Intent(IzracunActivity.this,
GreskaActivity.class);
        intent.putExtra("origVrijednosti", origVrijednosti);
        intent.putExtra("predVrijednosti", predVrijednosti);
        startActivity(intent);
        finish();
    }
}
}

```

```

package hr.etf.zavrsni.mapleapp;

import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.Toast;
import android.widget.Toolbar;

import java.math.RoundingMode;

```

```

import java.text.DecimalFormat;
import butterknife.BindView;
import butterknife.ButterKnife;
import butterknife.OnClick;

public class IzracunActivity extends AppCompatActivity {

    @BindView(R.id.toolbar) Toolbar toolbar;
    @BindView(R.id.listOrigValues) ListView origLista;
    @BindView(R.id.listPredValues) ListView predLista;
    double[] origVrijednosti;
    double[] predVrijednosti;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_izracun);
        ButterKnife.bind(this);
        // custom toolbar
        setSupportActionBar(toolbar);
        getSupportActionBar().setDisplayHomeAsUpEnabled(false);

        Intent intent = getIntent();
        origVrijednosti = intent.getDoubleArrayExtra("brojevi");

        ArrayAdapter adapter = new ArrayAdapter(this,
                R.layout.activity_listview,
                pretvoriuDouble(origVrijednosti));
        origLista.setAdapter(adapter);

        Toast.makeText(getApplicationContext(), "Vidoviti Milan...", Toast.LENGTH_SHORT).show();
        pokreniAlgoritam(origVrijednosti);
    }

    @OnClick(R.id.btnInfo)
    public void openInfo(){
        Intent intent = new Intent(IzracunActivity.this, InfoActivity.class);
        startActivity(intent);
    }

    public Double[] pretvoriuDouble(double[] lista){
        Double[] uredjenaLista = new Double[lista.length];
        try{
            for(int i = 0; i<lista.length; i++){
                uredjenaLista[i] = lista[i];
            }
        } catch (Exception e){
            Toast.makeText(getApplicationContext(), "Dogodila se pogreška",
                    Toast.LENGTH_SHORT).show();
        }
        return uredjenaLista;
    }

    public void pokreniAlgoritam(double[] origVrijednosti){
        // origVrijednosti uvijek ima 5
        // agoNiz - 4 broja
        double[] agoNiz = {origVrijednosti[0],
                           origVrijednosti[0] + origVrijednosti[1],
                           origVrijednosti[0] + origVrijednosti[1] +
                           origVrijednosti[2],
                           origVrijednosti[0] + origVrijednosti[1] +
                           origVrijednosti[2] + origVrijednosti[3]};
    }
}

```

```

// srednjiNiz - 3 broja
double[] srednjiNiz = {(agoNiz[0] + agoNiz[1])/2,
                       (agoNiz[1] + agoNiz[2])/2,
                       (agoNiz[2] + agoNiz[3])/2};

double[][] matricaY = {{origVrijednosti[1]}, {origVrijednosti[2]},
{origVrijednosti[3]}};

double[][] matricaB = {{-srednjiNiz[0], 1}, {-srednjiNiz[1], 1}, {-srednjiNiz[2], 1}};

double[][] matricaTransB = transponirajMatricu(matricaB);
double[][] tempMatrica1 = pomnoziMatrice(matricaTransB, matricaB);
double[][] tempMatrica2 = inverzMatrice(tempMatrica1);
double[][] tempMatrica3 = pomnoziMatrice(tempMatrica2,
matricaTransB);
double[][] tempMatrica4 = pomnoziMatrice(tempMatrica3, matricaY);
// tempMatrica4 se sastoji od 2 reda i 1 stupca (a/b)
// a,b - koeficijenti
double a = tempMatrica4[0][0];
double b = tempMatrica4[1][0];

predVrijednosti = predvidivrijednosti(origVrijednosti, a, b);

Double[] predVDouble = pretvoriuDouble(predVrijednosti);
Double[] pv = new Double[5];
// kako bi prikazao prvih 5
for(int i = 0; i<5; i++){
    pv[i] = predVDouble[i];
}
ArrayAdapter adapter = new ArrayAdapter(this,
    R.layout.activity_listview, formatirajRezultat(pv));
predLista.setAdapter(adapter);
}

private String[] formatirajRezultat(Double[] predVDouble) {
    DecimalFormat dcf = new DecimalFormat("#.###");
    dcf.setRoundingMode(RoundingMode.CEILING);
    String[] formatirano = new String[predVDouble.length];
    for (int i =0; i < predVDouble.length; i++) {
        formatirano[i] = dcf.format(predVDouble[i]);
    }
    return formatirano;
}

private double[][] inverzMatrice(double[][] m) {
    // kvadratna matrica - jednak broj stupaca i redaka
    double sk = 1/((m[0][0] * m[1][1]) - (m[0][1] * m[1][0]));
    double[][] tmpMatrica = {{m[1][1], -m[0][1]},
                             {-m[1][0], m[0][0]}};
    double[][] rezMatrica = {{sk * tmpMatrica[0][0], sk *
tmpMatrica[0][1]},
                             {sk * tmpMatrica[1][0], sk *
tmpMatrica[1][1]}];
    return rezMatrica;
}

private double[][] pomnoziMatrice(double[][] matricaTransB, double[][] matricaB) {

    int tranRows = matricaTransB.length;
    int tranColumns = matricaTransB[0].length;
    int bRows = matricaB.length;
    int bColumns = matricaB[0].length;

```

```

        if (tranColumns != bRows) {
            throw new IllegalArgumentException("Broj redaka transponirane: "
+ tranColumns +
                " ne odgovara broju stupaca B matrice: " + bRows + ".");
        }
        double[][] rezMatrica = new double[tranRows][bColumns];
        for (int i = 0; i < tranRows; i++) {
            for (int j = 0; j < bColumns; j++) {
                for (int k = 0; k < tranColumns; k++) {
                    rezMatrica[i][j] += matricaTransB[i][k] * matricaB[k][j];
                }
            }
        }
        return rezMatrica;
    }

    private double[][] transponirajMatricu(double[][] matricaB) {
        int m = matricaB.length;
        int n = matricaB[0].length;
        double[][] transB = new double[n][m];
        for (int i = 0; i < m; i++) {
            for (int j = 0; j < n; j++)
                transB[j][i] = matricaB[i][j];
        }
        return transB;
    }

    private double[] predvidivrijednosti(double[] origVrijednosti, double a,
double b){
        double[] predVrijednosti = new double[6];
        // prva vrijednost se prepisuje
        predVrijednosti[0] = origVrijednosti[0];
        for(int k = 1; k <= origVrijednosti.length; k++){
            predVrijednosti[k] = (1 - Math.exp(a))*(origVrijednosti[0] -
b/a)*Math.exp(-a * k);
        }
        return predVrijednosti;
    }

    @OnClick(R.id.btnIzracunajPogresku)
    public void otvoriGreskaActivity(){
        Intent intent = new Intent(IzracunActivity.this,
GreskaActivity.class);
        intent.putExtra("origVrijednosti", origVrijednosti);
        intent.putExtra("predVrijednosti", predVrijednosti);
        startActivity(intent);
        finish();
    }
}
}

```

```

package hr.etf.zavrsni.mapleapp;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.Toolbar;

import butterknife.BindView;
import butterknife.ButterKnife;
import butterknife.OnClick;

public class InfoActivity extends AppCompatActivity {

```

```
@Bindview(R.id.toolbar) Toolbar toolbar;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
     super.onCreate(savedInstanceState);  
     setContentView(R.layout.activity_info);  
     ButterKnife.bind(this);  
     // custom toolbar  
     setActionBar(toolbar);  
     getActionBar().setDisplayShowTitleEnabled(false);  
 }  
  
 @OnClick(R.id.btnExit)  
 public void goBack(){  
     onBackPressed();  
 }  
}
```