

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 19 Bratislava 4

Objektovo orientované programovanie

Aukcia koní

Filip Remšík

Obsah

Zámer.....	2
Fungovanie aplikácie	3
Účastník	3
Administrátor	3
Majiteľ	3
Aukcia	3
Hlavné kritériá:	3
Hierarchia tried koní.....	3
Dedenie	3
Polymorfizmus.....	3
Hierarchia tried používateľov.....	4
Dedenie	4
Polymorfizmus.....	4
Zapúzdrenie	4
Agregácia	4
Kód organizovaný do balíkov.....	4
Ďalšie kritériá:.....	4
Použitá vlastná výnimka	4
Serializácia	5
Viacnitosť.....	5
Oddelenie GUI od aplikačnej logiky.....	6
Použitie lambdy v aplikačnej logike.....	6
RTTI.....	6
Diagram	7

Zámer

Môj projekt je zameraný na aukcie koní. Na začiatku sa užívateľ prihlási do aplikácie. Obyčajní účastníci majú možnosť sa len zúčastňovať na dražbách. Vyberú si niektorú z aktuálne prebiehajúcich dražieb a pošlú svoju ponuku. Administrátori môžu vytvárať nové aukcie, poprípade ich upravovať, ale nebudú mať možnosť sa na nich zúčastňovať. Majiteľ má ešte možnosť rozhodovať o administrátoroch. Pri vytváraní aukcií sa vyberie, na čo sa daný kôň využíva (či je to športový typ, či slúži na prácu, výstavný kus..), následne sa vyplnia všetky potrebné údaje o koňovi a na záver informácie o samotnej dražbe. Typ aukcie bude Vickreyho aukcia. Každý účastník odovzdá svoju ponuku. Vyhrá účastník s najvyššou ponukou, pričom ale zaplatí cenu odovzdanú v druhej najvyššej

ponuke. Pri koňovi bude uvedená aj minimálna cena, ktorú môže uchádzač ponúknuť. Aukcia bude ukončená buď po uplynutí časového limitu alebo môže byť zmazaná počas jej priebehu.

Fungovanie aplikácie

Počet hlavných verzií: 5

Účastník

Môže do každej aukcie prihodiť jednu sumu, keď aukcia skončí, môže si pozrieť výsledky.

Meno	Heslo
n	n
user	user

Je možné vytvoriť ďalšie účty.

Administrátor

Môže vytvárať nové aukcie a upravovať vytvorené aukcie.

Meno	Heslo
a	a
admin	admin

Je možné vytvoriť ďalšie účty.

Majiteľ

Okrem toho čo robí administrátor má možnosť ešte spravovať administrátorov.

Meno	Heslo
o	o
owner	owner

Nie je možné vytvoriť ďalšie účty.

Aukcia

Pri vytváraní aukcie sa najskôr vyberie typ koňa, ktorý bude v aukcii a následne sa vyplnia potrebné dáta. Počas priebehu aukcie sa dajú meniť niektoré informácie (tie ktoré výrazne neovplyvnia jej priebeh).

Hlavné kritériá:

Hierarchia tried koní

Dedenie

Od triedy Horse dedia triedy Cart horse, Racing horse, Show horse, pričom každá z týchto dediací tried má okrem vlastností rodiča aj svoju špecifickú vlastnosť.

Polymorfizmus

Triedy obsahujú funkciu data(), ktorá na základe typu koňa vracia príslušné údaje.

Hierarchia tried používateľov

Dedenie

Trieda User, od ktorej dedí Normal user (ukladá si aj dáta o aukciách) a Admin (má aj informácie o počte aukcií), od ktorého ešte dedí Owner (má aj dáta o adminoch).

Polymorfizmus

Triedy Admin a Owner obsahujú metódu createAuction(), ktorá vytvorí novú aukciu na základe typu koňa.

Zapúzdzrenie

Triedy využívajú gettery a settery na prácu s ich údajmi a dáta v nich sú zapuzdrené skrz atribút protected.

Agregácia

Bežný užívateľ má v sebe zoznam aukcií do ktorých vložil svoju ponuku.

```
public Normal_user(String username, String password) { su  
protected ArrayList<Auction>myBid=new ArrayList<>();  
|
```

Kód organizovaný do balíkov

Kód je organizovaný do 5 balíkov.

```
> app  
> controler  
> model  
> Users  
> view
```

Ďalšie kritériá:

Použitá vlastná výnimka

Trieda s vlastnou výnimkou

```
package app;  
  
public class OwnException extends Exception {  
    public OwnException(String errorMessage) { super(errorMessage); }  
  
}
```

Používa sa pri nesprávne zadanom vstupe

```

}catch (Exception exception){

    try {
        throw new OwnException("Zlý formát");
    }
}

```

Serializácia

Použitá na ukladanie údajov v aplikácii skrz triedu databáza, pričom aj triedy ktoré sú ukladané tiež musia implementovať Serializable.

```

public class Database implements Serializable {

    private static final long serialVersionUID = 622197535515987367L;
    private static Database db;
    private Database(){}
    public static Database getDb(){
        if(db ==null){
            db =new Database();
        }
        return db;
    }

    public static ArrayList<Auction>auctions = new ArrayList<>();
    public static ArrayList<User>users = new ArrayList<>();
    public static ArrayList<Auction>endAuctions= new ArrayList<>();
}

```

Grafické používateľské rozhranie so spracovávaním udalostí

Viacnitéžnosť

Trieda time beží od spustenia programu ako samostatná niť a slúži na zisťovanie ukončenia aukcií a spracovanie výsledkov z nich.







```

public class Time extends Thread{
    public void run(){
        while (true){
            ArrayList<Auction> auctions=Database.getDb().auctions;
            Date date = new Date();
            date.getTime();
            Integer minutes=date.getMinutes()+(date.getHours()*60);
            Integer minauc;
            for(Auction auc:auctions){
                //over minúty a hodiny
                ArrayList<Auctioner> auctioners=auc.getParticipants();
                Auction auc_end= new Auction();
                auc_end=auc;
                minauc=auc.getDate().getMinutes()+(auc.getDate().getHours()*60);
            }
        }
    }
}

```

Oddelenie GUI od aplikačnej logiky

Ovládanie obrazoviek a spracovávanie dát z nich s prepojením na aplikačnú časť zabezpečujú kontrolery, v ktorých sa ovláda spracovávanie dát.

- ▼  controler
 -  AddHorseControl
 -  AdminsetControl
 -  EditAuctionControl
 -  LoginViewControl
 -  MainViewControl

Použitie lambdy v aplikačnej logike

Lambda je použitá ako náhrada cyklu

```

public static void output(){
    auctions.forEach(auction -> {
        StringBuilder sb = new StringBuilder();
        sb.append("Minimálna cena: "+auction.minPrice());
        for (Horse h:auction.getHorse()){
            sb.append(h.data());
        }
    });
}

```

RTTI

Pri prihlasovaní zisťujem o aký typ užívateľa sa jedná a na základe toho ho presmerujem na ďalšiu obrazovku alebo mu povolím niektoré akcie.

Diagram

