

Fakulta informatiky a informačných technológií STU, Ilkovičova 2, Bratislava

Patrik Drdák

Drive Joy

Obsah

1	Zámer projektu.....	3
2	Controllers.....	3
3	Najdôležitejšie triedy.....	4
3.1	Trieda Database.....	5
3.2	Trieda RouteMemory	5
3.3	Trieda DisplayOptions	5
3.4	Trieda EditorRoles	6
4	Kritéria	7
4.1	Dedenie	7
4.2	Agregácia	7
4.3	Polymorfizmus.....	8
4.4	Rozhranie + default	8
4.5	Zapuzdrenie	9
4.6	Oddelenie aplikačnej logiky od GUI, balíky (packages)	9
4.7	Návrhový vzor.....	9
4.8	Výnimky (exceptions)	10
4.9	Handlers v GUI.....	10
4.10	Multithreading	10
4.11	Generickosť, vnhiezdené triedy	10
4.12	RTTI.....	10
4.13	Serializácia	11
5	Commits.....	11
6	Návod na používanie	12

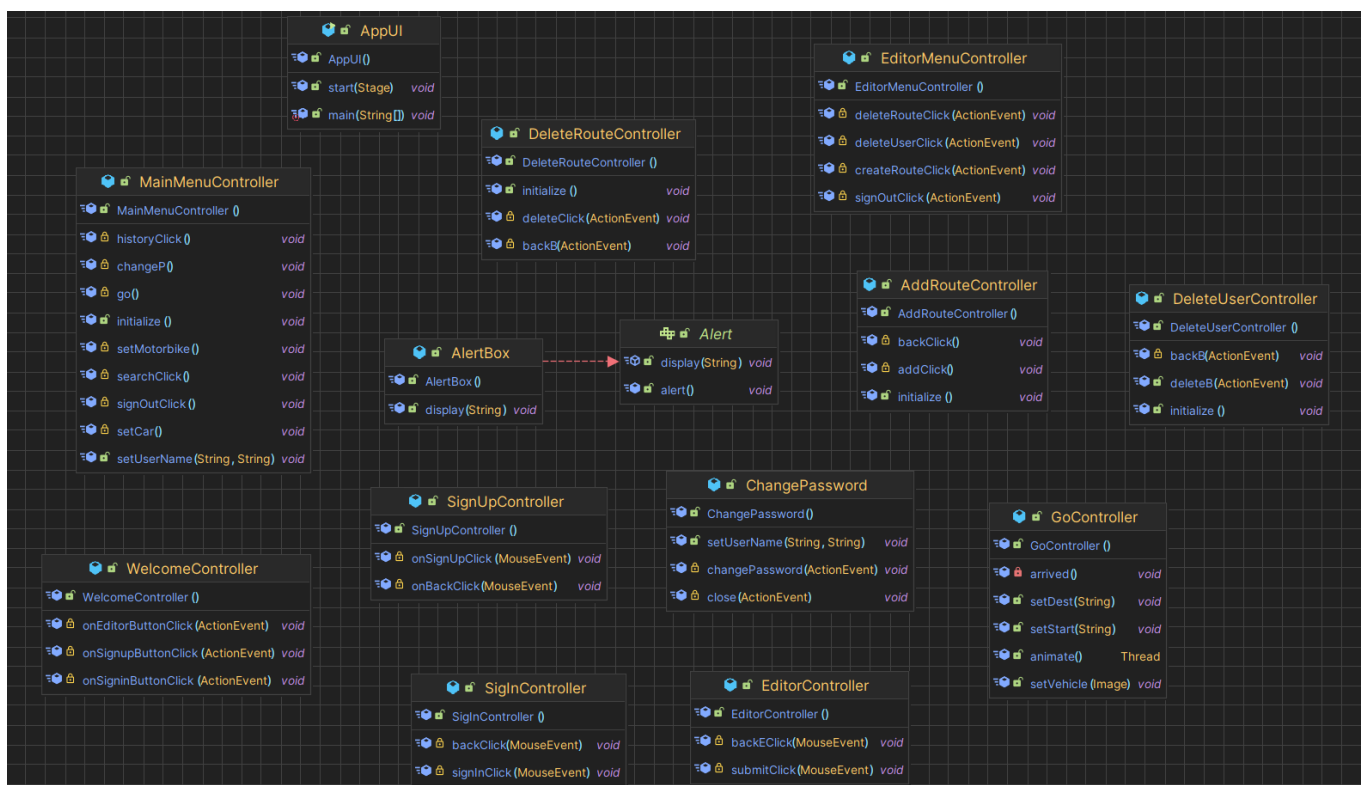
1 Zámer projektu

Táto aplikácia bude slúžiť na hľadanie ciest pre používateľov, ktorí si chcú užiť zábavu a adrenalín počas svojej jazdy. Bude umožňovať používateľom vyhľadať trasu na základe polohy, ktorú zadajú. Táto aplikácia by mohla byť veľmi užitočná pre nadšencov motorizmu, ktorí hľadajú nové výzvy a zábavu na cestách. Pri vývoji tejto aplikácie by bolo dôležité zohľadniť faktory, ktoré ovplyvňujú bezpečnosť jazdy, ako napríklad stav ciest a poveternostné podmienky. Preto by aplikácia mala obsahovať aj informácie o aktuálnych podmienkach na cestách. Výsledná aplikácia by mala byť intuitívna a jednoduchá na použitie, aby si ju mohli užívať nielen skúsení vodiči, ale aj bežní ľudia s obmedzenými skúsenosťami v jazde autom.

2 Controllers

Tieto controlleri slúžia na ovládanie jednotlivých FXML súborov (scén). Každý FXML súbor má svoj vlastný controller, v ktorom sú rôzne funkcie slúžiace na vykonanie akcií súvisiacich so stlačením tlačidla alebo s písaním do poľa.

Trieda `AlertBox` nemá svoj FXML súbor, ale priamo vytvára novú scénu.



3 Najdôležitejšie triedy

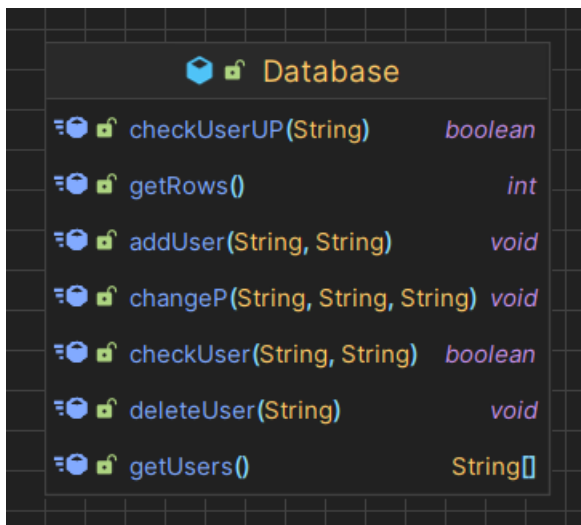
- Database
- RouteMemory
- DisplayOptions
- EditorRoles

Diagram všetkých tried (vyznačené sú najdôležitejšie):



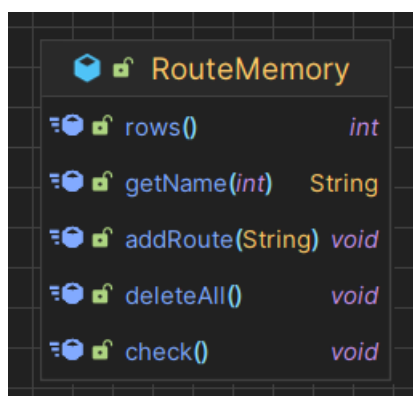
3.1 Trieda Database

V tejto triede sa vykonáva väčšina logických akcií spojených s prihlasovaním a registrovaním používateľov, ako aj akcie spojené s týmito údajmi (zmena hesla (user), vymazanie používateľa (editor)).



3.2 Trieda RouteMemory

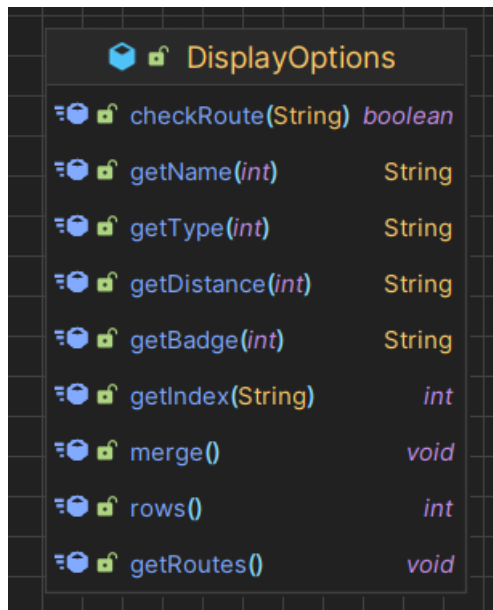
V tejto triede sa vytvára história ciest, ktoré používateľ vyhľadával, ako aj následne automatické vymazanie histórie pri prekročení maximálnej kapacity. Údaje sa ukladajú do textového súboru s názvom RouteMemory.txt.



3.3 Trieda DisplayOptions

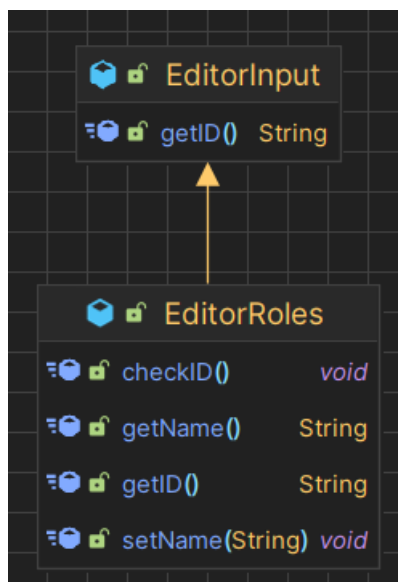
Trieda DisplayOptions slúži na načítanie trás z databázy (textový súbor s názvom Routes.txt) do polí. Tieto polia sú zapuzdrené a každé má vlastný getter. V tejto triede sú obsiahnuté aj funkcie na overenie, či sa daná trasa nachádza v databáze, a funkcia na spojenie destinácií (štart a cieľ) do jedného poľa (Locations), pretože

sa ľahšie pracuje s jedným poľom. Údaje sú uložené v tvare: štart(cieľ) – cieľ(štart) – odporúčaný smer – vzdialenosť – číslo cesty – odporúčané vozidlo.



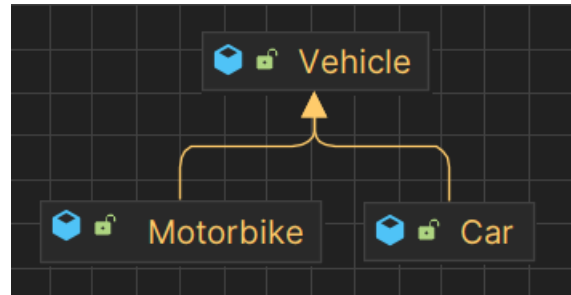
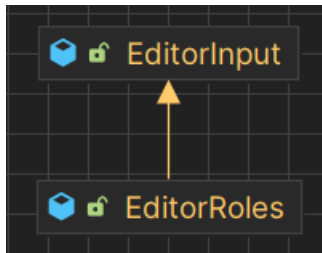
3.4 Trieda EditorRoles

Táto trieda zabezpečuje overenie ID pri prihlásení editora do aplikácie. Hodnotu zo vstupu porovnáva s triedou `Editor`, ktorá má pevne dané ID.



4 Kritéria

4.1 Dedenie



Dedenie je použité v dvoch oddelených hierarchiách ako je vyžadované. Prvé dedenie je v triede EditorRoles, ktorá od triedy EditorInput dedí ID, ktoré pri prihlasovaní zadá editor. Toto ID je následne vo funkcii checkID() porovnávané s triedou Editor, kde je pevne uložené ID pre objekt editor. Druhé dedenie je použité v triede Car a Motorbike, ktoré dedia od triedy Vehicle funkciu setGarage(Image image) na nastavenie obrázka podľa príslušnej triedy (Car alebo Motorbike, default je obrázok garáže) a funkciu setType(String Car)/setType(String Motorbike), ktorá nastaví typ vozidla podľa príslušnej triedy, na neskoršiu prácu s ním.

4.2 Agregácia

```

private UserSign userSign;

/**
 * Sets users name.
 * @param user the user
 */
Patrik-03 *
public void setUser(UserSign user) // set the username and password
{
    this.userSign = user;
}
  
```

Agregáciu používam v triede ChangePassword na získanie prístupu k vstupným údajom ktoré zadal používateľ pri prihlásení sa do aplikácie. Tieto údaje potrebujem na zmenu práve toho hesla ktorým sa používateľ prihlásil. Trieda UserSign existuje nezávisle od triedy ChangePassword, keďže trieda

ChangePassword je controller na scénu pre zmenu hesla a nemusí byť vôbec vytvorený, ak používateľ nezvolí možnosť zmeny hesla.

4.3 Polymorfizmus

Polymorfizmus je v triede MainMenuController, kde používateľ si volí typ vozidla (auto alebo motorku). Vytváram tu objekt Vehicle a podľa zvoleného typu vozidla buď objekt Car alebo Motorbike. Objekty Car a Motorbike metódou override prepisujú funkcie setGarage(Image image), setType(String Car)/setType(String Motorbike).

4.4 Rozhranie + default

```
/**
 * Interface for the exceptions
 */
2 implementations  🧑 Patrik-03
public interface Alert
{
    /**
     * String constant for the error message
     */
    1 usage
    String ERROR_MESSAGE = "Error";

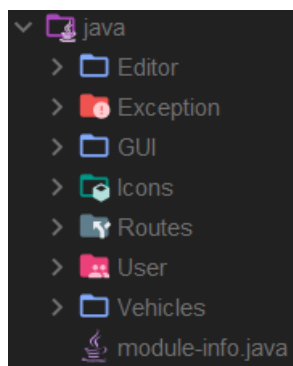
    /**
     * Method to display the alert window
     * @throws EmptyDatabase if the database is empty
     * @see EmptyDatabase
     * @see AlertBox
     */
    2 usages  🧑 Patrik-03
    default void alert() throws EmptyDatabase
    {
        AlertBox alertBox = new AlertBox();
        alertBox.display(ERROR_MESSAGE);
        throw new EmptyDatabase("Empty Database");
    }
}
```


Toto rozhranie implementujem v triede Database, kde sa nachádza funkcia `deleteUser(String name)`, ktorá slúži pre editora na vymazanie používateľa z databázy. Na začiatku tejto funkcie overujem, či argument nie je null, čo by znamenalo, že databáza je prázdna a nie je čo vymazať. V takom prípade volám default metódu z rozhrania ktorá do konzoly vyhodí chybovú hlášku a vytvorí pop up okno s nápisom „ERROR“. Rozhranie ešte implementujem v triede DeleteRoutes, kde je funkcia `deleteRoute(T name)`, slúžiaca pre editora na vymazanie trasy z databázy. Princíp fungovania rozhrania je rovnaký ako pri predošlej implementácii.

4.5 Zapuzdrenie

Vo všetkých triedach, v ktorých sú vytvorené premenné a je potrebný prístup k týmto premenným, sa nachádzajú gettre a settre na získanie alebo zápis údajov. K premenným nepristupujem priamo.

4.6 Oddelenie aplikačnej logiky od GUI, balíky (packages)



Kód je organizovaný v balíkoch pomenovaných podľa toho aké triedy sa v ňom nachádzajú. Napríklad v balíku „User“ sa nachádzajú triedy `RouteInput`, `UserSign` a `Vehicle`, ktoré slúžia na získanie zadaných údajov od používateľa, či už sa jedná o prihlasovacie údaje alebo typ zvoleného vozidla, či výber cesty.

Grafické rozhranie sa nachádza v balíku „GUI“, v ktorom sa nachádzajú všetky controlleri pre jednotlivé scény. V tomto balíku sa nenachádza žiadna aplikačná logika a zase v žiadnom inom balíku sa nenachádza grafické rozhranie.

4.7 Návrhový vzor

V projekte som použil návrhový vzor Factory. Implementoval som ho v triede `EditorRoles` kde cez triedu `EditorFactory` vytváram instanciu `Editor`, a tým vytváranie objektu `Editor` závisí na triede `EditorFactory` a nie `EditorRoles`. (takýmto spôsobom

by som vedel vytvárať aj prípadné ďalšie objekty, pomocou jednej triedy, ako napríklad Owner alebo Manager)

4.8 Výnimky (exceptions)

V projekte mám vytvorené dve vlastné výnimky, WrongIdException a EmptyDatabaseException. Prvá menovaná je uplatnená pri zadanom zlom ID pri prihlásení sa editora do aplikácie. Druhá výnimka je uplatnená v triede Database, kde sa nachádza funkcia deleteUser(String name), ktorá slúži pre editora na vymazanie používateľa z databázy. Táto výnimka sa uplatní pokiaľ je databáza prázdna, teda nie je čo vymazať. Tento istý princíp je implementovaný aj v triede DeleteRoutes, kde je funkcia deleteRoute(T name), slúžiaca pre editora na vymazanie trasy z databázy.

4.9 Handlers v GUI

Handlers používam v takmer každej funkcii slúžiacej na vykonanie určitej operácie po stlačení tlačidla v GUI, ku ktorému je priradená daná funkcia.

4.10 Multithreading

V triede MainMenuController pri zavolaní sa spustí funkcia initialize(). V tejto funkcii sú vytvorené tri vlákna. Prvé vlákno slúži na zavolanie funkcie na načítanie ciest z databázy, druhé slúži na načítanie lokácií pre autocomplete vo vyhľadávacom poli a tretie na načítanie histórie posledných zvolených trás z databázy.

Multithreading tiež využívam pri animácii v triede GoController, kde pohyb obrázka je realizovaný na novom vlákne.

4.11 Generickosť, vnhiezdené triedy

Trieda AddDeleteRoutes je generická trieda, ktorá je zavolaná v triedach DeleteRouteController a AddRouteController. V triede AddDeleteRoutes sa nachádzajú aj dve vnhiezdené triedy AddRoutes a DeleteRoutes.

4.12 RTTI

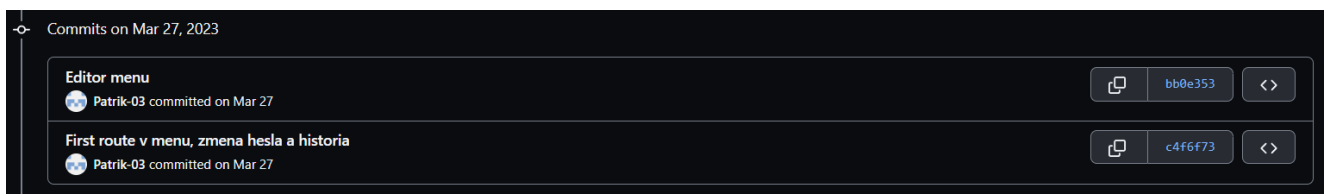
RTTI je uplatnené v triede AddRouteController, kde v metóde initialize() zisťujem názvy tried v balíku Vehicle, ktorý obsahuje typy vozidiel, ktoré si môže používateľ zvoliť. Tieto názvy následne pridávam ku informáciám o ceste aby aplikácia vedela odporučiť vhodnejší typ vozidla aký bol zvolený.

4.13 Serializácia

Serializácia je uplatnená v triede SetIcons, kde touto metódou získavam z triedy Vehicle typ vozidla, ktoré si užívateľ zvolil v aplikácii. Tento údaj slúži pre nastavenie vozidla v animácii (auto alebo motorka).

5 Commits

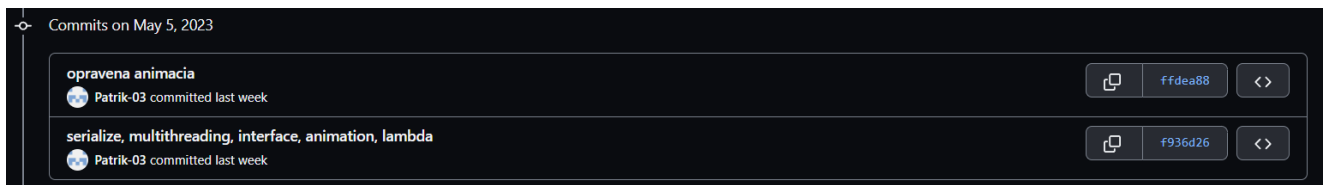
Tieto dva commity obsahujú vytvorenie menu pre editora a pridanie možnosti na zmenu hesla ako aj pridanie prvých ciest do databázy.



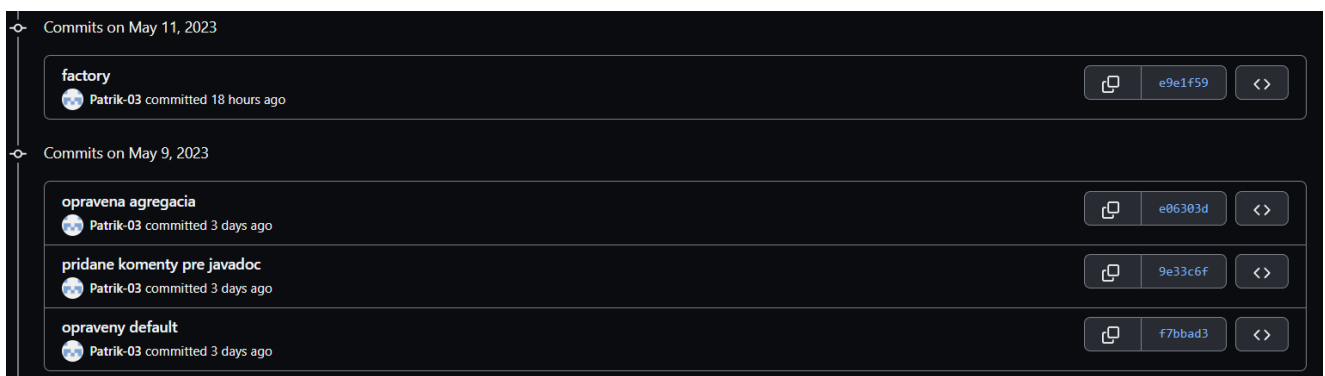
V tomto commite bolo pridané dedenie a polymorfizmus a vylepšené GUI.



Tieto commity boli asi najviac dôležité a obsahujú zmeny ako pridanie serializácie, multithreading, interface a aj vytvorenie animácie pre cestovanie.



Toto sú posledné commity v ktorých iba opravujem niektoré veci aby úplne spĺňali kritéria.



6 Návod na používanie

Po spustení je možné si vybrať z troch možností: Sign in, Sign up, Editor. Prvá možnosť je prihlásenie pre používateľov s vytvoreným účtom. Druhá možnosť je na založenie si účtu. Tretia možnosť je určená pre editora na prihlásenie.

Po prihlásení je potrebné si v menu „Profile“ v podmenu „Vehicle“ zvoliť typ vozidla, ktorým budeme cestovať. V tomto menu sa nachádzajú aj ďalšie možnosti ako sú „Change password“ pre zmenu aktuálneho hesla a „Sign out“ pre odhlásenie sa. V menu „Menu sa nachádza podmenu „History“, kde sú vyobrazené posledné cesty, ktoré používateľ vyhľadával.

Po zvolení typu vozidla môže používateľ zadať názov miesta odkiaľ plánuje začať trasu (sú dostupné iba niektoré lokácie, ktoré sa zobrazia napríklad pri napísaní písmena „a“, keďže toto sa nachádza v každej lokácii). Kliknutím na tlačidlo s lupou alebo stlačením klávesy Enter sa zobrazia údaje o trase ako aj tlačidlo „GO!“, po ktorého stlačení sa spustí animácia simulujúca cestovanie (vozidlá v tejto animácii sú generované náhodne a niekedy sa tam môže objaviť aj zaparkované policajné auto).

Uložené prihlasovacie údaje pre používateľa:

- Meno: a
- Heslo: a

Alebo

- Meno: test
- Heslo: test

V menu editora sa nachádzajú tlačidlá na pridanie alebo vymazanie trasy a vymazanie používateľa. Pre pridanie trasy je potrebné do prvého poľa zadať názov v tvare: Bratislava -> Trnava.

Údaje pre prihlásenie editora:

- ID: 1