

**UNIVERZITA MATEJA BELA V BANSKEJ BYSTRICI
FAKULTA PRÍRODNÝCH VIED**

**ZOBRAZENIE A RIEŠENIE PROBLÉMU OBCHODNÉHO
CESTUJÚCEHO NA MAPE SLOVENSKA**

Systémová dokumentácia

Banská Bystrica, 2023

Michal Mojžiš

Zdrojové kódy

Zdrojové kódy aplikácie sú dostupné pre:

- serverovú časť na adrese: <https://github.com/misko3001/TSP-na-Slovensku-be>
- prezentačnú časť na adrese: <https://github.com/misko3001/TSP-na-Slovensku-rcapp>

Požiadavky na spustenie aplikácie

Pre optimálny beh aplikácie je potrebné spĺňať nasledujúce požiadavky:

- JRE 17 (Java Runtime Environment)
- Operačná pamäť: aspoň 4GB
- Pevný disk: 150 MB voľného miesta

V prípade, že je v počítači nainštalovaných viacej JRE je potrebné zaistiť, aby príkaz java odkazoval na verziu JDK 17. V operačných systémoch Windows je možné nakonfigurovať systémovú premennú JAVA_HOME. Alternatívne na ľubovoľnom operačnom systéme je možné sa definovať celú cestu adresára, kde je nainštalovaný JRE 17 a spustiť program napríklad pomocou príkazu:

```
<cesta-k-java>\jdk\corretto-17.0.7\bin\java -jar <cesta-k-suboru>\slovak_tsp-1.0.0.jar
```

Programové prostredie

Serverová časť aplikácie je postavená pomocou JDK 17 a využíva framework Spring Boot 2.7.3. Aplikácia využíva klasickú MVC architektúru a H2 databázu na potreby dátovej vrstvy. Po skompilovaní je výsledkom jeden .jar balíček, ktorý obsahuje obe aplikácie, a teda na spozajzdnenie celej aplikácie stačí skompilovať len serverovú časť. Výsledný balíček je možné spúšťať so štandardnými Spring Boot prepínačmi. Na vývoj boli použité nasledujúce knižnice:

- spring-boot-starter-web
- spring-boot-starter-data-jpa
- spring-boot-starter-websocket
- spring-boot-starter-validation
- spring-boot-starter-test

- flyway-mysql
- h2 (2.1.214)
- jenetics (7.1.2)
- jpx (3.0.1)
- lombok
- mapstruct (1.5.3.Final)
- mapstruct-processor (1.5.3.Final)

Prezentačná vrstva aplikácie je postavená pomocou framework-u Angular 13.3.0. Táto aplikácia je sprostredkovaná pomocou serverovej časti, v ktorej sa priamo nachádzajú skompilované zdrojové kódy, a tak je potrebné spúšťať iba jednu aplikáciu. Na vývoj boli použité nasledujúce knižnice:

- @angular/animations
- @angular/cdk
- @angular/common
- @angular/compiler
- @angular/core
- @angular/forms
- @angular/localize
- @angular/material
- @angular/platform-browser
- @angular/platform-browser-dynamic
- @angular/router
- @fortawesome/fontawesome-free (6.0.0)
- @ng-bootstrap/ng-bootstrap (12.1.2)
- @popperjs/core (2.10.2)
- @stomp/rx-stomp (2.0.0)
- bootstrap (5.1.3)
- chart.js (3.1.1)
- mdb-angular-ui-kit (2.3.0)
- ol (7.1.0)
- primeng (13.4.1)

- rxjs (7.5.0)
- sweetalert2 (11.6.8)
- tslib (2.3.0)
- zone.js (0.11.4)

Pridávanie nových funkcionáľít

Do aplikácie je možné pridávať vlastné funkcionality, ako napríklad nové implementácie selekcie, mutácie alebo kríženia. Ak by niekto chcel napríklad pridať nový typ mutácie, je potrebné vykonať nasledovné kroky:

1. Vytvoriť novú triedu, ktorá bude obsahovať logiku mutácie.

```
public class NovaMutacia<G extends Gene<?, G>, C extends Comparable<? super C>>
    extends Mutator<G, C> {

    public NovaMutacia(final double probability) {
        super(probability);
    }

    @Override
    protected MutatorResult<Chromosome<G>> mutate(final Chromosome<G> chromosome,
                                                    final double p,
                                                    final RandomGenerator random) {
        // SEM PRIDAŤ VLASTNÚ IMPLEMENTÁCIU MUTÁCIE, KTORÁ UPRAVÍ
        // CHROMOZÓM V PARAMETROCH TEJTO METÓDY
    }
}
```

2. Zaregistrovať túto možnosť v číselníku TSPMutator v oboch aplikáciách.
3. Pridať do switch v EngineFactory možnosť pre nový TSPMutator.
4. V aplikácii je následne možné používať novo vytvorenú mutáciu.

Pridávanie nových operátorov selekcie a kríženia je založené na podobnom postupe.