PROJEKTNA DOKUMENTACIJA

**DOKUMENTACIJA ZA APLIKACIJU „SELLINGDIGITAL“**

**(tema pod brojem 12**:**"Vođenje radnje sa robom široke potrošnje")**

Table of Contents

[I Opšti dio 2](#_Toc14818339)

[Instaliranje Java programskog jezika 2](#_Toc14818340)

[Instaliranje MySql Servera i baze podataka 2](#_Toc14818341)

[ZAHTJEVI PROGRAMA: 3](#_Toc14818342)

[Karakteristike Hardware-a: 3](#_Toc14818343)

[Karakteristike Software-a: 3](#_Toc14818344)

[POKRETANJE PROGRAMA: 3](#_Toc14818345)

[II UPUTSTVO ZA KORISNIKA: 3](#_Toc14818346)

[PRODUCT TYPE (Vrsta proizvoda) 4](#_Toc14818347)

[Dugme Add 4](#_Toc14818348)

[Dugme Edit 4](#_Toc14818349)

[Dugme Delete 4](#_Toc14818350)

[PRODUCT BRAND (marka proizvoda) 5](#_Toc14818351)

[PRODUCT (konkretan proizvod) 5](#_Toc14818352)

[VENDOR (blagajnik/korisnik) 6](#_Toc14818353)

[PAYMENT (plaćanje i račun) 6](#_Toc14818354)

[EXIT (izlaz iz programa) 7](#_Toc14818355)

[III UPUTSTVO ZA DEVELOPERA: 7](#_Toc14818356)

[Tehnički detalji: 7](#_Toc14818357)

[KLASA PRODUCT TYPE: 8](#_Toc14818358)

[Metoda table\_update(): 8](#_Toc14818359)

[KLASA PRODUCT BRAND: 9](#_Toc14818360)

[KLASA PRODUCT: 9](#_Toc14818361)

[KLASA PAYMENT: 10](#_Toc14818362)

[KLASA PRINT 10](#_Toc14818363)

[KLASA LOGIN – ULAZNA TAČKA PROGRAMA 11](#_Toc14818364)

[IV Uputstvo -DEPLOYMENT DOCUMENT- za dio programa koji se odnosi na web-komponentu projekta 11](#_Toc14818365)

[V GENERALNO O Javi i GUI-u 12](#_Toc14818366)

[1. Uvod 12](#_Toc14818367)

[Generalno o javi 13](#_Toc14818368)

[1.2 Objektno-orijentirano programiranje 13](#_Toc14818369)

[Osnove GUI Aplikacija: 14](#_Toc14818370)

[Grafika i slike 16](#_Toc14818371)

[Swing: 16](#_Toc14818372)

[JavaFX: 16](#_Toc14818373)

# I Opšti dio

Aplikacija **Java SELLINGDIGITAL** je dizajnirana da pomogne pravnim licima u vođenju svakodnevnih poslova registracije proizvoda, njihovog razvrstavanja po tipu, brendu i samom proizvodu. Također aplikacija registruje ukupnu dostupnost nekog proizvoda, njegovu dostupnu količinu, prodanu količinu i konačno stanje. Pretraga proizvoda se može vršiti po bar-kodu određenog proizvoda.

Aplikacija automatski množi prodanu količinu sa jediničnom cijenom svakog proizvoda te izračunava ukupnu cijenu za odabrani tip proizvoda i ukupnu cijenu za više izabranih proizvoda ( narudžba). Jednostavnim klikom na dugme „Pay Inovice“ automatski se izračunava kusur za klijenta zatim konačno stanje, datum transakcije i štampa račun.

Svi ključni, gore-navedeni, podaci se čuvaju u bazi podataka.

Aplikacija ne posjeduje autorska ograničenja vezana za sam program.

## Instaliranje Java programskog jezika

Java SELLINGDIGITAL koristi Java programski jezik, Java EE paket, SWING grafički interfejs i MySql bazu podataka.

**Koristena verzija Jave je: Version 8 update 191 (build 1.8.0\_191-b12); Copyright (c) 2018, Oracle.**

Prije korištenja programa obavezna je instalacija **Java Runtime Envirometa (JRE)** samo za pokretanje aplikacije ili **Java Developpment Kita (JDK)** za manipulaciju koda koji zahtjeva **Java Virtualnu Mašinu (JVM)** a koja kompajlira i prevodi napisani kod u binarni kod na način da ga može razumjeti svaki operativni sistem. Podrazumjeva se da minimalna verzija bude gore-navedena (Java 8) ili viša.

Neophodni Java paket može se preuzeti sa ove stranice: <https://www.oracle.com/technetwork/java/javase/downloads/index.html>

## Instaliranje MySql Servera i baze podataka

Također, potrebno je instalirati i MySql Server za bazu podataka koja se može preuzeti sa stranice:

<https://dev.mysql.com/downloads/>

i potrebno je reproducirati strukturu baze podataka za ovaj program, koja se nalazi u paketu sa programom „Java SELLINGDIGITAL“, u fajlu pod imenom: „bazaZaProjekt.sql“. Pravilna instalacija servera i strukture baze podataka je neophodna za rad programa SellingDigital.

## ZAHTJEVI PROGRAMA:

### Karakteristike Hardware-a:

Hardware na kojem je napravljen ovaj program ima slijedeće karakteristike:

|  |  |
| --- | --- |
| Vrsta Hardwarea | Korišteni Hardware |
| Procesor | Intel Core i5 |
| Brzina sata | 2.50 GHz |
| RAM | 4 GB |
| Hard-disk | >10GB |

### Karakteristike Software-a:

Software na kojem je napravljen ovaj program ima slijedeće karakteristike:

|  |  |
| --- | --- |
| Vrsta Software-a | Korišteni Software |
| Operativni sistem | Windows 7 |
| JDK | JDK1.8.0 |
| Driver za bazu podataka | MySql Driver 1.5.47 |
| MySql Server za bazu podataka | Server Version 8.0.13 |

## POKRETANJE PROGRAMA:

Nakon što ste instalirali **JDK ver 8 i MySql** **Server i strukturu baze podataka**, instalacioni program („Instalation file“) će instalirati aplikaciju „SELLINGDIGITAL „ po početnim postavkama u direktorij Program Files/Projekt SellingDigital sa ekstenzijom .jar.

Za pokretanje programa dovoljno je doći mišem na ***Projekt SellingDigital.jar*** fajl te desnim klikom odabrati ***Open with*** i zatim izabrati ***Java platformu***  koja je prethodno instalirana.

Brisanje programa se vrši klikom na ikonu ***Uninstall Projekt SellingDigital*** u istom direktoriju.

# II UPUTSTVO ZA KORISNIKA:

Projekt sellingDigital i MySql baza podataka čine jedinstven sistem Java SELLINGDIGITAL za vođenje radnje sa robom široke potrošnje.

Nakon instalacije neophodnog Java paketa i instalacije MySql baze podatka, program se pokreće

desnim klikom miša na SellingDigital.jar fajl i nakon toga potrebno je izabrati opciju: Open with Java SE.

Nakon toga pojavit će se **prozor Login** koji omogućava identifikaciju korisnika i u koji treba unijeti svoje korisničko ime: Username i svoju lozinku: Password.

Ukoliko su korisničko ime ili lozinka pogrešni pojavit će se obavještenje i mogućnost ponovnog unosa.

Klikom na dugme Cancel , program se prekida.

Ako se korisničko ime i lozinka podudaraju, bit će dozvoljeno dalje korištenje programa Java SELLINGDIGITAL i prikazano sučelje prodaje narudžbe: Payment.

U desnom dijelu sučelja se nalazi izbornik u kojem se nalaze funkcionalnosti:

* Product type
* Product brand
* Product
* Vendor
* Payment
* Exit

Za prvi unos identifikatora korisnika koriste se podaci koji su dati od strane programera i koji se nalaze u bazi podataka (usename: menil, password: menil). Nakon toga moguće je dodavati nove ili brisati postojeće identifikatore korisnika za prozor Login.

## PRODUCT TYPE (Vrsta proizvoda)

### Dugme Add

Klikom na naslov **Product type** u izborniku otvara se prozor putem kojeg se može dodati, izmijeniti ili izbrisati kategorija proizvoda u bazi podataka.

Dodavanje kategorije proizvoda u bazu podataka obavlja se unosom imena kategorije u polje Product type. Nakon toga u izborniku statusa potrebno je odabrati opciju **Active** ako je kategorija postojeća ili **Deactive** ako kategorija nije više postojeća, (u zalihama ili iz drugog razloga).

Klikom na dugme **Add** kategorija proizvoda bit će dodana u sistem i bit će te obavješteni o tome putem poruke „*Product type sucessfully added*“.

Ako su svi koraci uspješno izvedeni nova kategorija će biti prikazana u tabeli na sredini prozora sa svojim pripadajućim podacima:

* **ID kategorije proizvoda**
* **Ime kategorije proizvoda**
* **Status kategorije proizvoda**

### Dugme Edit

Klikom miša na kategoriju proizvoda u tabeli biramo onu kategoriju koju želimo izmjeniti ili izbrisati. Ukoliko želimo izmijeniti podatke o odabranoj kategoriji potrebno je unijeti nove podatke u polja Product type i status i kliknuti na dugme **Edit**. Ako je izmjena uspjela pojavit će se poruka „*Product type sucessfully updated*“. Izmjene će biti vidljive i u tabeli.

### Dugme Delete

Ukoliko želimo izbrisati neku kategoriju, odaberemo kategoriju u tabeli i kliknemo na dugme **Delete.** Program će pitati da li zaista želite izbrisati tu kategoriju. Potvrdom na dugme **Yes** pojavit će se poruka da je kategorija uspješno izbrisna i ta stavka neće više biti prikazana u tabeli.

## PRODUCT BRAND (marka proizvoda)

Klikom na naslov **Product brand** otvara se sučelje za unos, izmjenu, ili brisanje podataka o brendu proizvoda.

Ovo sučelje radi na sličan način kao i sučelje Product type. Unosom podatka o brendu u polje Product brand i odabirom statusa postavljaju se podaci za unos novog brenda određene kategorije proizvoda.

Klikom na dugme **Add** bend će biti dodan, klikom na dugme **Edit** brend će biti izmjenjen, a klikom na **Delete** Product brand će biti izbrisan iz sistema.

Sve izmjene trebaju biti prikazane u tabeli na isti način kao i u sučelju Product type.

## PRODUCT (konkretan proizvod)

Klikom na naslov **Product** otvara se sučelje za unos podataka za sam proizvod koji je objekt prodaje. Sučelje Product ima više elemenata:

* polje Product - u koji se unosi ime samog proizvoda
* polje Description – u koje se upisuju detalji i opis proizvoda
* izbornik Product type - u kojem se bira kategorija kojoj proizvod pripada
* izbornik Product brand – u kojem se bira brend kojem proizvod pripada
* polje Cost price – polje u koje se upisuje nabavna cijena
* polje Retail price – u koje se upisuje prodajna cijena
* polje Quantity – u koje se upisuje nabavljena količina proizvoda
* polje Barcode – u koje se unosi jedinstveni barkod proizvoda
* izbornik Statusa- u kojem se bira dostupnost proizvoda: **Active** za dostupan, **Deactive** za nedostupan

Nakon što smo unijeli sve neophodne podatke za novi proizvod, klikom na dugme Add proizvod će biti upisan u sistem i pojavit će se u tabeli na dnu sučelja sa svim svojim karakteristikama.

Ako želimo izmjeniti podatke o upisanom proizvodu, kliknemo mišem na željeni proizvod u tabeli i u referentna polja unesemo nove podatke, a u izbornicima (drop-list) odaberemo željenu kategoriju i kliknemo na dugme Edit.

Promjene u izbornicima Product type i Product brand neće bit automatski preuzete iz tabele te ih je potrebno ručno odabrati svaki put. Ako se ručno ne odaberu opcije iz izbornika Product type i Product brand, klikom na dugme Edit neće se dogoditi ništa oism prvi put kada uđemo u sučelje Product.

Dakle nakon prvog klika na Edit treba ručno odabrati Product type i Product brand i tek onda opet kliknuti na Edit.

Ako želimo izbrisati neki proizvod iz sistema kliknemo na njega u tabeli i pritisnemo mišem dugme Delete.

Sve promjene statusa Proizvoda bilo da je riječ o dodavanju, izmjeni podataka ili brisanju proizvoda bit će registrirane u tabeli i o njima će korisnik biti obaviješten putem poruke u pop-up prozoru.

## VENDOR (blagajnik/korisnik)

Klikom na naslov **Vendor** bit ćete preusmjereni na sučelje za unos podataka o blagajniku.

Ovi podaci su neophodni za pristup sučelju prodaje-Payment i prvo pokretanje programa jer sadrže korisničko ime i lozinku blagajnika.

U polju username treba upisati novo korisničko ime a u polju password treba upisati novu lozinku koja će se koristiti sa odabranim korisničkim imenom u paru.

Nakon toga potrebno je izabrati opciju statusa blagajnika **Active** ako je u radnom odnosu ili **Deactive** ako iz nekog razloga nije u radnom odnosu.

Klikom na dugme **Add** nove korisničke informacije o imenu i lozinki bit će unijete i spašene u sistemu. U tabeli će biti ispisana samo postojeća korisnička imena u koloni username dok podaci o lozinki neće biti prikazani radi sigurnosti.

Odabirom blagajnika u tabeli i klikom miša na njegovo ime otvara se mogućnost izmjene podataka za tog korisnika. Nakon što izmijenite podatke potrebno je kliknuti na dugme **Edit** i novi podaci će biti spašeni u sistemu te prikazani u tabeli.

Na isti način se bira blagajnik kojeg želimo brisati iz sistema, samo je nakog odabira potrebno kliknuti na dugme **Delete** i blagajnik više neće biti prikazan ni u tabeli.

Izbrisani blagajnik više neće biti u mogućnosti pristupiti sistemu jer će njegove informacije potrebne za pristup biti uklonjene.

## PAYMENT (plaćanje i račun)

Payment je glavno (iako nije entry-point) sučelje sistema Java SellingDigital i za pristup ovom sučelju uvijek je potrebno uraditi log-in, bilo da se program pokreče prvi put ili da se pristupa iz nekog drugog sučelja!

Ovo sučelje obavlja funkciju same prodaje određenog proizvoda i izdavanja računa. Ima više funkcionalnsti:

* **polje product code** – upisuje se barkod proizvoda i potvrđuje se klikom na Enter

Ukoliko se unese pogrešan ili nepostojeći barkod, korisnik će biti o tome obaviješten te je potrebno ponovo unijeti ispravan barkod i pritisnuti tipku Enter.

* **polje product name** – ispisuje ime proizvoda iz sistema koje odgovara traženom barkodu
* **polje price** – ispisuje prodajnu cijenu traženog proizvoda u odnosu na barkod
* **polje quantity** – u ovo polje potrebno je upisati željenu količinu za prodaju. Ukoliko tražena količina proizvoda nije dostupna, korisnik će biti obaviješten o tome kao i o dostupnoj količini te je potrebno da odabere onu količinu koja je manja od maksimalne dostupne količine.
* **polje subtotal** – vraća ukupnu cijenu cijele transakcije za sve proizvode
* **polje payed** - u ovo polje treba unijeti koliko je kupac platio
* **polje balance** – ispisuje kusur nakon klika na dugme Pay Invoice

Total, Subtotal, Payed

Nakon što se unesu potrebni podaci (barkod proizvoda i željena količina) potrebno je kliknuti mišem na dugme Add. Program će zatim automatski pomnožiti jediničnu cijenu proizvoda sa traženom količinom i izračunati ukupnu cijenu po proizvodu. Svi podaci će biti upisani u tabelu na sredini sučelja a ukupna cijena će biti prikazana u tabeli pod kolonom **Total**.

Ukoliko kupac želi drugi proizvod treba ponoviti postupak i opet kliknuti na dugme Add. Novi proizvod će biti dodan u tabelu ispod predhodnog.

Ako je došlo do greške, potrebno je odabrati proizvod u tabeli i kliknuti na dugme Delete. Proizvod i količina će biti izbrisani.

Ukupna cijena iz kolone Total će biti sabrana za svaki proizvod iz tabele i upisana u polje **Subtotal**.

Nakon primitka novca potrebno je ukucati primljenu sumu u polje **Payed**.

Nakon ove faze kada je prodaja odabranih proizvoda i količine zaključena treba pritisnuti dugme **Pay Invoice**. Klikom na ovo dugme program automatski računa kusur(**Balance**) koji je potrebno vratiti klijentu te ispisuje račun sa potrebnim podacima za klijenta.

Ako ne želimo isprintati račun u prozoru **Print** kliknemo na **Cancel**.

Program će nas obavijestiti da je ovaljena transakcija snimljena u bazu podataka. Također u sistem se upisuje i **username prodava**ča i **datum** kada je transakcija obavljena te se time omogućava statistika efikasnosti pojedinačnih prodavača u firmi.

Zatvaranjem prozora računa, program se prekida. Klikom na dugme **OK** u dnu prozora računa, program nastavlja sa radom.

## EXIT (izlaz iz programa)

Ako iz nekog razloga želimo prije krajnje faze prodaje izaći iz programa kliknemo na naslov **Exit** i program se prekida.

# III UPUTSTVO ZA DEVELOPERA:

## Tehnički detalji:

Java SELLINGDIGITAL je java aplikacija pisana u JDK 8 update 1.9.1.

Korišteno radno okruženje je NetBeans IDE 8.2.

MySql baza podataka - Server version 8.0.13

Pristup bazi SellingDigital (baza supermarket) se vrši preko MySql JDBC konektora ver 1.5.47 koji se mora kao .jar fajl dodati u datoteku Libraries u kojoj je i defaultni JDK8.

Instrukcije: U paketu SellingDigital nalaze se sve klase koje ovaj program koristi i to imenom:

* Product type
* Product brand
* Product
* Payment
* Vendor
* Print
* Login

Klasa Login je ulazna tačka (entry-point) u program.

## KLASA PRODUCT TYPE:

U ovoj klasi korištene su klase iz biblioteka java.sql, java.util i java.swing.

Klasa Product type nasljeđuje klasu java.swing.JFrame. U konstruktoru se pozivaju metoda za inicijalizaciju komponenti Swinga **initComponents()** i metoda za ažuriranje prikaza tabele **table\_update().**

Instanciran je objekat klase Connection- **con1,** preko kojega se povezuje sa bazom podataka: **supermarket**.

Instanciran je objekat klase PreparedStatment – **pst,** preko kojega se ostvaruje SQL querry.

### Metoda table\_update():

Instancira varijablu **c** koja prima broj kolona iz tabele.

Zatim se u try-catch bloku instancira MySql Driver i on ostvaruje konekciju na bazu: **supermarket**, izvršava se querry te preuzete podatke spašava u objekat klase **ResultSet – rs**.

Instanca klase **ResultSetMetaData –rsmd** preuzima podatke o strukturi tabele (tj. kolone) preko objekta rs. Zatim se broj kolona tabele preko rsmd objekta pohranjuje u varijablu c.

Instancira se objekat d klase **DefaultTableModel** koji preuzima virtualni model tabele nacrtane u Swingu. Nad tim objektom se postavlja brojač redova tabele na nultu poziciju.

U while-petlji postavlja se uvijet da postoji slijedeći rekord (upis) u bazi podataka tj. petlja će se zaustaviti kada dođe do kraja. Ako je on ispunjen, u for-petlji se preuzimaju podaci iz kolona id, product type i status, te se spašavaju u objekt **v2 klase Vector** koji implementira **List<E>(kolekcije objekata).**

Kroz **d** objekt koji drži virtualni model tabele, preuzimaju se podaci iz v2 i ispisuju u redove tabele za potrebe korisnika.

Metoda **table\_update()** se poziva odmah na početku da bi korisnik imao prikaz već postojećih podataka iz tabele category u bazi.

Zatim su dodane funkcije dugmadima **Add, Edit i Delete**. Ovo se izvršava putem **Eventa i Listenera** koji trigeruju događaj klika na dugme. Event handler dalje izvršava različite querryje prema tabeli brand\_table. Add dugme preuzima upisane podatke iz polja product type i izbornika statusa kategorije, dodaje podatak u bazu preko SQL-a te postavlja vrijednosti polja i izbornika na početne postavke za eventualni drugi korisnički upis.

Zatim se nakon dodavanja kategorije, poziva metoda koja obrađuje događaj klika mišem na stavku u tabeli. Ona preuzima preko modela tabele podatke i šalje ih u njihova relativna polja. Time omogućava izmjene podataka u samoj tabeli. Ovo se radi preko **ID-a kategorije** **(getValueAt(selectedIndex,1))**-što odgovara prvoj koloni koja čuva zapise **ID-a** svih kategorija.

Klikom na dugme Edit izvršava se querry koji ažurira podatke u tabeli za odabranu kategoriju.

Klikom na dugme Delete brišu se podaci za onu kategorju čiji ID odgovara ID-ju koji je izabran u tabeli od strane korisnika.

Nakon svakog izvršavanja zahtjeva, preko **JOptionPane**-a šalje se poruka korisniku o izvršenom rezultatu komande Add,Edit ili Delete.

Zatim se omogučuje da korisnik klikom na naslov(label) u izborniku odabere neko drugo sučelje tako što se instancira objekat klase odabranog sučelja i omogučuje se njegova vidljivost (setVisible(true)).

Main (String args[]){} je početna tačka izvršavanja svakog Java programa.

Na samom kraju nalazi se main() metoda koja izvršava cijeli kod tako što poziva AWT metodu invokeLater(new Runnable) za čiji se argument šalje new Runnable = instanca nove niti/threada a koja preko metode run() instancira cijelu klasu sa pripadajučim elementima i prikazuje je na ekranu.

Ovo se radi na samome kraju zato da bi svi elementi, objekti i metode klase potrebni za njeno funkcioniranje bili izvršeni u momentu prikaza slike korisniku. U suprotnom mogli bi doći u situaciju da je element prikazan na ekranu ali da njegov objekt nije još uvijek obrađen (tj.kompajliran) što bi dovelo do greške. Swing ovo automatski spriječava, a svojim elementima postavlja kvalifikator private da se nebi mogli mijenjati. Ako bismo željeli prilagoditi elemente Swinga morali bi napraviti svoju klasu koja nasljeđuje neku klasu elemnta Swing-a te joj dodati nove funkcionalnosti.

## KLASA PRODUCT BRAND:

Sadrži iste elemente i funkcionalnosti kao i klasa Product type samo što se odnosi na Product brand i podaci se upisuju u bazu supermarket.brand\_table.

## KLASA PRODUCT:

Klasa Product je malo složenija od predhodne dvije. Ona u sebi sadrži dvije pomoćne klase (helper-class): a) **Product typeItem i b) Product brandItem**. One u svom konstruktoru preuzimaju podatke o ID-u i imenu kategorije i brenda iz baze podataka i pohranjuju ih u **varijable id i name**. Naime, metode **Product type() i Product brand()** preuzimaju ID i ime iz podrazumjevanih tabela u bazi podataka i proslijeđuju ih u konstruktor klasa Product typeItem i Product brandItema.

Ovo je potrebno da bi se combo-box , tj. izbornik sučelja Product, mogao napuniti novim kategorijama i brendovima ako su dodane ili ih ne prikazivati ukoliko su bile izbrisane u međuvremenu.

U ovoj klasi **metoda table\_update()** koristi praktično SQL querry funkcionalnost da bi preko **aliasa** tri tabele (category=c, product=p i brand\_table=b) vratila tzv. **Pogled (View)** iz tri tabele u jednu jedinstvenu tabelu, ispisanu na ekranu.

Nakon toga se ažurira tabela na isti način koji je opisan u prethodnim klasama. Funkcionalnosti dugmadi Add, Edit i Delete su praktično iste samo je razika u preuzetim podacima iz Swing elemenata.

## KLASA PAYMENT:

Ovo je klasa koja izvršava matematičke operacije množenja prodajne cijene sa količinom i upisuje ih u polje Total, operacije sabiranja totala svakog proizvoda,operacije plaćanja, izdavanja računa, upisa u bazu podataka o transakciji, blagajniku, iznosu i datumu.

Također kroz ovu klasu se bira traženi proizvod. Proizvod se odabere upisom barkoda proizvoda u polje **product code**. Nakon toga odabir koda treba potvrditi tipkom **Enter.** Nakon što je korisnik pritisnuo Enter if-petljom se ispituje da li uneseni barkod postoji ili ne. Ukoliko ne postoji program javlja da barkod nije pronađen. U slučaju kada upisani barkod postoji unesena vrijednost se poredi sa vrijednosti u bazi podataka te se preko njega dobiju podaci o imenu proizvoda i prodajnoj cijeni i oni se ispisuju na ekran.

Nakon toga potrebno je unijeti traženu količinu u **polje quantity** i kliknuti na **dugme Add**.

Ovo dugme poziva metodu **Payment().**

**Metoda Payment()** preko barkoda traži podatke o proizvodu, te množi količinu sa cijenom i računa total. If-petljom se provjerava količina proizvoda. Kada je tražena količina veća od dostupne o dostupnoj količini nekog proizvoda se obavještava korisnik. Moguće je upisati manju količinu od dostupne.

Ako je tražena količina dostupna, program upisuje sve podatke o prodaji koji se odnose na pojedinačni proizvod u tabelu.

Zatim se sabiraju totali prodaje, za svaki proizvod u tabeli, i njihov zbir je ispisan u polje Subtotal.

U slučaju da je došlo do greške operatera, selektuje se proizvod u tabeli i klikom na dugme Delete taj se proizvod briše iz tabele a ukupna suma se ponovno preračunava i upisuje u polje Subtotal.

Nakon toga korisnik treba unijeti iznos uplate u **polje Payed**.

Dugme **Pay Invoice** računa kusur tako što od varjable Subtotal oduzme vrijednost varijable Payed i rezultat spasi u varijablu Balance te je ispiše na ekran.

Također, klik na ovo dugme poziva i druge dvije metode: **sales() i print().**

**Metoda sales()** upisuje podatke o datumu i blagajniku koji je izvršio transakciju, ukupnoj sumi, plaćenom iznosu i kusuru u **tabelu sales**. Zatim u **tabelu sales\_product** upisuje podatke o svakoj obavljenoj transakciji tj. izdanom računu.

Na kraju **metoda print()** ispisuje, u elementu **JTextArea**, sve potrebne podatke na računu i poziva prozor za podešavanje konekcije na printer za ispis računa.

Ako ne želimo isprintati račun klikom na Cancel prozor se ukida a ostaje obavijest o upisu u bazu i virtualni izgled računa na ekranu. Klikom na dugme OK, u dnu računa, ponovno se instancira sučelje Product a program nastavlja sa radom.

Zatvaranjem prozora virtualnog računa prekida se i sam program SellingDigital.

## KLASA PRINT

Ova klasa kroz svoj konstruktor vraća neophodne podatke računa i ispisuje ih. Ispis se radi manipulacijom stringova i vrijednosti argumenata koji su proslijeđeni u konstruktor. Argumenti za konstruktor klase Print se proslijeđuju iz klase Payment kroz metodu print().

Klikom na dugme OK se vračamo na sučelje Payment, kao što je spomenuto iznad.

## KLASA LOGIN – ULAZNA TAČKA PROGRAMA

Klasa početnog sučelja programa ima funkciju provjere identiteta korisnika programa.

Ona iz **polja username i password** preuzima upisani String i spašava ga u varijable. Te varijable proslijeđuje u SQL querriju u bazu podataka i na osnovu unjetoga traži podatke o korisniku.

Ako se vrijednosti varijabli username i password podudaraju sa onima iz **tabele Login**, bit će instanciran objekat klase sučelja Payment. Drugim riječima, korisniku će biti omogućen rad u SellingDigital programu.

Ukoliko se te vrijednosti ne podudaraju korisnik će biti obaviješten o grešci prilikom upisa.

Sve **varijable tipa Double** koje su korištene u programu su zaokružene pomoću biblioteke **Math.round()** radi ispisa na ekran od dva decimalna mjesta.

# IV Uputstvo -DEPLOYMENT DOCUMENT- za dio programa koji se odnosi na web-komponentu projekta

U direktoriju **Projekt ver 1** nalazi se i datoteka **Web**/**Web4**/**WebContent** koja sadrži fajlove neophodne za Web-aplikaciju. Dio je pisan HTML-Servlet načinom a dio u JSP.

To su fajlovi:

* **FormaKorpa.jsp**
* **IndexJSP.css**
* **Index.html**
* **IndexHTML.css**
* **Pozadina.jpg**

Dio koji se odnosi na servlete nalazi se u **datoteci src/gorcin** i sadrži fajlove:

* **Kon.java**
* **Kvadrat.java**
* **Login.java**

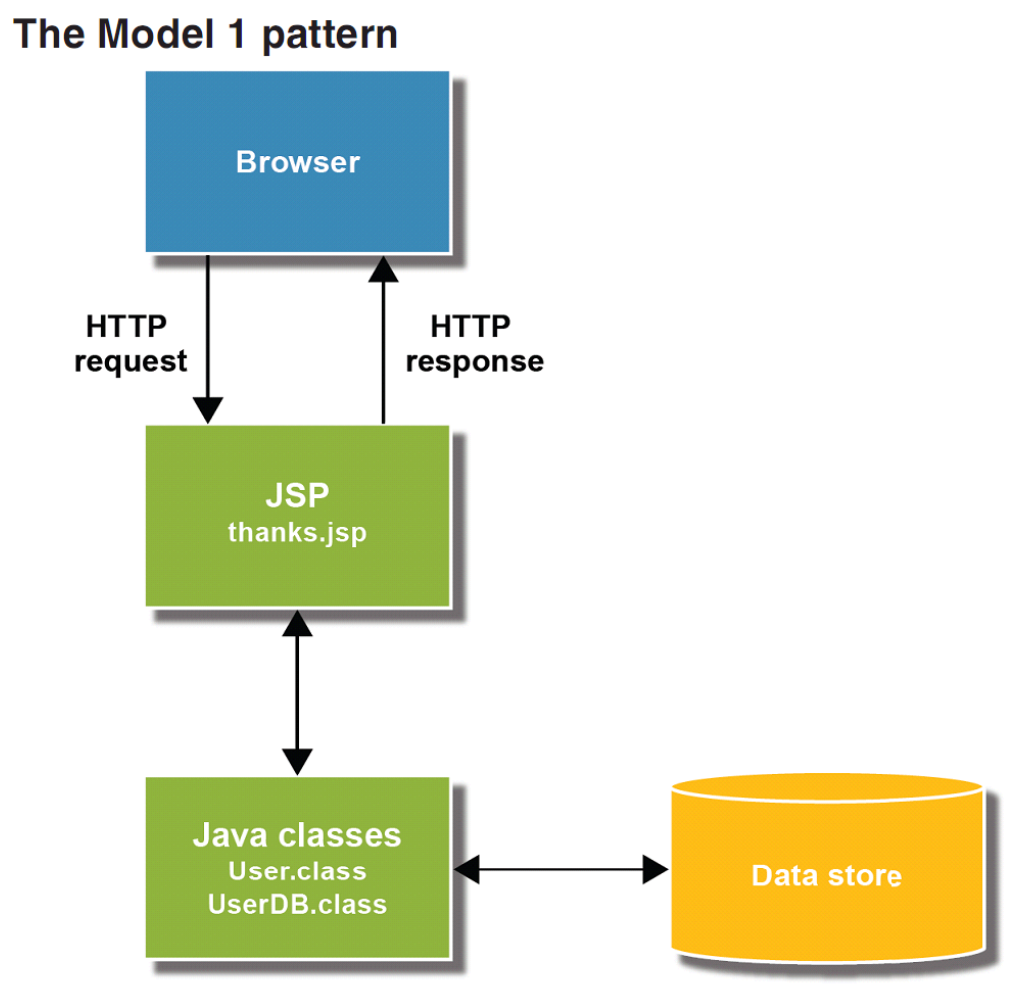
Ovaj dio aplikacije je zamišljen kao Web-aplikacija povezana na bazu podataka koju koristi desktop aplikacija SellingDigital. Preko Index.html stranice, moguće je unjeti user-name i password direktno u bazu koristeći HTML formu za slanje podataka servletu **Login.java** koji ih obrađuje i komunicira sa bazom preko klase **Kon.java.**

Klasa Kon.java ima zadatak da ostvaruje vezu sa samom MySql bazom preko JDBC Drivera.

Servleti su krajnje tačke nekog Java web-programa i postoji najmanje jedan sa klijentske strane i jedan sa serverske strane.

**FormaKorpa.jsp** koristi tehnologiju **Java Server Pages – jsp.** Ona omogućava simultano pisanje dijela koda koji se odnosi na samu strukturu web-stranice i koda koji omogućava interakciju **Klijent-Program-Server-DB.**

**Slika 2**



U Java Server Pages tehnolgiji operacije servleta su podrazumjevane, a dokument je pisan naizmjenično: u Hyper-text-mark-up-laguage –u (HTML) i Java programskom jeziku. U izradi web-stranice, koršten je, mada vrlo malo, i Cascade Style Sheet(CSS).

HTML5 je vrsta XML-a prilagođena HTTP protokolu (jedan od protokola prenosa podataka putem interneta). Uloga HTML-a je struktuiranje web-stranice putem elemenata na način koji je lak za čitanje Browseru (pretraživaču). CSS ima zadatak da tu strukturu stilizira. Međutim ova dva opisna jezika (HTML i CSS) mogu realizirati samo web-stranice statičnog tipa bez integrisanih funkcija opisanih u Slici 2.

Da bi web-stranica postala dinamična ( reactive web-page) potrebno je koristiti i neki ozbiljniji programski jezik za relaciju: server – aplikativni server- program/aplikacija- klijent.

U te svrhe se mogu koristiti i Java i JavaScript, dva različita programska jezika, čija je razlika (između ostalih) i u tome što se Java izvršava na serverskoj strani a JavaScript na klijentskoj tj. na samom browseru.

Komunkacija se vrši najčešće putem spomenutog HTTP protokola, predefinisanim metodama GET, POST, PUT, DELETE itd. Ovo ostvaruju i sami servleti koji imaju integrisane metode doGet(), doPost() service()(sadrži obje prethodne). Podaci se mogu čuvati u objektu sesije –Session a svaki HTTP upit tretira se kao nov.

Nažalost, u ovom projektu preko fajla FormaKorpa.jsp, podaci o proizvodima iz SellingDigital se mogu samo ispisati u HTML tabelu. Za programiranje njihovog odabira, pohranjivanje u Kolekciju, držanje u Sesiji i ispis Korpe odabranih proizvoda klijenta na web-stranici nisam imao dovoljno vremena da završim jer mi je fokus bio na funkcionalnosti desktop programa.

Web-aplikaciju je potrebno postaviti na neki server (u GlassFish-u je to folder: deployable).

U izradi web-dijela projekta korišteni su serveri: Tomcat i Glassfish.

Korištena su okruženja: Netbeans 8.2, Eclipse 2019-03, i nešto manje Intelijj.

# V GENERALNO O Javi i GUI-u

## 1. Uvod

### Generalno o javi

Općenito Java je konkurentni, objektno-orjentirani programski jezik, opće namjene koji se zasniva na klasama i objektima i koji je dizajniran da ima što manje ovisnosti i prepreka prilikom implementacije.

Namjena je bila dopustiti razvojnim programerima "pisanje jednom, pokretanje bilo gdje" (Write Once Run Anywhere), što znači da kod koji se pokreće na jednoj platformi ne treba ponovno kompajlirati da bi se izvodio na drugoj. Java aplikacije se obično sastavljaju u bajtni kod (.class fajlovi) koji se može izvoditi na bilo kojoj Java virtualnoj mašini (JVM) bez obzira na arhitekturu kompjutera.

Rad Java programa:

Java kompajler kompajlira kod koji prije kompajliranja ima ekstenziju .java a nakon kompajliranja .class. Zatim Java Virtuelna Mašina (JVM) prevodi ulazni,kompajlirani kod, u izlazni,binarni (tj. niz 0 i 1) te šalje Operativnom sistemu.

Upravo ovo omogučuje Java programima da budu neovisni od platforme i OS-a na kojem se pokreću što je i bio osnovni zadatak Java programskog jezika.



Java je od 2012. godine jedan od najpopularnijih programskih jezika u upotrebi, sa prijavljenih preko 10 miliona korisnika . Javu je izvorno razvio James Gosling iz firme Sun Microsystems (koja se zatim spojila s Oracle Corporation). Jezik nasljeđuje puno svoje sintakse iz C i C ++, ali je i jednostavniji. Java se može koristiti za pisanje desktop, web ili mobilnih aplikacija. Java aplikacija je slična bilo kojem drugom jeziku visokog nivoa: može se samo kompajlirati, a zatim pokrenuti na istom računaru. Java dolazi s velikom bibliotekom gotovih klasa i objekata. Od Jave Standard edition 1.2. postoji biblioteka za rukovanje korisničkim sučeljima (u daljem tekstu GUI) koja se zove Swing.

Od 2005 Swing nasljeđuje jedan novi GUI framework koji se zove JavaFX i koji ima nekoliko prednosti jer je lakši za upravljanje, ima kontroler klasu, CSS stiliziranje, koristi SceneBuilder alat i pogodan je za web-aplikacije.

U ovom projektu je korišten SWING umjesto JavaFX grafičkog sučelja zbog sporosti učitavanja komponenti Java-FX preko plug-ina u NetBeans razvojnom okruženju.

### 1.2 Objektno-orijentirano programiranje

Java podržava tehnike objektno-orijentiranog programiranja koje se temelje na hijerarhiji klasa i dobro definiranim objektima koji surađuju međusobno.

#### Klase:

Klasa je struktura koja definira podatke i metode (funkcije) za rad nad tim podacima. Klasa je nešto slično obrascu po kojem se generira objekat te vrste.

Kada pišemo programe u Javi, svi programski podaci su obmotani u klasu, bilo da se radi o klasi koju smo mi kreairali ili o klasi koja dolazi iz Java API biblioteka.

Klase u Java API bibliotekama definiraju skup objekata koji dijele zajedničku strukturu i ponašanje.

#### Objekti:

Instanca je sinonim za objekat klase. Novostvorena instanca ima podatke i metode koje je klasa definirala za tu instancu.

Objekat ima jedinstvenu lokaciju u memoriji i predstavlja instancu te klase.

Definicija klasa mora omogućiti objektima međusobnu komunikaciju prilikom izvršavanja programa.

#### Nasljeđivanje i polimorfizam:

Koncept koji omogučuje suradnju među objektima je nasljeđivanje. Nasljeđivanje definira odnose među klasama u objektno-orijentiranom jeziku. Odnos je takav da objekt-dijete naslijeđuje sve atribute i metode od objekta klase-roditelja sa mogučnosti da ih proširi za vlastite potrebe, uz određena ograničenja.

U Javi se sve klase izvode iz java.lang.Objecta i nasljeđuju njegove metode.

**Slika 1** prikazuje hijerarhiju klasa koje se izvode iz java.lang.Object-a za klase u primjeru korisničkog sučelja ovog projekta.

Java.lang.Object

JFrame implementira Window:

JButton, JLabel, JTable,JtextField

(SWING Kontrole)

JPanel

(SWING Kontejner)

### Osnove GUI Aplikacija:

**U Javi su se za stvaranje grafičkog sučelja koristila tri Frameworka:**

* **AWT – Abstract Window Toolkit**
* **Swing – Dio Java Foundation klasa –API za grafiku**
* **JavaFX – CSS/FXML Framework**

Mnoge funkcionalnosti uz određena unapređenja naslijeđene su iz AWT-a u Swing i zatim u JavuFX.

GUI programi nude bogato sučelje sa nizom funkcionalnosti kojima korisnik pristupa upotrebom miša, tipkovnice i drugih inputa za interakciju sa GUI komponentama kao što su dugmad (buttons), polja za čekiranje (check-fields), polja za tekst (text-fileds), tabele itd.

U ovome projektu korišteno je sedam sučelja:

* Product type
* Product brend
* Product
* Vendor
* Login
* Print
* Payment

#### JFrame

U Java GUI programu svaka komponenta interfejsa (sučelja) predstavljena je posebnim objektom u kodu. Jedna od osnovnih komponenti svakog sučelja je Window. Njegove dimenzije se mogu mijenjati, može imati naslov, može biti prikazan na ekranu ali najvažnija karakteristika je što u sebi sadrži druge elemente. Postoje razni tipovi Windowsa ali u ovom projektu je za to korištena **JFrame** klasa koja dolazi u paketu javax.swing.

#### JPanel

JPanel je druga fundamentalna klasa u Swingu. To je prazan pravokutnik ali može biti koristan na 2 načina. Ili da se na njemu nešto crta ili da se na njega stavljaju (kače) druge komponente, a što je slučaj u ovom projektu.

#### Komponente i Layout (raspored)

Komponente moraju zauzimati svoju određenu poziciju na sučelju a za njihov raspored zadužen je Layout. Postoji više vrsta Layouta (Grid, Box itd.) čije se komponente mogu programirati ali radna okruženja (npr. Scene builder) dozvoljavanju namještanje komponenti na Stage pomjeranjem miša te time znatno olakšavaju rad programera.

#### Događaji i Osluškivaći (Events & Listeners)

Iako struktura komponenti GUI-ja treba lijepo da izgleda ona sama ne daje funkcionalnost programu.

Šta korisnik može raditi sa programom i kako će program odgovoriti?

Program se uglavnim vodi događajima tj. akcijama koje korisnik poduzima i koje za posljedicu imaju okidanje događaja (event trigger) kao što su npr. klik miša, pritisak tipke Enter itd. Kada je događaj okinut, program odgovara pozivom metoda koja trebaju tretirati taj događaj (event-handling). Najčešća tehnika tretiranja događaja je korištenje osluškivaća (listener). Osluškivać je objekt sa određenim metodama i kada je događaj okinut od strane nekog drugog objekta recimo Button-a, osluškivač trigeruje predefinisane metode koje trebaju obraditi taj događaj. Sam događaj je treći objekt koji u sebi nosi informaciju.

Ta podjela odgovornosti pojednostavljuje programiranje.

## Grafika i slike

Fizička struktura GUI-a je izgrađena od komponenti. Pojam komponenta odnosi se na vizualni element u grafičkom korisničkom sučelju, uključujući dugmad, izbornike, polja za unos teksta, trake za pomicanje i tako dalje. U Javi, GUI komponente su predstavljene objektima koji pripadaju podklasama klase java.awt.Component.

U grafičkim sistemina alati sa prozorima (Windowing toolkit npr. JFrame na koji se postavi JPanel u Swingu ili Stage na koji se postavlja Scene sa Layoutom u JavaFx-u) su odgovorni za osigravanje okvira programa da bi omogućili renderovanje potrebnih bita u određeno vrijeme.

Da bi smo koristili grafiku u Java programima postoji niz biblioteka koje je potrebno importovati.

Neke od važnijih korištenih u ovom programu su slijedeće:

### Swing:

**javax.swing.JFrame** - inicijalizira sve komponente metodom - initComponents(). Svaka klasa u programu nasljeđuje ovu metodu iz javax.swing.JFrame biblioteke. Ona također pruža funkcionalnosti prozora (Window frame).

**javax.swing.JPanel** - je kontejner koji sadrži pripadajuće komponente.

**javax.swing.JLabel** – nasljeđuje JComponent i predstavlja područje u koje se može upisati neki tekst ili postaviti slika.

**javax.swing.JButton** – dugmad mogu biti konfigurirana i kontrolirana akcijama ( npr. klik, dupli klik itd.). Nasljeđuje klasu ApstractButton.

**javax.swing.JTable** – ovo je tabela dizajnirana za dvodimenzionalni prikaz podataka po ćelijama.

**javax.swing.JTextFiled** – polje za jednodimenzinalni unos teksta.

**javax.swing.JText Area** – polje za unos teksta u više redova.

**javax.swing.JOptionPane** – ima pop-up funkciju koja se trigeruje na događaj i pogodan je za prikaz poruke programa pozivom **showMessageDialog** metode ili potvrdu neke akcije pozivom metode **showConfirmationDialog.**

### JavaFX:

U JavaFx grafičkom frameworku pandan JFramu kao temeljni element nalazimo **Stage objekt** koji pruža funkcionalnost prozora. Na njega se postavlja scena –S**cene objekat**, koji nosi komponente što bi odgovaralo JPanu u Swingu.

Da bi cijeli komponente i grafika bile vidljive nad Scene objektom potrebno je pozvati metodu **setVisible()** i postaviti logički parametar:

Scene scene1 = new Scene() -> Scene1.setVisible(true).

Na scenu treba postaviti i Layout koji raspoređuje komponente po predefinisanim poljima.

Ovo se ostvaruje pomoću **Layout Managera**, koji pružaju neke AWT ili Swing klase.

Layouta ima više vrsta npr: BoxLayout, BorderLayout, GridLayout itd...

*Napomena:*

*Tokom izrade projekta nailazio sam na probleme vezane za Hibernate tools (ostvarivanje perzistencije kroz ORM alat), probleme lociranja i prepoznavanja Servera GlassFish i probleme sa samim JDBC konektorom koji se tiču različite vremenske zone Java SE i MySQL dijela aplikacije*.

Rad je pisan za završni projekt, IT-Academy Sarajevo, šk. 2018-2019,smijer: Java programiranje

Rad pripremio: Gorčin Stojanović

Sarajevo, 23.7.2019