Lagatoria

1. Názov projektu je Lagatoria. Hlavnou úlohou projektu je vydávanie kníh, teda to zahŕňa napísanie textu, jeho korektúru, návrh obálky, zistenie aký je dopyt po knihe, vytlačenie kníh s tým spojené rozhodovanie sa, koľko kusov sa vytlačí a následná distribúcia výtlačkov. Ďalej sa knihy predávajú v kníhkupectve, kde si ich môže kúpiť zákazník.

2. Program je z veľkej časti funkčný. Funguje pridávanie a odoberanie ako autorov, tak aj odoberateľov, Používateľské rozhranie je rozdelené podľa vzoru MVC . Funguje vydávanie kníh ako aj ich distribúcia medzi odoberateľov. Dokážeme zmeniť stratégiu ako sa knihy vydávajú. Čo sa týka kníhkupectva, tak dokážeme objednať knihy. Premiestňovať ich do rôznych sekcií a pod.

Hlavný súbor s metódou main() je junas.robert.lagatoria.gui.Main.java

Parametre na spustenie nie sú žiadne, treba mať povolene javaFx

Pravdepodobne je potrebné si vytvoriť priečinok /res/, kde sa uloží serializacia

3. Aspoň **3 vrstvovú hierarchiu dedenia** nám tvorí dedenie používateľov. Na vrchu je interface lagatoria.utils.InputProcess ten je implementovaný abstraktnou triedou lagatoria.users.Pouzivatel, Pouzivatel je dedený triedou Zakaznik a abstraktnou triedou Zamestanec. Nakoniec triedy Skladnik, Predajca, Distributor, Manazer a aj vnorené triedy Korektor, Dizajner (v triede Vydavatelstvo.java) dedia triedu Zamestnanec.









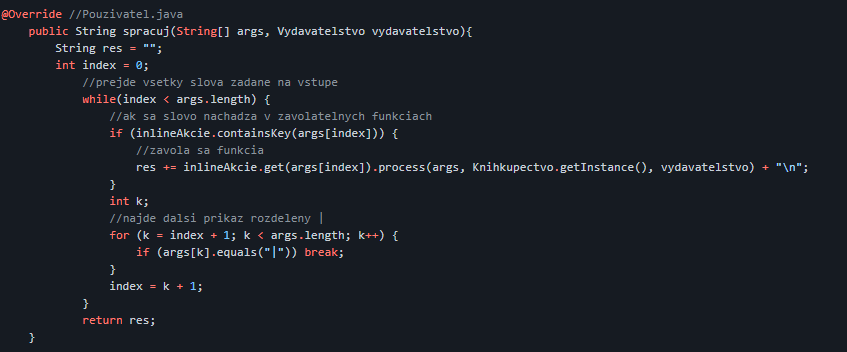


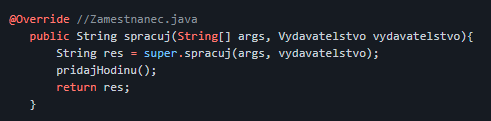


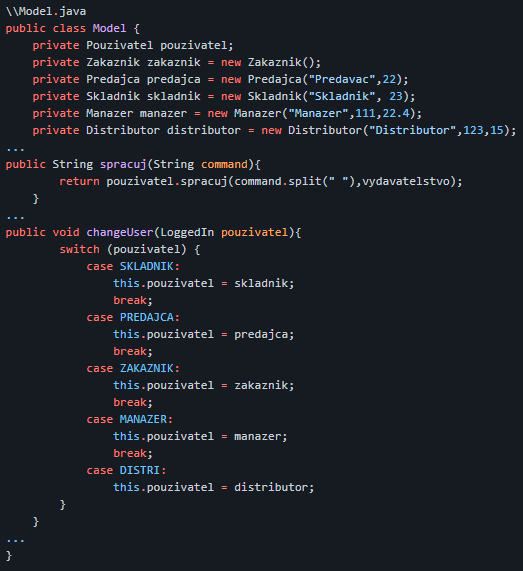




4. Polymorfizmus je napríklad v tom ako sa spracovávajú vstupy v triede Pouzivatel.java sa definuje spôsob akým sa spracováva vstup a následne Zamestnanec.java prekonáva funkciu tým že s ňou ešte volá pridajHodinu(), čo pridá zamestnancovi jednu odrobenú hodinu, za použitie metódy.

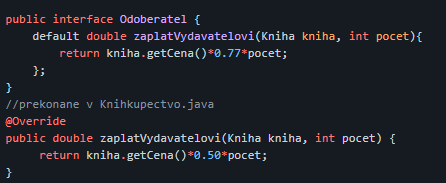






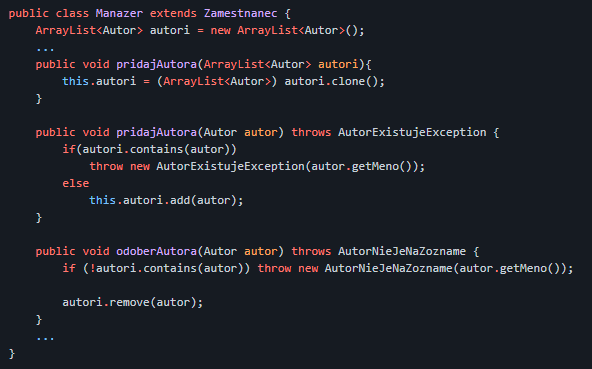
Obrázok 1 Volanie funkcie spracuj z premennej pouzivatel

Ďalší príklad:

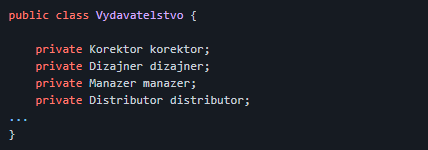


Obrázok 2 Override funkcie v Knihkupetvo.java

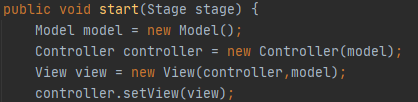
5. Napríklad trieda Vydavatelstvo.java v sebe agreguje Manazer.java. Knihy sa agregujú v regáloch v prípade Knihkupectva. Ďalej v sebe Manazer agreguje autorov, ktorý sú pripravený písať.



Ďalší príklad:



6. Používateľské rozhranie je definovane v triede View.java a aplikačná logika sa kumuluje v Model.java a Controller.java je iba na komunikáciu medzi nimi. Nakoniec sa všetky časti prepájajú v Main.java.



7. Máme navyše implementovanú serilizáciu na strane kníhkupectva, kde si ukladáme stav všetkých poličiek a kníh v kníhkupectve súbor s dátami ukladáme do zložky res/.

Ďalej máme implementovaný návrhový vzor observer medzi Manazer a Autor. Ďalším implementovaným návrhovým vzorom je Composite a je spojený s vytváraním kníh. Ďalej sme implementovali model Visitor a to pri implementácií Autora. Ako posledný sme implantovali Strategy vo Vydavatelstvo.java, kde sa mení metóda vydávania kníh.

Implementované máme aj vlastné výnimky: AutorExistujeException, AutorNieJeNaZozname, InvalidFormatException. Nachádzajú sa v core.utils.exceptions.

Využili sme aj multithreading pri autoroch, ktorý dokážu naraz písať knihy.

Explicitne sme využili RTTI, pri zisťovaní kam sa máju knihy a ako poslať.

Máme implementované aj vnorene triedy vyhniezdené triedy a rozhranie vo Vydavatelstvo.java.

Máme implementovanú aj default metódu na platenie vydavateľstvu v rozhraní Odoberatel.java