

Slovenská technická univerzita v Bratislave  
Fakulta informatiky a informačných technológií

Objektovo orientované programovanie

**Správa o realizácii projektu**

**Systém e-aukcií výtvarného umenia**

Akademický rok 2021/2022

**Meno:** Ján Ágh

**Cvičiaci:** Ing. Peter Križan

**Dátum:** 10.5.2022

**Počet strán:** 7

# Obsah

<b>1 Zámer projektu .....</b>	<b>1</b>
<b>2 Diagram najdôležitejších tried.....</b>	<b>2</b>
<b>2.1 UML diagram.....</b>	<b>2</b>
<b>2.1 Popis dôležitých tried a rozhraní.....</b>	<b>2</b>
<b>3 Splnené kritériá hodnotenia.....</b>	<b>4</b>
<b>3.1 Hlavné kritériá .....</b>	<b>4</b>
<b>3.2 Vedľajšie kritériá.....</b>	<b>5</b>
<b>4 Zoznam hlavných verzií programu .....</b>	<b>7</b>

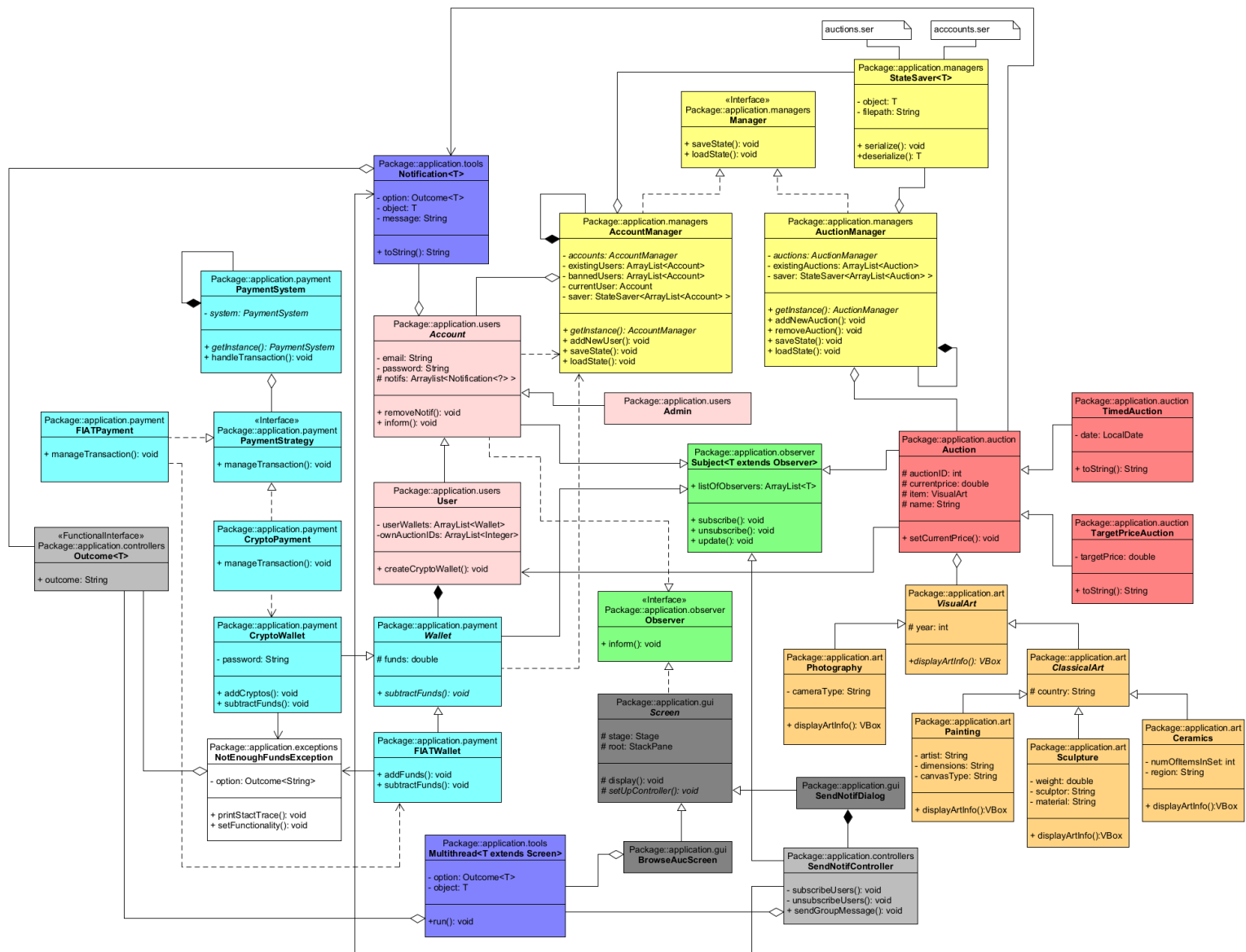
# 1 Zámer projektu

## Systém e-aukcií výtvarného umenia

Cieľom projektu je vytvorenie systému e-aukcií zameraného na obchodovanie so vzácnymi tvorbami výtvarného umenia (moderná aj klasická tvorba). Systém umožní používateľom otvoriť si nové konto uvedením mena, priezviska, emailovej adresy v platnom tvare a hesla. Na jednu emailovú adresu môže byť viazané iba jedno konto. Správou kont (tvorbou, uchovávaním, mazaním a komunikáciou) sa bude zaoberať systém manažovania kont. Každý používateľ dostane automaticky k svojmu kontu aj elektronickú peňaženku na uchovávanie tradičných finančných prostriedkov a bude mu umožnené aj vytvorenie kryptopeňaženky (pri tvorbe aukcií bude možné zadať, akými prostriedkami sa dá platiť). Používatelia budú môcť vytvárať vlastné aukcie iba po vyplnení ďalších osobných informácií (tel. číslo, adresa...), nakupovať však budú vedieť aj bez vyplnenia. Systém umožní používateľom tvorbu dvoch druhov aukcií – ohraničených (aukcia sa skončí po dosiahnutí vopred zadefinovanej sumy peňazí alebo 5 dní po poslednej ponuke) a neohraničených (aukcia sa skončí po uplynutí zadefinovanej doby bez ohľadu na aktuálnu čiastku). Po ukončení aukcie systém manažovania aukcií odstráni aukciu a v prípade úspešného predaja informuje systém riadenia platieb, ktorý spracuje žiadosť o presun peňazí a premiestni peniaze / kryptomeny z účtu kupujúceho na účet predajcu. Systémy manažovania aukcií a kont budú komunikovať s externou databázou, ktorá bude realizovaná formou textových súborov.

## 2 Diagram najdôležitejších tried

### 2.1 UML diagram



### 2.1 Popis dôležitých tried a rozhraní

#### Package application.payment:

Wallet – abstraktná trieda, obsahuje používateľove peňažné prostriedky (štandardné aj kryptomeny)

FIATWallet – obsahuje metódy na manipuláciu so štandardnou peňaženkou

CryptoWallet – obsahuje metódy na manipuláciu s peňaženkou kryptomien

PaymentSystem – singleton, slúži na realizáciu transakcií medzi dvoma používateľmi

PaymentStrategy – rozhranie, tvorí kostru návrhového vzoru Strategy

FIATPayment – súčasť návrhového vzoru Strategy, obsahuje metódu na platbu so štandardnou menou

CryptoPayment – súčasť návrhového vzoru Strategy, obsahuje metódu na platbu s kryptomenami

### **Package application.users:**

Account – abstraktná trieda, obsahuje základné údaje používateľov (meno, heslo, email, notifikácie)

User – obsahuje všetky dôležité vlastnosti používateľa (peňaženky, zoznam aukcií)

Admin – obsahuje všetky dôležité vlastnosti administrátora

### **Package application.managers:**

StateSaver – generická trieda, poskytuje metódy na serializáciu a deserializáciu objektov

AccountManager – singleton, obsahuje zoznam všetkých používateľov, komunikuje s StateSaver

AuctionManager – singleton, obsahuje zoznam všetkých aukcií, komunikuje s StateSaver

### **Package application.art:**

Painting – obsahuje všetky údaje o predávanej maľbe

Sculpture – obsahuje všetky údaje o predávanej soche

Ceramics – obsahuje všetky údaje o predávanej keramike

Photography – obsahuje všetky údaje o predávanej fotografii

### **Package application.auction:**

Auction – obsahuje druh predávaného tovaru a iné dôležité vlastnosti (názov, aktuálna cena)

TimedAuction – špeciálny druh aukcie, skončí po zadanom čase

TargetPriceAuction – špeciálny druh aukcie, skončí po dosiahnutí zadaného cieľa

### **Package application.observer:**

Subject – dedič z nej sledované triedy a obsahuje zoznam sledovateľov, ktorých môže informovať

Observer – rozhranie, implementujú ho sledovatelia

### **Package application.tools:**

Notification – generická trieda, obsahuje text notifikácie alebo správy, slúži na komunikáciu

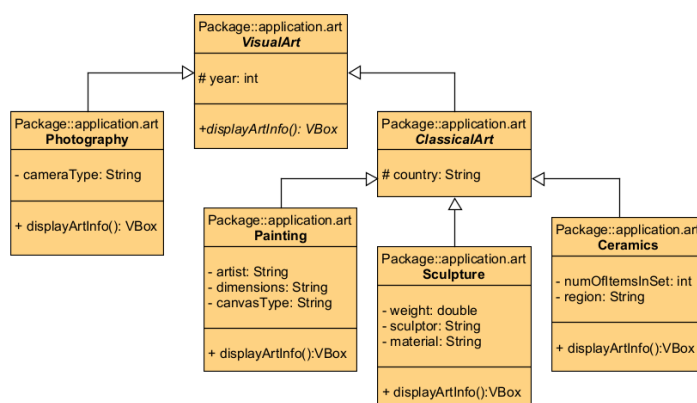
Multithread – generická trieda, slúži na vykonávanie viacerosti

## 3 Splnené kritériá hodnotenia

### 3.1 Hlavné kritériá

**Miera splnenia zadania a organizácia projektu** – Projekt spĺňa všetky dôležité záležitosti uvedené v kapitole č. 1, vrátane dvoch druhov peňaženíek, dvoch druhov aukcií a možnosti uloženia údajov v externých súboroch. Jednotlivé súbory aplikácie sú organizované do balíkov.

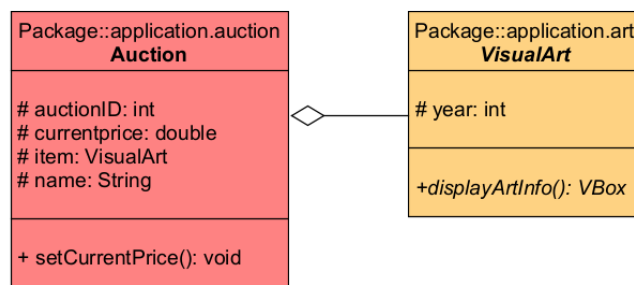
**Dedenie** – V projekte sa nachádza celkom päť rozličných hierarchií dedenia (ak nepočítame dedenie od triedy Subject).



**Polymorfizmus** – Príkladom polymorfizmu je trieda `Subject` používaná v návrhovom vzore `Observer`, konkrétne metóda `update()` slúžiaca na upovedomenie sledovateľov (u každého sledovateľa sa vykoná metóda `inform()`, pričom každý sledovateľ si sám určuje implementáciu metódy). Polymorfizmus sa tiež používa v balíku `application.art` (obrázok vyššie, abstraktná trieda `VisualArt` v sebe definuje metódu `displayArtInfo()`, ktorá je prekonávaná vo zvyšných triedach. V aplikácii sa nachádzajú iba premenné typu `VisualArt`, do ktorých sú vkladané jednotlivé inštancie `Ceramics`, `Sculpture`... a vykonáva sa vždy implementácia `displayArtInfo()` uvedená v danom type.).

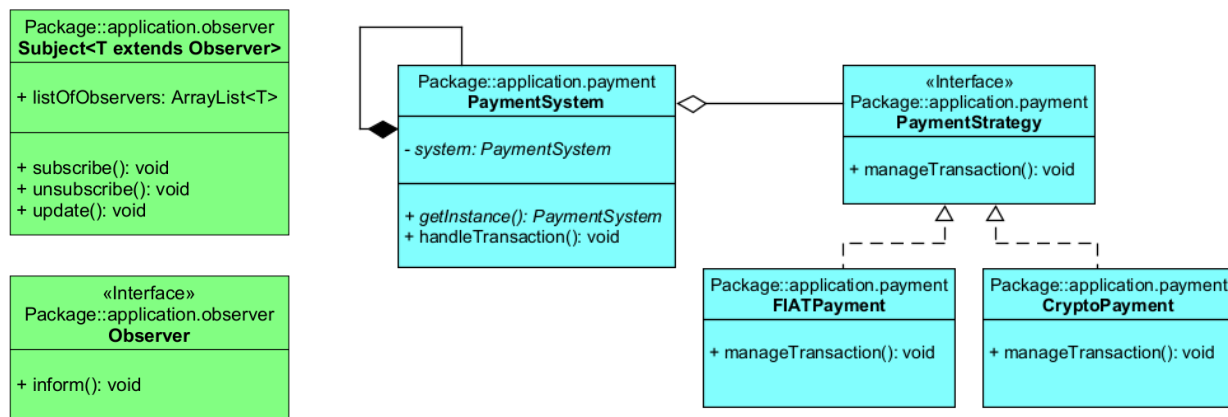
**Zapuzdrenie** – V celom projekte sa používajú iba *private* a *protected* premenné, ku ktorým sú v prípade potreby vytvorené príslušné `get()` a `set()` metódy.

**Agregácia** – Ukážka agregácie sa nachádza na obrázku nižšie.

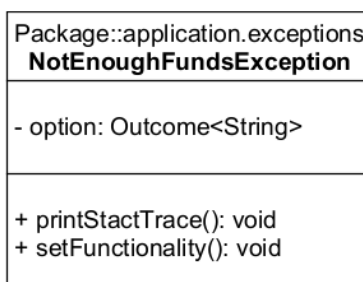


### 3.2 Vedľajšie kritériá

**Návrhové vzory** – V projekte sa okrem návrhových vzorov Singleton a MVC používajú aj Observer a Strategy. Observer sa prevažne používa na odosielanie hromadných správ adminom alebo aktualizovanie informácií na obrazovke, Strategy má svoje využitie v systéme transakcií.

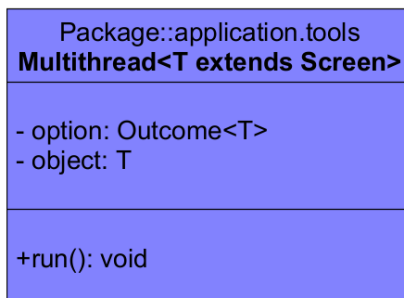


**Vlastné výnimky** – Projekt obsahuje tri vlastné výnimky, pričom každá z nich sa používa v odlišných situáciách. Sú to: *NotEnoughFundsException*, *ExistingAccountException* a *WrongPasswordException*.

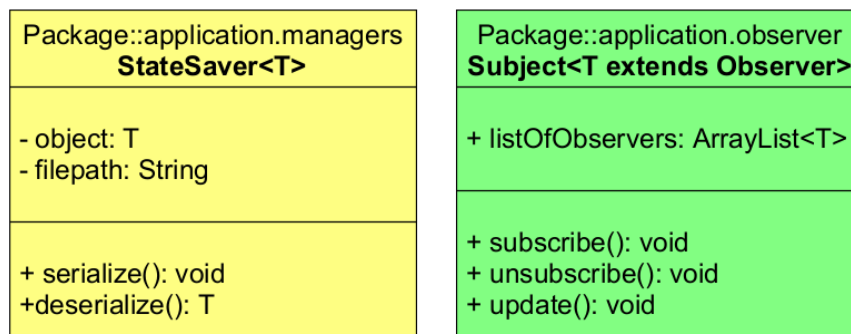


**GUI oddelené od aplikačnej logiky** – V projekte sa striktne dodržiava vzor MVC. Jednotlivé Views a Controllers sa navyše nachádzajú v oddelených balíkoch.

**Multithreading** – Používa sa v dvoch podobách – na zmenu GUI prvkov alebo vykonanie akcie, ktorá nesúvisí s GUI, v oddelenom vlákne.



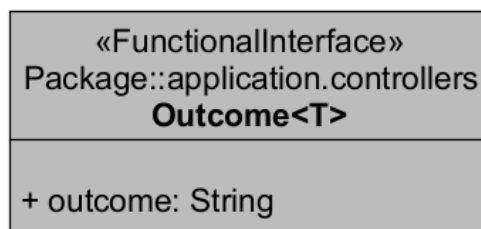
**Generickosť** – Projekt obsahuje mnoho generických tried. Príkladmi môžu byť trieda *Subject*, ktorá takto umožňuje vkladanie mnoho rozličných druhov sledovateľov, alebo *StateSaver*, ktorý pomocou generickosti dokáže zaserializovať hocijaký typ objektu.



**RTTI** – Používa sa hlavne v controlleroch pri downcastingu na overenie, či sa daný typ rodičovského objektu skutočne dá downcastnúť na želaného potomka.

**Vhniezdené triedy** – Projekt obsahuje jednu vhníezdenú triedu, a to *AuctionIDGenerator* vrámci triedy *NewAucController*. Táto vhníezdená trieda slúži na generovanie unikátneho kľúča pri vytvorení novej aukcie.

**Lambda výrazy** – Projekt využíva lambda výrazy predovšetkým na rozšírenie počtu situácií, v ktorých sa niektoré triedy môžu používať. Príkladom môže byť dvojica *CryptoPassDialog* a *CryptoPassController*, kde sa pomocou lambda výrazu rozhodne, čo sa má vykonať po zadaní správneho hesla ku kryptopeňaženke, alebo trieda *Multithread*, kde tiež pomocou lambda výrazov dokážeme vykonávať akúkoľvek časť kódu v odlišnom vlákne. Lambda výrazy sú ukladané do premenných typu *Outcome*, čo je rozhranie typu *FunctionalInterface*.



**Serializácia** – Používa sa na uloženie stavu dôležitých objektov po každej vykonanej zmene v nich. Serializáciu, rovnako ako aj deserializáciu, vykonáva generická trieda *StateSaver*. V projekte sa serializujú dynamické polia obsahujúce všetkých používateľov a aukcie.



## 4 Zoznam hlavných verzií programu

Táto kapitola obsahuje zoznam všetkých verzií programu, ktoré boli nahrané na platformu GitHub.

- 23.3.2022 – Project structure updated
- 23.3.2022 – Additional structure updates
- 29.3.2022 – MVC implementation – beginning
- 30.3.2022 – Experiments with V-C communication
- 31.3.2022 – Creation of additional V-C pairs + beginning of work on M
- 2.4.2022 – New Views created + Login system
- 5.4.2022 – Extensive work on multiple use cases
- 5.4.2022 – Working on auction creation
- 6.4.2022 – Working on auction creation + browsing
- 6.4.2022 – ID fix
- 7.4.2022 – GUI changes and improvements
- 9.4.2022 - Admin menu added + work on auction browser
- 11.4.2022 – Added ability to bid
- 16.4.2022 – Auction items are now implemented as objects + observer pattern
- 17.4.2022 – Implemented first own exception
- 17.4.2022 – Implemented FIAT payment system + strategy pattern
- 22.4.2022 – Minor changes + work on notification system
- 23.4.2022 – Implemented notification system using observer pattern
- 25.4.2022 – Created crypto payment system + admin pannel
- 26.4.2022 – Testing (NOT FUNCTIONAL VERSION)
- 28.4.2022 – Message system + multithreading + many more improvements
- 30.4.2022 – Multiple changes + README rework
- 30.4.2022 – Further README additions
- 10.5.2022 – Finished project + documentation + javadoc