**Semestrální práce**

**KIV/PT**

Vypracovali: David Muhr a Jan Pelikán

Obsah

[Zadání 3](#_Toc58531053)

[Analýza problému 3](#_Toc58531054)

[Řešení 3](#_Toc58531055)

[Návrh programu 3](#_Toc58531056)

[Uživatelská dokumentace 3](#_Toc58531057)

[Závěr 4](#_Toc58531058)

# Zadání

Zadáním semestrální práce bylo vytvořit program, který by minimalizoval náklady pro přepravu zboží. Zboží se přepravuje mezi továrnami a supermarkety.

Celé zadání je přiložené v .zip souboru. (**zadani.pdf**)

# Analýza problému

Po přečtení zadání jde vidět, že se bude jednat o grafově založenou úlohu. Cílem bylo zjistit, jak správně rozvážet zboží aby byla celková cena co nejmenší. Zadané jsou ceny převozu mezi supermarkety a továrnami. Ze kterých lze odvodit, že chceme dovážet zboží do supermarketu z nejlevnějších továren. Některá data jsou obrovská, tudíž bude potřeba algoritmus který by měl i dobrou složitost.

## Řešení

Nakonec jsme problém vyřešili dovolím si říct selským rozumem, který funguje dostatečně dobře. Celá simulace se skládá z hlavních tří *for* cyklů, první cyklus protáčí počet dní, druhý cyklus je pro supermarkety a třetí a poslední je pro typy zboží. Dále zjistíme, kolik zboží potřebujeme dostat do supermarketu. Zjistíme si, jestli zásoby ze supermarketu stačili. Pokud ne, tak zjistíme nejlevnější továrnu pro daný supermarket a převezeme zboží. Pokud zboží stále nestačilo, hledáme další továrny. Jakmile supermarket potřebuje zboží z továrny, ale žádné továrny už zboží nemají, simulace se ukončí a vypíše se, že zboží je potřeba dokoupit z Číny. Pro načtení dat, jsme použili ArrayListy a z nich poté data přesunuli do čtyř matic podle zadání.

Data generátory jsme udělali nakonec jen dva, a to pomocí rovnoměrného rozložení a Gaussovského rozložení. Generování čísel bylo celé pseudonáhodné a byla využita třída Java.util.Random. Pro vkládání dat, byli využity for cykly, podle rozměru matice, dva nebo tři. Výpis matic probíhal úplně stejně, akorát že se hodnota z matice vypisovala.

# Návrh programu

Pro celý program jsme se rozhodli udělat jednoduché grafické rozhraní pomocí knihovny Swing. Okno je viditelně rozdělené na dvě strany, levou a pravou. Levá strana funguje pro simulování, nejdříve si vyberete soubor, ze kterého chcete načítat data. Můžete si zvolit vlastní délku simulace (funguje je pro zkrácení). A program poté vytiskne finální cenu. Pravá strana okna je pro generování náhodných dat. Po zadání povinných polí, se podle vašich specifikací vygeneruje vámi zvolený druh dat. O vykreslení okna se stará třída MainMenu, tato třída potom přistupuje k ostatním třídám, a to podle požadavku. Buďto k třídě Simulace, nebo k různým třídám generátorů. Okno pro práci s programem jsme se snažili udělat jednoduché a přehledné, aby nebylo těžké ho obsluhovat.

# Uživatelská dokumentace

Program se spouští z příkazové řádky, hlavní třída je MainMenu, která zároveň obsahuje metodu main(). Po spuštění se otevře jednoduché okno, které dává na výběr mezi generováním a spuštěním simulace. Pro spuštění simulace stačí vybrat zdrojový soubor dat, pro které chceme simulovat. Poté změnit počet dní simulace pokud chceme. Stisknutím tlačítka simulovat započne simulace. Po doběhnutí se vypíše finální cena a vygenerují se dva soubory: simulace.txt a statistiky.txt. V souboru se simulací je popsaná detailně simulace a v souboru statistiky jsou vypsané požadované statistiky.

Pokud zrovna nemáte náladu kazit si náladu tím, že zjistíte, kolik Vás bude stát další rozvoz zboží. Můžete si generovat pseudonáhodná čísla pomocí dvou druhů generování. Stačí vyplnit základní údaje typu počet továren, supermarketů nebo minimální a maximální hranice generovaných dat. Dále si můžete zvolit z dvou druhů. Rovnoměrné generování a Gaussovské generování. Rovnoměrné generování pevně dodrží vaše stanovené hranice a bude vybírat náhodně čísla v zadaném rozmezí. Gaussovské rozložení může hranice překročit, ale je udělané tak, aby bylo co nejvíce hodnot v zadaném rozmezí.

# Závěr

Byla to naše první semestrální práce, kterou jsme měli dělat v týmech a ne samostatně. Osobně mi to takhle vyhovovalo víc, když jeden z nás něco nevěděl nebo tomu nerozuměl tak mu to druhý vysvětlil nebo to prostě udělal za něj. Komunikace mezi námi probíhala dobře, používali jsme pro domluvu Discord. (Také pro verzování :)) ) Zadání bylo na první pohled jednoduché a čekal jsem, že to bude lehké. Ale opak byl pravdou, když jsme měli vymyslet kostru programu, tak jsme strávili pár hodin na discordu pouze tím, abychom všechno pochopili. Práce ve dvojici nám obou vyhovovala. Celkově bych výstup hodnotil kladně, líbí se mi naše grafické rozhraní.