

Slovenská technická univerzita v Bratislave  
Fakulta Informatiky a Informačných Technológií

# TatrasPlaner

Objektovo Orientované Programovanie  
ak. rok 2022/23

## Abstrakt

Program TatrasPlaner naplánuje trasu podľa užívateľových preferencií, s ohľadom na jeho kondičnú zdatnosť a turistickú výbavu.

Dávid Truhlář

xtruhlar@stuba.sk

ID: 120897

## 1. Obsah

1. Obsah .....	2
2. Zadanie .....	3
a. Spresnenie rámcovej témy .....	3
3. TatraPlaner .....	4
a. Návod na používanie: [IDE : Eclipse / IntelliJ] .....	4
b. Aplikované OOP Princípy .....	6
I. Povinné .....	6
Dedenie .....	6
Polymorfizmus .....	6
Agregácia .....	6
Zapuzdrenie .....	6
II. Dobrovoľné .....	6
Návrhové vzory .....	6
Model-View-Controller .....	6
Ošetrenie mimoriadnych stavov .....	6
RTTI .....	6
Oddelenie aplikačnej logiky od používateľského rozhrania .....	7
Lambda výrazy .....	7
Viacnitosť .....	7
Vhniezdené triedy .....	7
AspectJ .....	7
Serializácia .....	7
4. Hierarchia tried .....	8
5. Dôležité commity na GitHub .....	9

## 2. Zadanie

### Plánovanie ciest

Cesty môžu predstavovať veľmi zložité podujatia, ktoré naozaj môžu mať prospech zo softvérovej podpory. Často zahŕňajú odbočky za účelom navštívenia zaujímavých miest v blízkosti hlavnej trasy. Rôzne služby a udalosti môžu byť ponúkané na týchto miestach. Tieto miesta môžu byť odporúčané na základe potrieb a ohraničení cestujúceho. Počas cesty sa pôvodný plán môže zmeniť. Cestovanie v skupine môže zahŕňať synchronizáciu odbočiek. Či plán cesty vytvoril profesionál alebo samotní cestujúci, môže byť hodný zdieľania s ostatnými.

#### a. Spresnenie rámcovej témy

##### TatrasPlaner

Projekt TatrasPlaner predstavuje cenného spoločníka pre každého turistu vysokohorských oblastí. Užívateľ si sám navolí, aký typ trasy hľadá, aké má vybavenie a fyzickú kondíciu, a na základe týchto preferencií mu TatrasPlaner poskytne detailný plán výletu.

TatrasPlaner obsahuje komplexný zoznam turistických chodníkov vo Vysokých Tatrách, vrátane všetkých chát, hotelov a zastávok s občerstvením. Poskytuje tiež podrobné informácie o jednotlivých chodníkoch, ako sú začiatok, koncová zastávka, prevýšenie, terén a celkový predpokladaný čas strávený na chodníku. Projekt zohľadňuje starostlivo posúdenú náročnosť jednotlivých chodníkov, aby užívateľa neposlal na trasu, na ktorú nie je pripravený z hľadiska fyzickej kondície alebo výbavy.

Projekt TatrasPlaner ponúka rôzne možnosti ubytovania v závislosti od finančných možností používateľa, vrátane možností hotela, privátu alebo horskej chaty. Okrem toho, berie do úvahy aj celkový čas, ktorý užívateľ plánuje stráviť v Tatrách, či už ide o výlet na jeden deň, víkend alebo celotýždňový pobyt.

Pretože náš cieľ je poskytnúť užívateľom komplexný zážitok z pobytu v Tatrách, TatrasPlaner sa snaží poskytnúť najlepšie rady pre plánovanie trás a ubytovania, aby bola návšteva tejto krásnej oblasti Slovenska nezabudnuteľná.

### 3. TatraPlaner

a. Návod na používanie: [IDE : Eclipse / IntelliJ]

b. Run

c. Po spustení sa zobrazí úvodné okno s prihlásením

Ak chceme otestovať niektorý z predpripravených účtov, zoznam je tu:

- a. Username: David Password: heslo [user]
- b. Username: Jan Password: heslo [horolezec]

Prípadne hore v záložkách zvolíme možnosť New user a zaregistrujeme sa.

d. Po (registrácii a) úspešnom prihlásení sa ocitneme v hlavnom okne programu

V tejto sekcii user zvolí odkiaľ a kam chce ísť, a aké zaujímavosti chce na trase absolvovať

Prihlásený user – ak by bol typu Horolezec za jeho meno by bolo v tvare @usernameC

Táto oblasť slúži na výpis výslednej trasy

## e. Ukážka

The image displays two screenshots of the 'TatrasPlanner!' application interface.

The top screenshot shows the 'Login' screen. It has a title bar with 'TatrasPlanner!' and window controls. Below the title bar are two tabs: 'Existing user' and 'New user'. The main content area is titled 'Login' and contains a text input field with the value 'David', a password input field with masked characters '....', and a 'Login' button. A mouse cursor is pointing at the 'Login' button.

The bottom screenshot shows the main application window. The title bar also says 'TatrasPlanner!'. The interface includes a logo on the left, a user profile section on the right showing '@ David' and an 'Odhlásiť' button, and two main sections: 'Začiatok' (Start) and 'Koniec' (End). Under 'Začiatok', there is a dropdown menu showing 'Smokovec' and checkboxes for 'Občerstvenie' (unchecked) and 'Vodopád' (checked). Under 'Koniec', there is a dropdown menu showing 'Zamkovského Chata' and checkboxes for 'Zlaňovanie' (unchecked) and 'Vlek' (checked). A blue box on the left contains the following text: 'Trasa: start: Smokovec, cieľ: Zamkovského Chata, čas: 132, vzdialenosť: 6.5, náročnosť: 1, zaujímavosti: Vlek, Vodopád'. At the bottom right, there is a blue button labeled 'Vyhľadať' (Search) with a mouse cursor pointing at it.

Užívateľ zvolí Začiatok a Koniec. Potom si vyberie, aké zaujímavosti chce na trase. Klikne na vyhľadať.

## b. Aplikované OOP Princípy

Zadanie určuje niektoré *oop* princípy ako povinné a niektoré ako dobrovoľné.

### I. Povinné

#### *Dedenie*

Trieda User tvorí základ pre triedu Horolezec.

```
../src/main/java/com/example/Pouzivatelia/Horolezec.java
../src/main/java/com/example/Pouzivatelia /User.java
```

#### *Polymorfizmus*

Ak je User Horolezec, výpočet času stráveného na Trase je kratší, pretože niektoré úseky dokáže zlaníť namiesto toho aby kráčať dlhšou cestou. Podobne na čas má vplyv aj kondícia User-a, čím horšia je kondícia, tým pomalšie Trasu prejde.

```
../src/main/java/com/example/Pouzivatelia/Horolezec.java
../src/main/java/com/example/Pouzivatelia /User.java
```

#### *Agregácia*

User aj Horolezec sú agregáty triedy Kondícia.

```
../src/main/java/com/example/Pouzivatelia/Horolezec.java
../src/main/java/com/example/Pouzivatelia/User.java
../src/main/java/com/example/Pouzivatelia/Kondícia.java
```

#### *Zapuzdrenie*

V triede User sú atribúty private a v triede sú definované getteri a setteri.

```
../src/main/java/com/example/Pouzivatelia/User.java
```

---

### II. Dobrovoľné

#### *Návrhové vzory*

##### Model-View-Controller

Napríklad login screen ale aj hlavné okno programu, kde View je vo fxml súboroch a logika je ošetrená v Controlleroch.

```
../src/main/java/com/example/MAIN/HelloController.java
../src/main/ressources/com/example/MAIN/hello-view.fxml
```

#### *Ošetrenie mimoriadnych stavov*

V triede HelloController, ak User nie je v databáze, program vyhodí exception.

```
../src/main/java/com/example/MAIN/HelloController.java
```

#### *RTTI*

V triede SuccesLogin ak prihlásený User spĺňa podmienku.

```
if (user instanceof Horolezec horolezec)
```

```
../src/main/java/com/example.MAIN.SuccesLogin.java
```

### *Oddelenie aplikačnej logiky od používateľského rozhrania*

HelloController je trieda, ktorá obsahuje funkcie, ktoré reagujú na užívateľské akcie ako napr. stlačenie Buttonu alebo CheckBoxu. GUI je definované a oddelené v samostatnom fxml súbore hello-view.fxml

```
../src/main/java/com/example/MAIN/HelloController.java  
../src/main/ressources/com/example/MAIN/hello-view.fxml
```

### *Lambda výrazy*

V Triede HelloController.java pri kontrole, či je User, ktorý sa snaží prihlásiť využívam lambda výraz, na vytiahnutie daného Usera z databázy.

```
.filter(user -> user.getUsername().equals(username) &&  
user.getPassword().equals(password))
```

```
../src/main/java/com/example/MAIN/HelloController.java
```

### *Viacnitosť*

V triede Tura v metóde combTrasy(), sa využívajú dve vlákna na prehľadanie a spojenie trás, ak majú spoločný bod.

```
../src/main/java/com/example/secondary/Tura.java
```

### *Vhniezdené triedy*

V triede Tura je vzhniezdená trieda Trasa.

```
../src/main/java/com/example/secondary/Tura.java
```

### *AspectJ*

Implementácia AspectJ, sa mi nepodarila spojazdniť na 100%, pretože rovnako v IntelliJ aj v Eclipse som mal problémy implementovať potrebné knižnice, ktorý by mi umožnili s ním pracovať. Po štúdiu literatúry a prednáške som však usúdil, že najlepšie použitie v mojom programe by bolo na generovanie error\_log – u. Súbor, ktorý by to ošetroval je súčasťou projektu, len je zakomentovaný.

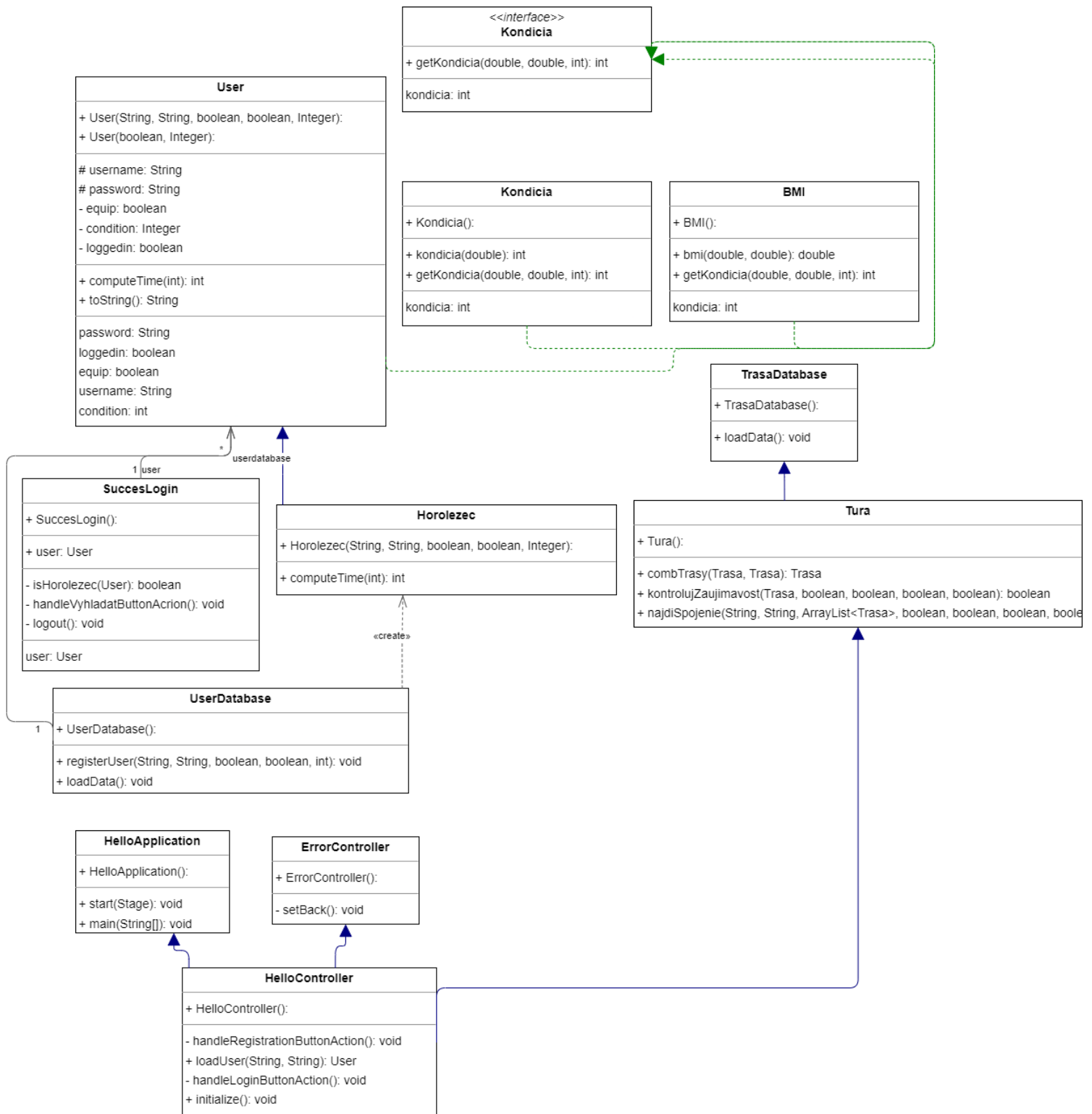
```
../src/main/java/com/example/MAIN/ErrorLoggerAspect
```

### *Serializácia*

Serializácia je využitá pri registrácii, keď sa zaregistruje nový užívateľ, triedu User pomocou metódy toString() program prepíše do Stringu a zapíše do 'uzivatelia.txt'.

```
../src/main/java/com/example/Pouzivatelia/User.java  
../src/main/java/com/example/MAIN/HelloController.java
```

## 4. Hierarchia tried





## 5. Dôležité commity na GitHub

	Commit	Dátum	Komentár
1	6b31ef6f7cf3acfa08902ea0f21ef4ebc6223689	9.3.2023	Vloženie Spresnenia rámcovej témy
2	19cde5b90b2e573bc56a171bed9d6256dee35bc7	23.3.2023	Vytvorenie Tried - základná hierarchia
3	c9769c5ebeab682d041f92aa2375c29b279c3c93	1.4.2023	Vytvorenie a pridanie login logiky
4	c9769c5ebeab682d041f92aa2375c29b279c3c93	2.4.2023	Databáza trás + Databáza userov
5	96959b6d42efd6965b925db749d5a9daeda35187	3.4.2023	Kombinácia dvoch trás do túry
6	dc66ab1f4581d93efcc74f9a9d740164763f38d0	4.4.2023	Spájanie trás na podnet užívateľa
7	544fc5bee060f132ef199591b34984dd43020cce	4.4.2023	Spojenie Loginu s databázou a GUI
8	babe5e489421326ad6baf0cc000cff21a65bc49f	5.4.2023	Pridaný log out, Scény - main window
9	f64435863039819c75b6fbe14285f9e235f3e900	6.4.2023	Ošetrovanie dedenia - Trieda Horolezec
10	47266a7e704ef02fbf55956ba820329c7a334ff2	10.4.2023	Pracovná verzia + readme
11	114e3892da076a16cf8d8456a38c6ecfd7820f30	11.4.2023	Vytvorená dokumentácia
12	a59b85e60a10cf45fa9813034a8e85eee8f100d1	21.4.2023	Vytvorenie JavaDoc kometárov
13	42d434f4735c28dddb0fa10b9a9bd38d3ae992bf	24.4.2023	Lambda, exceprions
14	22b9d970d38143b30639acf2d8419f7d72af5108	24.4.2023	Viacnitovosť
15	ae33c363ac14c67f5330bee4095d14f3766dd62e	24.4.2023	Diagram Tried
16	98cae735a8235cd3b491026372b34c3d5829a3c5	24.4.2023	Významné commity