

# **Slovenská technická univerzita v Bratislave**

## **Fakulta informatiky a informačných technológií**

### **Objektovo orientované programovanie**

### **Dokumentácia k projektu**

### **Cras 4 everyone**

Auto: Martin Beňa

Cvičiaci: Mgr. Pavle Dakić

Prednášajúci : doc. Ing. Valentino Vranić, PhD.

## Obsah

Zámer projektu .....	3
<b>Štruktúra aplikácie .....</b>	<b>3</b>
Diagram tried .....	3
Vnútorné triedy.....	4
Balíky .....	4
Dedenie a abstraktné triedy .....	5
Kompozícia.....	6
Celý diagram .....	7
<b>Hlavné kritéria .....</b>	<b>8</b>
<b>Dodatočné kritéria .....</b>	<b>9</b>
Ošetrovanie mimoriadnych stavov .....	9
Oddelenie aplikačnej časti od grafickej .....	9
V hniezdenie tried .....	9
Použitie serializácie .....	9
<b>Záver .....</b>	<b>10</b>

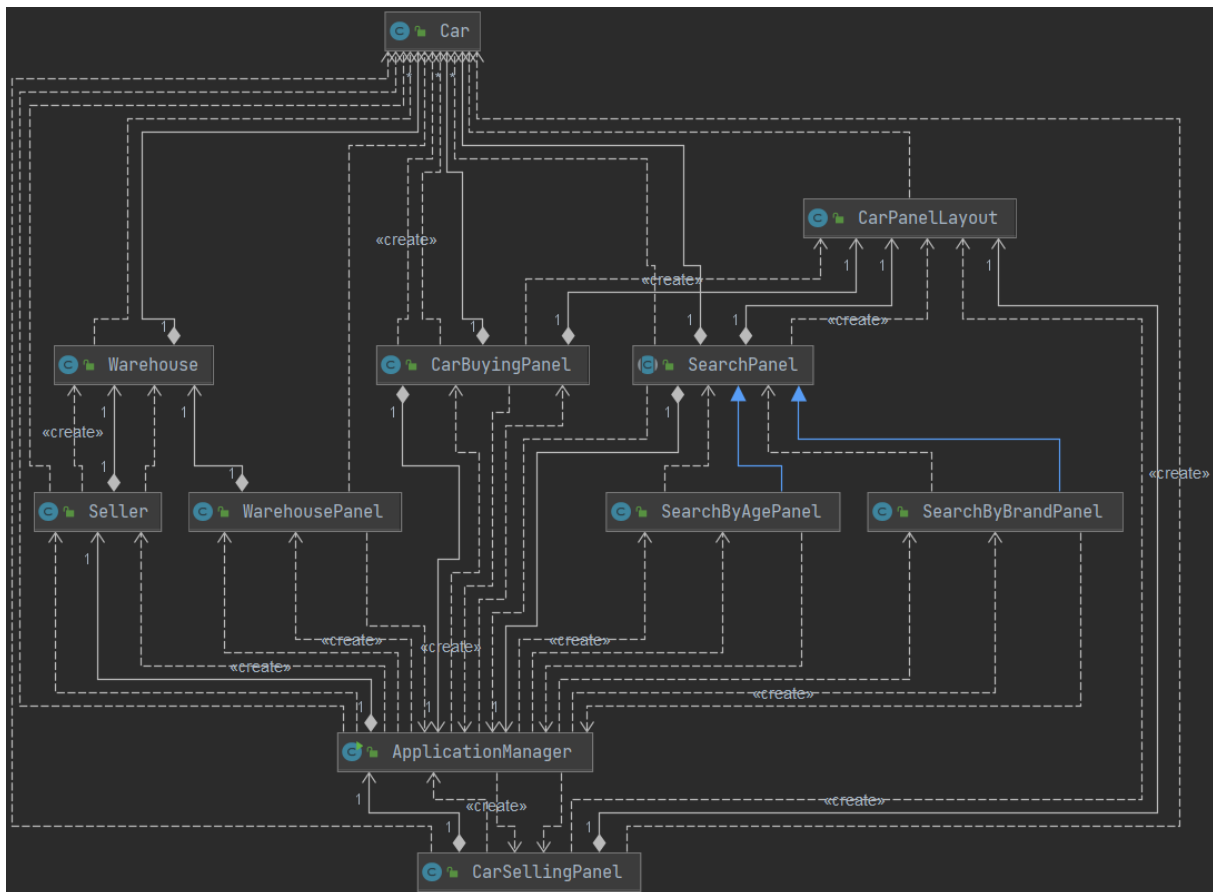
## Zámer projektu

Zámerom môjho projektu bolo vytvoriť aplikáciu pre autobazár s ojazdenými automobilmi. Vo finálnej verzii si zákaz môže vybrať automobil z aktuálnej ponuky. Vyhľadávať môže podľa veku alebo výrobcu auto. Taktiež si môže pozrieť všetky dostupné autá, ktoré sa aktuálne vyskytujú na predaj v bazáre.

Prvotným zadaním môjho projektu bolo o kúsok iné, malo sa jednať iba o kúpu motorového vozidla.

## Štruktúra aplikácie

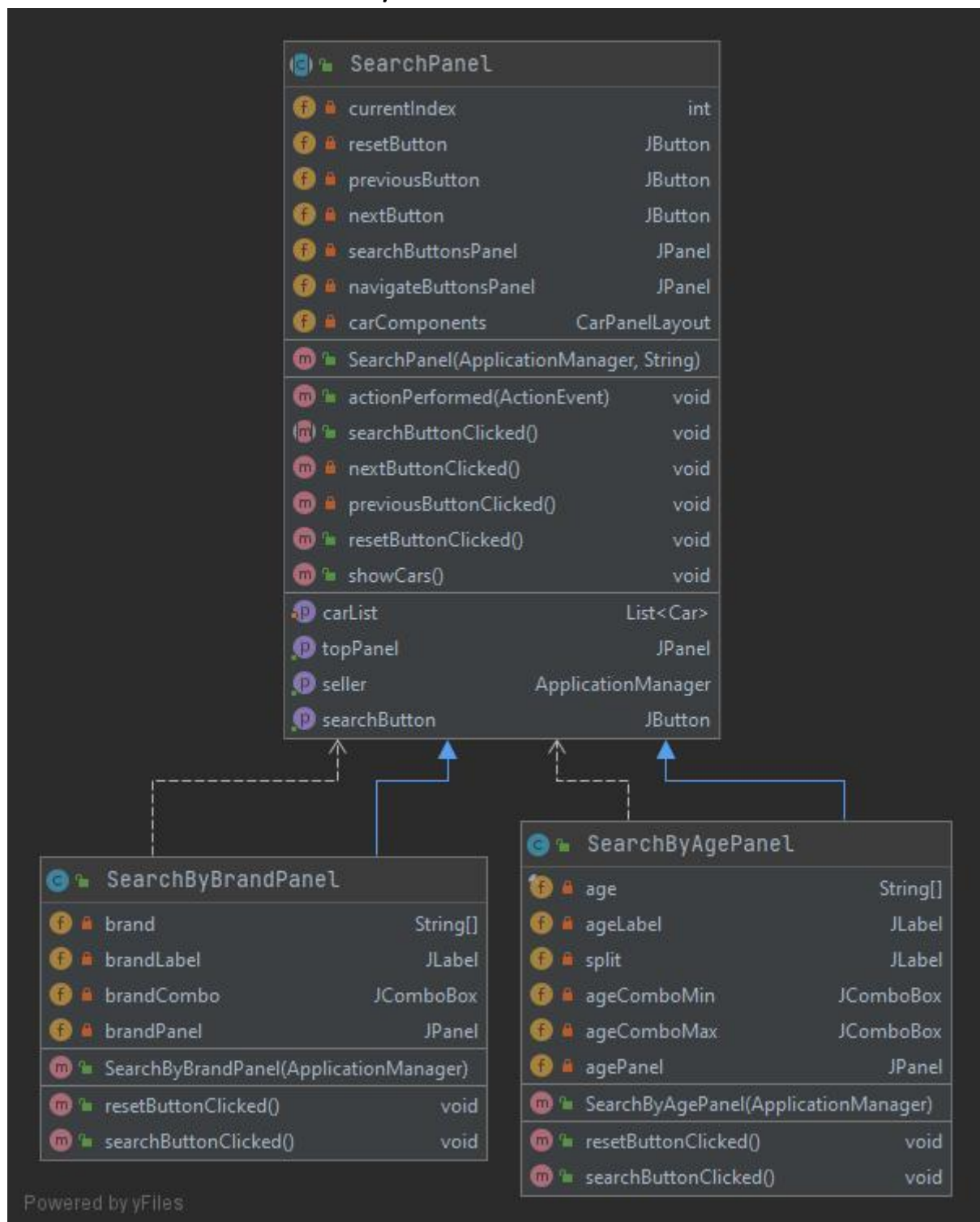
### Diagram tried



Tento diagram obsahuje triedy z oboch mojich balíkov lebo majú vzájomnú asociáciu. Ale neobsahuje vnútornú triedu, lebo sa mi ich štýl ich zobrazenia zdá rušivý, tak ich zobrazím v obrázku nižšie. Z diagramu môžeme vyčítať jednotlivé závislosti ako napríklad:

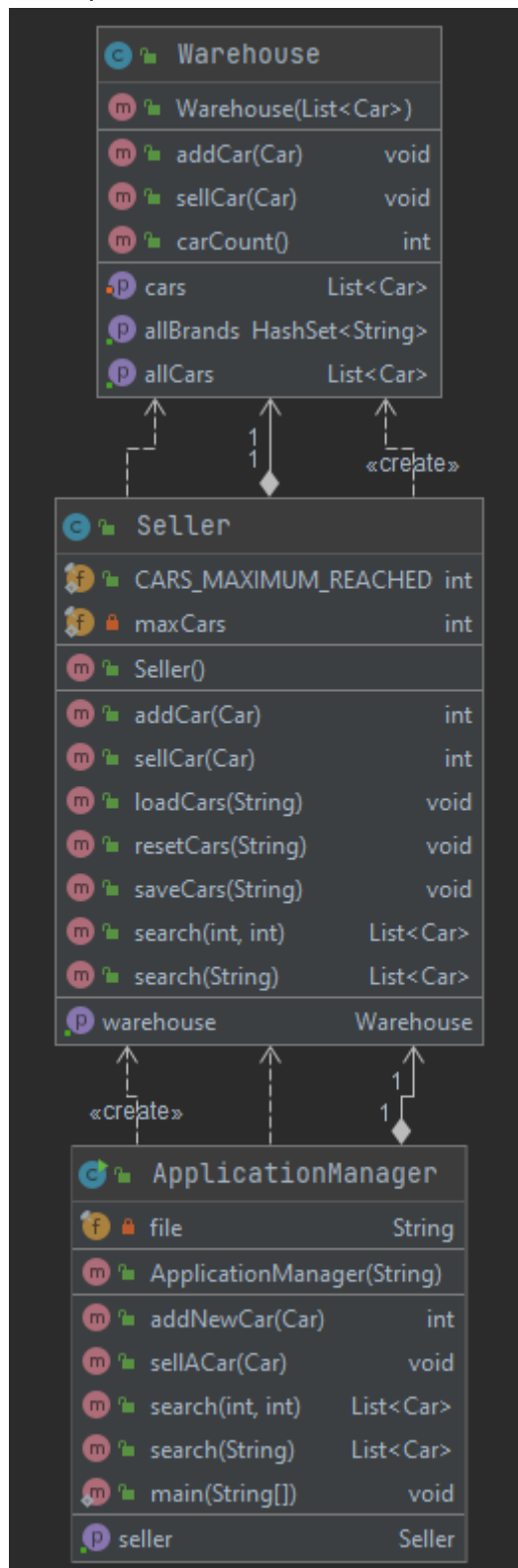


## Dedenie a abstraktné triedy



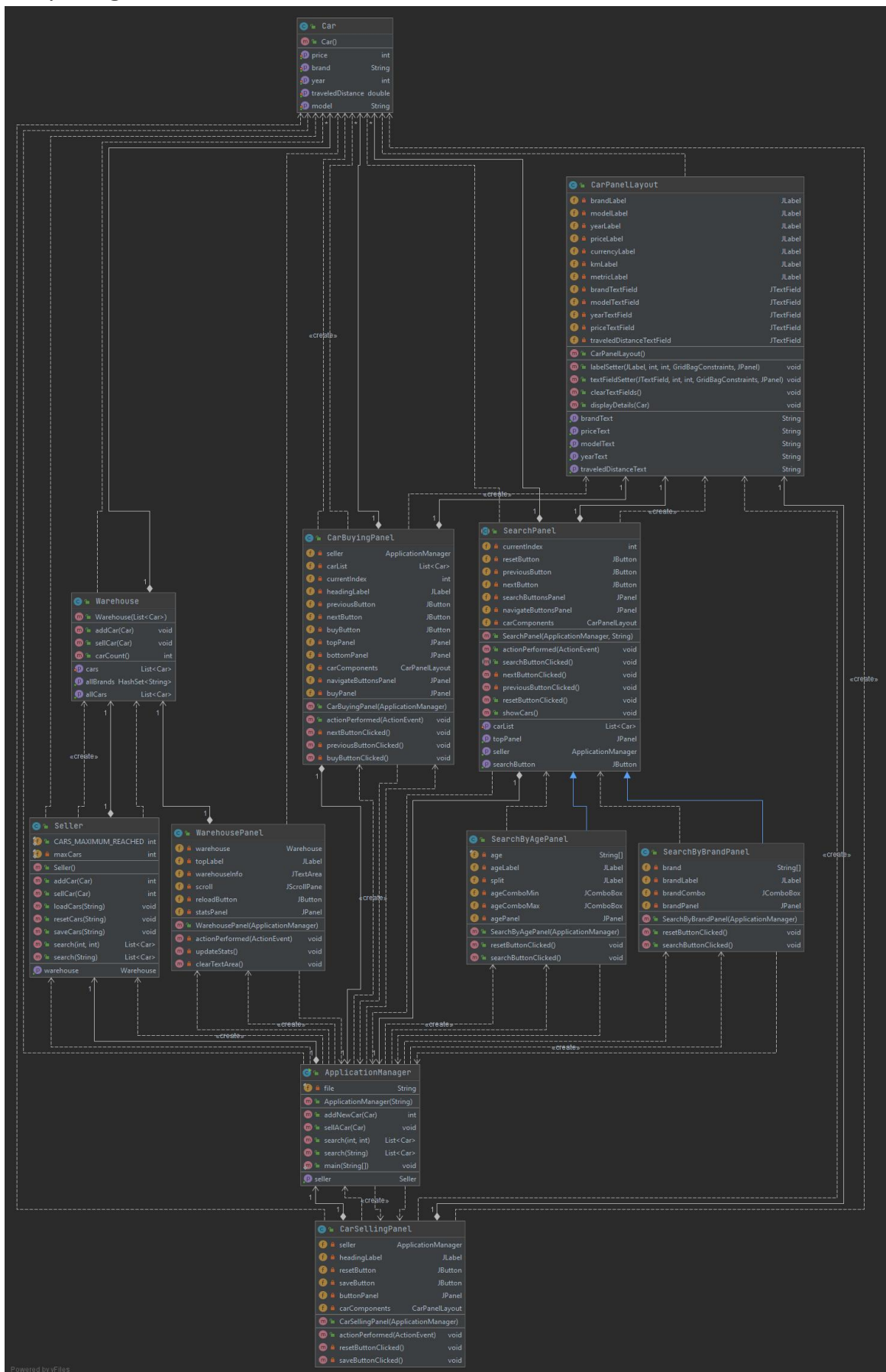
Na tom to diagrame môžeme vidieť ako 2 triedy **SearchByBrandPanel** a **SearchByAgePanel** dedia z tej istej abstraktnej triedy. V týchto trieda vzniká aj prekonavanie (Overload) metódy, konkrétne abstraktnej metódy `searchButtnoClick` a metódy `resetButtonClicked`.

## Kompozícia



Na tomto diagrame sú ukázané dve kompozície. Kompozícia je v UML diagramoch zväčša zobrazovaná ako čierny kosoštvorec ale nakoľko používam tmavé prostredie v mojom IDE tak je tu použitá biela farba.

## Celý diagram



Ale je vidno tak celý diagram je pomerne veľký a tým pádom aj do cela nečitateľný. Tak som si vo svojom projekte vytvoril zložky Diagram, ktorá bude mať v sebe všetky použité hierarchické diagramy vrátane celého diagramu.

## Hlavne kritéria

Zadané hlavné kritéria, ktoré by ma mal každý projekt spĺňať:

1. *Program musí byť funkčný a zodpovedať zadaniu a zámeru projektu, ktoré schválil vyučujúci, a zásadným požiadavkám vyučujúceho, ktoré vznikli počas realizácie projektu.*
2. *Odovzdaný zdrojový kód musí zahŕňať všetky potrebné súbory a musí sa dať preložiť v prostredí Eclipse inštalovanom v učebni, v ktorej sa realizujú cvičení.*

Bohužiaľ neviem ktorá verzia Eclipse je nainštalovaná v učebniach ale projekt bol spustiteľný na verzii 2021-03 a fungoval. Verzia JDK bola 1.8.

3. *Program musí obsahovať zmysluplné dedenie medzi vlastnými triedami s prekonávaním vlastných metód.*

Dedenie a prekonanie vlastnej metódy som zahrnul vyššie v dokumentácii, konkrétne v sekcii Štruktúra aplikácie -> Dedenie a abstraktná trieda

4. *V programe musí byť použité zapuzdrenie.*

Projekt obsahuje zapuzdrenie a na prístup som využíval get a set metódy.

5. *Program musí obsahovať dostatok komentáru na pochopenie kódu.*

Program obsahuje komentáre k častiam, ktoré sa mi zdali komplikovanejšie ale inak mám javadoc ku každej jedenej metóde a triede.

6. *Dokumentácia musí zodpovedať programu a musí obsahovať diagram tried.*

7. *Pri záverečnej prezentácii študent musí vedieť zodpovedať otázky vyučujúceho v súvislosti s projektom.*



## Dodatočné kritéria

### Ošetrenie mimoriadnych stavov

Na ošetrenie mi postačili už vopred vytvorené výnimky, ktoré som používal v try-catch blokoch. Príkladom takejto výnimky je zadanie invalidného vstupu. Napríklad zadanie desatinné číslo do textového poľa, ktoré akceptuje iba celé čísla alebo ak by nastala chyba pri čítaní vstupných dát.

```
try {
    seller.loadCars(file);
    JOptionPane.showMessageDialog(this, "Data were successfully loaded.",
    "Success", JOptionPane.INFORMATION_MESSAGE);
}
catch (java.io.EOFException exp){
    JOptionPane.showMessageDialog(this, "Warehouse is completely
empty.", "Empty stock", JOptionPane.WARNING_MESSAGE);
}
catch (java.io.FileNotFoundException exp){
    JOptionPane.showMessageDialog(this, "The data file, 'cars.dat' doesn't
exist.\n\nThe new data file was created.", "Missing data file",
JOptionPane.ERROR_MESSAGE);
    seller.resetCars(file);
}
```

Na kóde vyššie je ukázaný try-catch blok, ktorý zachytáva chybu ak je súbor so vstupnými dátami prázdny alebo neexistuje. V oboch prípadoch sa objaví grafické okno s upozornením alebo varovaním, že sa niečo stalo so vstupnými dátami a tak musel byť vytvorený nový prázdny súbor pre dáta.

### Oddelenie aplikačnej časti od grafickej

V mojom prípade som mohol oddeliť iba pár tried do separátnych balíkov, nakoľko sa v mojom zadaní pracuje aj v triedach s GUI ale nie všetky triedy sa v GUI balíčku volajú samostatne. Práve preto som sa ich snažil pomenovať tak, aby bolo jasné, že Panel je trieda, ktorá sa volá v aplikačnej časti, ak môže a Layout, ktorý je volaný v triedach Panel.

### V hniezdenie tried

V hniezdenie tried sa vyskytuje v triede v triede ApplicationManager, ktorá ma v hniezdenú triedu Closer. Viac informácií sa nachádza vyššie v dokumentácii, konkrétne v sekcii Štruktúra aplikácie -> Vnútorne triedy

### Použitie serializácie

Serializácia sa používa v triedach Car Warehouse, nakoľko tieto metódy obsahujú dáta, ktoré sú uložené vo vstupných dátach. Na použitie serializácie som musel implementovať do každej triedy rozhranie Serializable.

```
public class Car implements java.io.Serializable
public class Warehouse implements java.io.Serializable
```

## Záver

Práca na tomto projekte ma náramne bavila lebo sa jednalo o môj prvý projekt, ktorý zahŕňal aj iné grafické prvky ako terminál. Taktiež som rád, že som sa mohol naučiť niečo o objektovo orientovanom programovaní a aspoň čiastočne pochopiť ich základ aj keď viem, že mám ešte toho dosť čo doháňať.