 Univerzitet u Novom Sadu

Tehnički fakultet „Mihajlo Pupin“

Zrenjanin

Predmet: Projektovanje softvera

Seminarski rad

*aplikacija – Dostava hrane*

Profesor: Student:

[Doc. dr Eleonora Brtka](http://www.tfzr.uns.ac.rs/Kadar/NastavnoOsoblje/20) Slobodan Kirćanski SI 16/18

Smer: Softversko inženjerstvo

Zrenjanin, 2019.

Sadržaj

[Predmetni cilj 2](#_Toc42206149)

[Cilj seminarskog rada 3](#_Toc42206150)

[1. Uvod 4](#_Toc42206151)

[2. Korišćene tehnologije 5](#_Toc42206152)

[2.1. Eclipse 5](#_Toc42206153)

[2.2. Java programski jezik 5](#_Toc42206154)

[2.3. WindowBuilder 5](#_Toc42206155)

[2.4. JFrame 5](#_Toc42206156)

[2.5. JDialog 5](#_Toc42206157)

[3. Postupak kreiranja aplikacije 6](#_Toc42206158)

[3.1. Prozor narucivanjeHrane.java 6](#_Toc42206159)

[3.2. Prozor narucivanjeGlavniP.java 7](#_Toc42206160)

[3.2.1. Prvi panel 7](#_Toc42206161)

[3.2.2. Drugi panel 8](#_Toc42206162)

[4. Ključni delovi koda 8](#_Toc42206163)

[5. Zaključak 11](#_Toc42206164)

[LITERATURA 12](#_Toc42206165)

# Predmetni cilj

Osposobljenost za samostalnu konstrukciju softvera koja je bazirana na standardnim procesima implementacije i testiranja.Osposobljenost za kreiranje projektne dokumentacije

# Cilj seminarskog rada

Cilj seminarskog rada je kreiranje aplikacije za naručivanje brze hrane koristeci *Windows Builder* i koriscenjem *Jframe-a* i *JDialog-a.*

.

# 1. Uvod

Aplikacija se sastoji od prozora koji *je Frame* i koji omogucava korisniku da se uloguje kako bi pristupio sledecem, glavnom, prozoru koji je *JFrame* i koji sadrži dva panela(dve stranice). Prvi panel sadrži listu obroka i dugmiće za više informacija o njima. Kada korisnik odabere određeni obrok ili obroke prelazi na sledeću stranicu koja je *Panel* i na kojem se nalazi račun, i prostor za informacije o korisniku. Korisnik može da prelazi sa stranice na stranicu u svakom trenutku, čuvaju se prošle odluke. Postoje 7 *JFrame* prozora koji sadrze više informacija o obrocima. Ispunjavanjem porudžbine i klikom na dugme *Poruci* pojavljuje se prozor *JDialog* koji se zove *Potvrda* gde se klikom na dugme *OK* završava naručivanje.

# 2. Korišćene tehnologije

Korišćene tehnologije predstavljaju tehnologije koje su bile upotrebljene prilikom izrade aplikacije.

## 2.1. Eclipse

Eklips (engl. Eclipse) je programska razvojna okolina (IDE) pisana u Javi, a može se koristiti za razvoj aplikacija u raznim programskim jezicima kao što su Java, Ada, C, C++, COBOL, Perl, PHP, Piton, R, Rubi (uključujući Ruby on Rails okolinu), Scala, Clojure i Scheme. Isto tako, može se koristiti za razvoj delova aplikacije Mathematica. Razvojna okolina (IDE) često se naziva Eklips ADA za Adu, Eklips CDT za C / C++, Eklips JDT za Javu i Eklips PDT za PHP.

## 2.2. Java programski jezik

Radi se o objektno orijentiranom programskom jeziku. Pojam objektno orijentirano programiranje znači metodu razvijanja programske opreme kod koje su programi koncipirani poput grupe objekata koji međusobno deluju. Programski jezik JAVA jednostavniji je za korištenje od programskog jezika C ili C++.Jedna od bitnijih značajki je „neovisnost o platformi“. Ovo omogućuje da se kod napisan u JAVA-i izvodi bez unošenja promena u različitim okruženjima na računarima (Windows, UNIX, …).JAVA kod prevodi se u u tzv. „bytecode“ kojeg pokreće operacijski sustav, drugi program ili uređaji pomoću JAVA interpretera. Još jedna u nizu značaji programskog jezika JAVA bila bi „višenitnost“, a znači podršku za višenitno programiranje. JAVA je dizajnirana za izvršavanje u distribuiranom okruženju, prvenstveno na internetu.

## 2.3. WindowBuilder

*WindowsBuilder* je mocan i lak za koriscenje „dvosmera“ (menjanjem koda,menja se i izgled aplikacije-sinhronizovan)*Java GUI designer*.

*WindowsBuilder* je sastavljen od *SWT Designer-*a i *Swing Designer*-a,sluzi za kreiranje *Java GUI* aplikacija bez da se potrosi puno vremena da se napise kod.

## 2.4. JFrame

Pri radu sa JFrame objektima postoje razlike u odnosu na rad sa njegovom kopijom AWT. Umesto da dodajete kontejnere i komponente direktno u okvir, one se moraju dodati na prelazni kontejner koji se naziva pano sadržaja-content pane.Objekat JFrame je podeljen na nekoliko različitih panoa. Glavni pano sa kojim se radi je content pane, koji predstavlja kompletnu oblast okvira u koji će se komponente dodavati.

## 2.5. JDialog

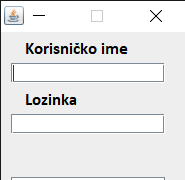
Pop-up prozor kao privremeni prozor za primanje dodatnih informacija od korisnika ili radi obaveštavanja korisnika.

# 3. Postupak kreiranja aplikacije

Kreira se Java projekat i naprave se dva paketa. Paket dostavaApplication.views u kome se nalaze klase, paket dostavaApplication.recources u kome se nalaze resursi kao sto su slike koje se koriste unutar aplikacije.

## 3.1. Prozor narucivanjeHrane.java

Prozor je J*Frame* i u njemu se nalaze tri labele jedan TextFIeld ,jedan PasswordField u kome se unosi lozinka. Postoji i JButton koji omogucava korisniku da se uloguje ako je uneo ime i lozinku kako treba. Treća labela je slika pozadine.



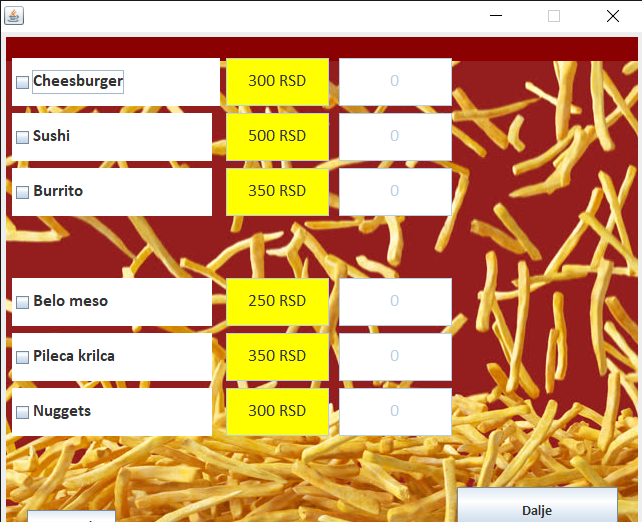
*Slika 1. Izgled prozora narucivanjeHrane.java*

## 3.2. Prozor narucivanjeGlavniP.java

Drugi prozor je glavni. JFrame je i sastoji se od 2 Panel-a.

### 3.2.1. Prvi panel

Ovde(slika 3) je prikazan meni, dugme za zatvaranje aplikazije, dugme *Dalje* za sledeći panel i pozadina u vidu labele. Meni sadrži sedam obroka i za svaki: CheckBox, TextField sa cenom, FormattedTextField za odabir količine i dugme za više informacija. FormattedTextField ograničava moguću količinu obroka.

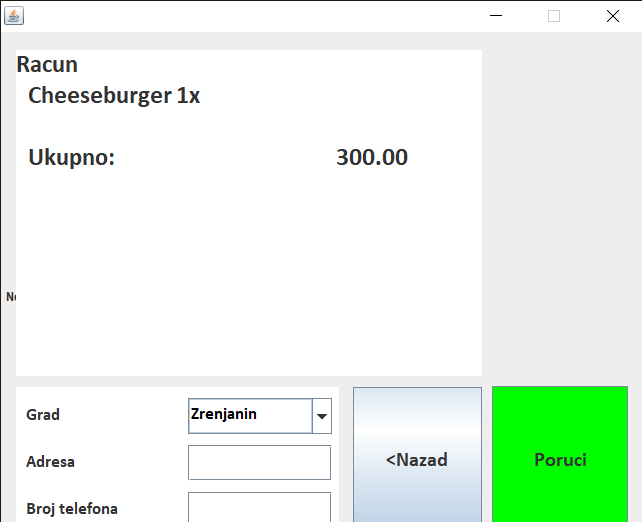


*Slika 3. Izgled prvog panela prozora narucivanjeGlavniP.java*

### 

### 3.2.2. Drugi panel

Na ovoj stranici se nalazi Račun koji je TextArea. Tu se pojavljuju obroci koji su čekirani na prošlom panelu i prikazuje ukupnu cenu porudžbine. Ispod je ComboBox za biranje grada, TextField-ovi za adresu i telefon pored labela koje usmeravaju korisnika. Button *<Nazad* vraća na prvi panel za odabir obroka. Button *Poruci* vodi do JDialog-a za potvrdu.



*Slika 4. Izgled drugog panela prozora narucivanjeGlavniP.java*

# 4. Ključni delovi koda

U prozoru *narucivanjeHrane* nalazi se label sa slikom. Ovo(Listing 1.) je kod za label sa slikom.

|  |
| --- |
| JLabel label = **new** JLabel("");  label.setIcon(**new** ImageIcon(LogIn.**class**.getResource"/slike/9.jpg")));  label.setBounds(0, 0, 185, 182);  frame.getContentPane().add(label); |

*Listng 1. Label sa slikom u prozoru narucivanjeHrane.java*

U prozoru narucivanjeGlavniP.java na prvom panelu se nalazi FormattedTextField koji ograničava unos na od 0 do 9 uz pomoć maske.(Listing 2.)

|  |
| --- |
| MaskFormatter maska = **null**;  **try** {  maska = **new** MaskFormatter("#");  maska.setPlaceholder("0");  } **catch** (ParseException e1) {  e1.printStackTrace();  }    textField1 = **new** JFormattedTextField(maska);  textField1.setToolTipText("Kolicina, najvise 9. ");  textField1.setVisible(**true**); |

*Listng 2. Label sa slikom u prozoru narucivanjeHrane.java*

U prozoru narucivanjeGlavniP.java u prvom panelu dugme *Dalje*(btnDalje) ima dve funkcije. Klikom na njega, pored toga što otvara drugi panel, izračunava ukupnu cenu. Ta cena se prikazuje u drugom panelu.

|  |
| --- |
| JButton btnDalje = **new** JButton("Dalje");  btnDalje.setFont(**new** Font("Calibri", Font.***BOLD***, 14));  btnDalje.addActionListener(**new** ActionListener() {  **public** **void** actionPerformed(ActionEvent e) {  racun.setText("\0");  cena[0]= Double.*parseDouble*(textField1.getText()) \* 300;  cena[1]= Double.*parseDouble*(textField2.getText()) \* 500;  cena[2]= Double.*parseDouble*(textField3.getText()) \* 350;  cena[3]= Double.*parseDouble*(textField5.getText()) \* 350;  cena[4]= Double.*parseDouble*(textField6.getText()) \* 250;  cena[5]= Double.*parseDouble*(textField7.getText()) \* 300;    cena[6]= cena[0]+cena[1]+cena[2]+cena[3];  cena[7]= cena[4]+cena[5];  cena[8]= cena[7]+cena[6];  ukupno =String.*format*("%.2f", cena[8]);    String order = **new** String();  order ="Racun";  **if**(checkChesse.isSelected()) {  order = order +"\n Cheeseburger\t"+textField1.getText()+"x";}    **if**(checkBox2.isSelected()) {  order = order +"\n Sushi \t"+textField2.getText()+"x";}    **if**(checkBox3.isSelected()) {  order = order +"\n Burrito \t\t"+textField3.getText()+"x";}    **if**(checkBox5.isSelected()) {  order = order +"\n Belo meso \t\t"+textField5.getText()+"x";}    **if**(checkBox6.isSelected()) {  order = order +"\n Pileca krilca \t\t"+textField6.getText()+"x";}    **if**(checkBox7.isSelected()) {  order = order +"\n Nuggets \t"+textField7.getText()+"x";}    lokacija.setVisible(**true**);  meni.setVisible(**false**);  racun.append(order + "\n\n Ukupno:\t\t" + ukupno);  } |

*Listing 3. Kod za izračunavanje i prikaz ukupne cene porudžbine*

# 5. Zaključak

Ova aplikacija predstavlja osnovne funkcionalnosti aplikacije za naručivanje hrane. Ne predstavlja izgled gotove, primenljiive aplikacije koje imaju vise funkcionalnosti i lepši interfejs. Aplikacija bi mogla da se usavrši sa konkretnom specifikacijom zahteva, složenijim sistemom za prijavljivanje članova i mogućnosti izmene već dodate porudžbine.

# LITERATURA

1. <https://sr.wikipedia.org/wiki/%D0%95%D0%BA%D0%BB%D0%B8%D0%BF%D1%81>
2. <https://www.ucionica.net/programiranje/uvod-u-java-programski-jezik-116/>
3. <http://poincare.matf.bg.ac.rs/~jelenagr/op/java/javask10.htm>