

# Kalkulačka matic

Ondřej Jakš

## Anotace

Úkol projektu byl vytvořit kalkulačku pro numerické matice. Kalkulačka řeší sčítání, odčítání, násobení matic, determinant matice, inverzní matice a regulovaný odstupňovaný tvar matice. Pro matice velikosti minimálně  $2 \times 2$  a maximálně  $10 \times 10$ .

# Obsah

1	Uživatelská část .....	4
1.1	Potřebný software .....	4
1.2	Spuštění .....	4
1.3	Velikost matice .....	4
1.4	Vložení čísel matice .....	4
1.5	Reprezentace výsledků .....	4
2	Programátorská část .....	4
2.1	Použití knihovny tkinter .....	4
2.2	Reprezentace dat matice .....	4
2.3	Operace s maticemi .....	4
2.3.1	Sčítání a odčítání .....	4
2.3.2	Násobení .....	5
2.3.3	Determinant .....	5
2.3.4	Inverzní matice .....	5
2.3.5	RREF .....	5
3	Závěr .....	6

# 1 Uživatelská část

## 1.1 Potřebný software

Pro spuštění programu je nutné mít nainstalovaný python a k tomu knihovnu Tkinter, díky níž vytváříme GUI programu.

## 1.2 Spuštění

Program se spustí otevřením souboru main.py. Po spuštění se zobrazí okno s nápis „Numeric matrix calculator“ a s šesti tlačítky, které slouží k výběru operace, kterou chce uživatel aplikovat.

## 1.3 Velikost matice

Po tom, co si uživatel vybere, co chce s maticemi dělat musí zadat velikost matice. Pro různé operace či funkce může být nějaký typ omezení jako například: nelze nastavit počet řádků u druhů matice při násobení, jelikož musí sedět s počtem sloupců první matice atp. Následně zmáčkne tlačítko „Enter“.

## 1.4 Vložení čísel matice

Následně se objeví okno, kde bude počet vstupů, které uživatel chtěl. Uživatel musí zadat do každého vstupu příslušné číslo. Potom musí zmáčknout „Enter“. Toto se bude opakovat pro operace sčítání, odčítání a násobení, aby uživatel zadal druhou matici.

## 1.5 Reprezentace výsledků

Po provedení operace se zobrazí okno, kde bude zapsán výsledek operace. Následně uživatel zavře okno s výstupem a může pokračovat s dalšími operacemi nebo také křížkem zavřít i hlavní menu.

# 2 Programátorská část

## 2.1 Použití knihovny tkinteru

Pro vytvoření oken a interaktivních grafických prvků jsem využil knihovnu tkinter. Metoda Button byla využita k tvorbě tlačítek, metoda Entry pro vstupní data a Label pro text.

## 2.2 Reprezentace dat matice

Reprezentace matic v programu je list, který obsahuje listy. Vnitřní listy jsou reprezentace řádků matice.

## 2.3 Operace s maticemi

### 2.3.1 Sčítání a odčítání

Operace sčítání a odčítání matic jsou implementovány jako dva cykly, které provede operaci mezi prvky matic na stejné pozici.

Následně pokud nemá prvek desetinný rozvoj, tak program daný prvek převede na celé číslo.

### 2.3.2 Násobení

Násobení je implementováno jako tři cykly, které znásobí prvky daného sloupce s prvky daného řádku, potom je sečte a tím vytvoříme prvek na určité pozici. Takto vytvoří celou matici.

Pokud nemá určitý prvek desetinný rozvoj, tak program daný prvek převede na celé číslo.

### 2.3.3 Determinant

Implementace je řešená rekurzivně. Nejdříve tato implementace zjistí, jestli matice nemá velikost  $2 \times 2$ . Pokud má tuto velikost, tak vrátí vynásobení diagonály a odečteme vynásobenou diagonálu druhou.

Pokud dostane matici větší než  $2 \times 2$ , tak algoritmus vytvoří determinanty pro malé matice pomocí rekurze a odečítá nebo přičítá tyto determinanty. Tímto zjistí determinant celé matice. Tato část algoritmu se rekurzivně zanořuje se do doby, než velikost zrovna počítané matice bude  $2 \times 2$ . Je to implementování adjungování matice.

### 2.3.4 Inverzní matice

Implementace nejdříve zkontroluje, aby neměla matice nulový determinant.

Pokud není determinant nula, tak vyřešíme nejdříve pro matice  $2 \times 2$ , kde přehodíme první a poslední prvek a vydělíme je determinantem a druhý a třetí prvek vydělíme determinantem a vynásobíme  $-1$ .

Pro matice větší, než  $2 \times 2$  nejdříve upravíme na matici kofaktorů následně vypočteme  $-1^{i+j} * \det(\text{Minor})$  a to přidáme jej do listu a následně tento list přidáme do dalšího listu.

Tento list zkopírujeme do jiné proměnné. Následně každý prvek vydělíme determinantem. Následně se čísla bez desetinného rozvoje přepíší na celá čísla a čísla s desetinným rozvojem se přepíší na čísla s rozvojem zaokrouhleným na dva prvky desetinného rozvoje.

### 2.3.5 RREF

Implementace používá Gauss-Jordanovu eliminaci. Nejdříve se spustí metoda RREF, která nejdříve zkontroluje, jestli nejsou na diagonále nuly, pokud takovou nulu nalezne přehodí tento řádek s následujícím řádkem, nebo pokud již nemá co prohodit, tak vyskočí z cyklu. Pokud nenalezne nulu, tak tak prvky řádku před diagonálou vydělí diagonálou, tuto hodnotu uloží a následně, kde od prvku

odečte uloženou hodnotu vynásobenou prvky matice. Takto získáme upravenou matici.

Následně metoda `GJ_elimination` upraví matici na co nejjednodušší tvar, tudíž aby na diagonále byly 1.

Hodnoty, které nemají desetinný rozvoj, tak se přepíše na celá čísla a ty které jej mají, tak se přepíše na zaokrouhlená čísla na dva prvky desetinného rozvoje.

### 3 Závěr

Aplikace se celkem povedla. Dle mého názoru má jednoduché použití a je snadno rozšiřitelná o další operace s maticemi. Není to jednoduchý, ale ani složitý projekt. Také je velmi užitečný pro pochopení některých věcí z Lineární algebry.