

# Alternativní semestrální zadání pro KIV/PT 2023/2024

Leonid MALAKHOV A22B0387P

Faiz SULEIMANOV A22B0392P

27.11.2023

## Obsah

<b>1</b>	<b>Popis zadání</b>	<b>2</b>
<b>2</b>	<b>Analýza zadání</b>	<b>2</b>
<b>3</b>	<b>Analýza problému a navrhovaná řešení</b>	<b>2</b>
3.1	Definice problému . . . . .	2
3.2	Možná řešení . . . . .	2
3.3	Zvolené řešení a zdůvodnění . . . . .	3
<b>4</b>	<b>UML diagram</b>	<b>4</b>
<b>5</b>	<b>Uživatelská dokumentace</b>	<b>4</b>
5.1	Požadavky pro spuštění . . . . .	4
5.2	Jak spustit program . . . . .	5
5.3	Popis vstupních argumentů . . . . .	5
<b>6</b>	<b>Závěr a zhodnocení</b>	<b>5</b>
<b>7</b>	<b>Rozdělení práce</b>	<b>5</b>
7.1	Faiz Suleimanov . . . . .	6
7.2	Leonid Malakhov . . . . .	6
7.3	Společná práce . . . . .	6

# 1 Popis zadání

Zadání semestrální práce pro předmět KIV/PT 2023/2024 vyžaduje vytvoření simulačního programu pro společnost Sto roků v šachtě zíl s. r. o., která se specializuje na dopravu uhlí. Majitel společnosti, Petr Beznoha, čelí finančním ztrátám způsobeným nepříznivým počasím, které poškodilo kovová kolečka používaná pro přepravu. Cílem je vytvořit program, který pomůže plánovat přepravu uhlí k zákazníkům tak, aby se minimalizovaly další ztráty, dodržely závazky a udržela klientela.

## 2 Analýza zadání

Program bude simulovat přepravní proces, přičemž bude zohledňovat počet skladů, polohu skladů a zákazníků, typy a nosnost koleček, a také počasí a podmínky nutné pro údržbu transportních prostředků. Požadavky zahrnují efektivní využití kapacity koleček a optimalizaci tras tak, aby se minimalizovala "cena" přepravy, tj. náklady spojené s provozem a údržbou. Hlavním cílem je tedy vytvořit efektivní a spolehlivý systém pro plánování logistiky, který zohlední všechny proměnné a omezí rizika poškození přepravních prostředků.

## 3 Analýza problému a navrhovaná řešení

### 3.1 Definice problému

Společnost Sto roků v šachtě zíl s. r. o. se potýká s logistickým problémem efektivní přepravy uhlí od skladů k zákazníkům. Hlavními výzvami jsou optimalizace trasy, minimalizace nákladů na údržbu přepravních prostředků a dodržování dodacích lhůt. Poškození přepravních prostředků vlivem nepříznivého počasí vede k finančním ztrátám a ohrožuje spokojenost zákazníků.

### 3.2 Možná řešení

Bylo zváženo několik algoritmů pro hledání nejkratší cesty v grafu, včetně:

- Algoritmus Dijkstry
- A\* algoritmus
- Bellman-Fordův algoritmus

Algoritmus Dijkstry byl vybrán pro jeho efektivitu v hledání nejkratší cesty v grafu s nezápornými váhami hran, což je vhodné pro náš případ, kde váhy představují vzdálenost nebo čas.

### 3.3 Zvolené řešení a zdůvodnění

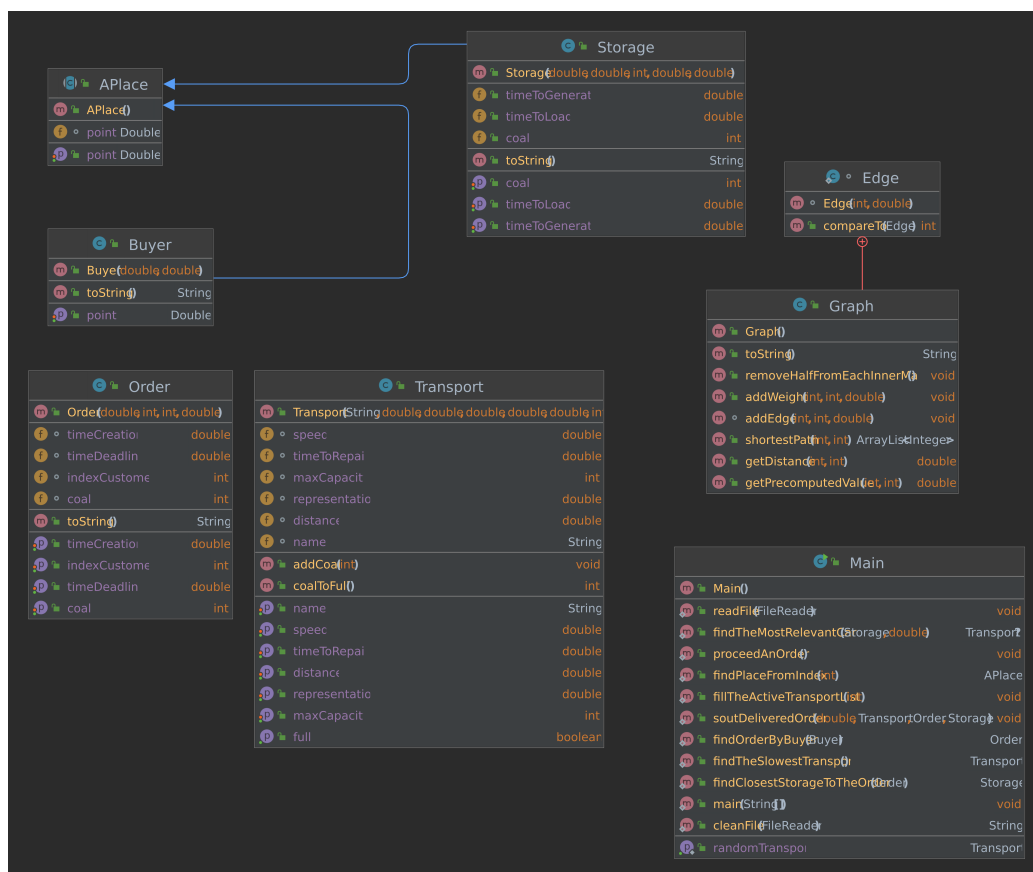
Pro řešení problému byl zvolen algoritmus Dijkstry. Tento algoritmus je ideální pro naše potřeby, protože poskytuje nejefektivnější způsob výpočtu nejkratších cest od jednoho uzlu (skladu) ke všem ostatním (zákazníkům), což je klíčové pro plánování logistiky. K ukládání grafu a příslušných dat bude použito dvou hashmap:

1. První hashmapa bude obsahovat seznam svázaných vrcholů, což umožní rychlé a efektivní vyhledávání sousedních skladů nebo zákazníků.
2. Druhá hashmapa bude sloužit jako cache pro již vypočítané cesty, což znamená, že když bude třeba znovu vypočítat trasu, která již byla v minulosti vypočítána, můžeme výsledek získat mnohem rychleji.

Tato kombinace nejenže zvýší efektivitu výpočtu, ale také sníží celkovou výpočetní náročnost, což je nezbytné pro rychlou reakci na dynamické změny v požadavcích zákazníků a stavu vozového parku.

Na začátku práce algoritmus vygeneruje náhodné kolečko, pak zkusí dodat první objednávku z fronty objednávek. Pak spočte, kolik se potřebuje auto pro vyřešení toho zadání. Auta pracují nezávisle mezi sebou. Pro každou objednávku bude vybráno volné auto, které je nejbližší k místu dodání. Pak to auto pojedí nejkratší cestou. Algoritmus spočte všechno tak, aby pracovalo co nejméně koleček.

## 4 UML diagram



Obrázek 1: UML diagram projektu

## 5 Uživatelská dokumentace

### 5.1 Požadavky pro spuštění

Pro spuštění zkompilevaného Java projektu je potřeba mít nainstalovanou Java Runtime Environment (JRE) nebo Java Development Kit (JDK) verze 19 nebo vyšší. Ujistěte se, že příkaz ‘java’ je dostupný v příkazové řádce nebo terminálu ve vašem operačním systému. Toto zajistí, že budete moci spustit programy zkompilevané pro Java 19.

## 5.2 Jak spustit program

Program lze spustit otevřením příkazové řádky nebo terminálu a navigací do složky, která obsahuje soubor JAR vašeho projektu. Poté spusťte program následujícím příkazem:

```
java -jar pt_semestral.jar [argumenty]
```

## 5.3 Popis vstupních argumentů

Program přijímá následující vstupní argumenty:

- **-cR** (volitelný): Příznak, který určuje, že program by měl použít aktuální adresář pro hledání vstupního souboru. Pokud je tento příznak použit a vstupní soubor se nachází ve stejné složce jako spustitelný JAR soubor programu, stačí zadat pouze název vstupního souboru bez cesty.
- **-f [soubor]**: Specifikuje cestu k vstupnímu souboru, který bude program zpracovávat. Tento argument je povinný, pokud není nastaven příznak **-cR**.

Příklad použití argumentů:

```
java -jar nazev_souboru.jar -f cesta/k/vstupnimu/souboru.txt
```

Pokud jsou argumenty zadány nesprávně nebo chybí, program vypíše chybové hlášení a návod k použití.

## 6 Závěr a zhodnocení

Na konce, práce není ve finálním stavu, kvůli nedostatku času a prvnímu termínu odevzdání. Chtěli bychom věnovat práci ještě cca 10 hodin. Zadání je splněno s podmínkami, některý soubory ne fungují. Mnoho hodin bylo ztraceno kvůli nepochopení zadání práce.

## 7 Rozdělení práce

Práce na projektu byla rozdělena mezi členy týmu podle jejich specializace a dovedností. V následujících odstavcích jsou popsány úkoly, které každý člen týmu vykonával.

## **7.1 Faiz Suleimanov**

Faiz Suleimanov se specializoval na vývoj a optimalizaci algoritmu pro hledání nejkratší cesty. Jeho práce zahrnovala nejen implementaci a testování algoritmu, ale také jeho neustálé zlepšování pro zajištění maximální efektivity a spolehlivosti. Kromě toho se Faiz věnoval návrhu a implementaci reprezentace grafu, včetně ukládání vrcholů, hran a nalezených cest, což bylo klíčové pro efektivní plánování logistiky.

## **7.2 Leonid Malakhov**

Leonid Malakhov měl na starosti vývoj řešení pro hlavní logistické problémy spojené s kolečky a požadavky. Tato práce vyžadovala hluboké porozumění logistického procesu a byla zaměřena na vytvoření robustního systému, který by řešil specifické potřeby firmy v oblasti dopravy a dodávek.

## **7.3 Společná práce**

Celková struktura programu a implementace čtení vstupních dat byla provedena společnými silami. Týmová spolupráce byla klíčová pro synchronizaci jednotlivých částí projektu a zajištění, že všechny komponenty systému pracují spolehlivě a efektivně. Vzájemná komunikace a pravidelné kontroly kódu byly nezbytné pro úspěšné dokončení projektu.