**Maticový kalkulátor**

**Richard Klíma, 2. ročník**

**Zimní semestr 2017/2018**

**Programování I**

## Stručné zadání

Program nabízí maticový kalkulátor s grafickým uživatelským rozhraním.

## Přesné zadání

Maticový kalkulátor pracuje s maticemi, jejichž prvky jsou racionální čísla v desetinném i zlomkovém zápisu. Pomocí maticového kalkulátoru je možné matice mezi sebou sčítat, odčítat a násobit, je-li to možné. Jednotlivé matice lze také násobit racionální konstantou, transponovat, převádět do Gaussova tvaru a počítat jejich determinant. S výslednými maticemi lze dále pracovat nebo je exportovat do textového souboru.

Maticový kalkulátor také umožňuje počítat soustavy lineárních rovnic až o sedmi neznámých, pokud mají právě jedno řešení. Uživatel zadá matici soustavy, a pokud má soustava právě jedno řešení, maticový kalkulátor ho najde. Pokud má soustava nekonečně mnoho řešení, program uživateli oznámí, že soustava nemá jednoznačné řešení. Pokud soustava řešení nemá, program tuto skutečnost uživateli rovněž oznámí.

## Zvolený algoritmus

Algoritmy použité v implementaci maticového kalkulátoru jsou přímočaré algoritmy pro provádění základních operací se zlomky a maticemi. Tyto algoritmy zahrnují např. sčítání zlomků, násobení zlomků, krácení zlomků na základní tvar, převod matice do Gaussova tvaru, vyhodnocení matice soustavy lineárních rovnic, atd.

## Diskuze výběru algoritmu

Algoritmy použité v programu jsou standardně užívané algoritmy pro práci s maticemi. Algoritmy pro práci se zlomky jsou navrženy tak, aby práce se zlomky byla co nejpřímočařejší a nejjednodušší.

## Program

Hlavní jednotkou je formulářová jednotka Unit1. Ta se stará o běh formuláře, o jeho vzhled a změny vzhledu, o načítání vstupních dat od uživatele a z textových souborů, výpis výstupních dat do určených míst na formuláři, exportování výstupních dat do textových souborů a zobrazování zpráv určených uživateli. Její funkcí je také volání metod ostatních jednotek. Nejdůležitějšími funkcemi této jednotky jsou:

* Button1Click – načtení matice z formulářového okna pro zadávání matic,
* Button3Click – načtení matice z textového souboru,
* Button14Click – exportování výsledné matice do textového souboru,
* Button15Click – volání Gaussovy metody v jednotce UMatice.

Další jednotkou v programu je jednotka UMatice. Ta se stará o práci s maticemi jako takovými, tedy o sčítání, odčítání, násobení a transpozici matic, převod do Gaussova tvaru, počítání determinantu a vyhodnocování matice soustavy lineárních rovnic. Jejími nejdůležitějšími funkcemi jsou:

* Secti(TMatice, var TMatice), Odecti(TMatice, var TMatice), Vynasob(TMatice, var TMatice) – sečte, resp. odečte, resp. vynásobí dvě matice mezi s sebou,
* NasobKonstantou(string) – vynásobí matici konstantou,
* Transponuj – transponuje matici,
* Gauss(var boolean, var TMatice), Gauss (var TMatice) – převede danou matici do Gaussova tvaru, pokud je to možné; varianta s referenčním parametrem typu boolean navíc vrátí informaci, jestli je potřeba změnit znaménko při počítání determinantu matice (operace prohození řádků při počítání determinantu mění jeho znaménko),
* VyhodnoceniGauss(var TDynPole) – vyhodnotí matici soustavy lineárních rovnic (převede matici do Gaussova tvaru a od posledního řádku určuje řešení) a vrací vektor řešení,
* Determinant(var TZlomek) – spočítá determinant matice (převede matici do Gaussova tvaru a vynásobí spolu prvky na hlavní diagonále, popř. podle informace z metody Gauss(var boolean, var TMatice) změní znaménko determinantu).

Všechny tyto funkce vrací hodnotu typu boolean, která říká, jestli daná funkce proběhla v pořádku, nebo jestli během jejího vyhodnocování nastal nějaký speciální případ a operace tak nemohla proběhnout (např. pokud matice, ze které uživatel chce počítat determinant, není čtvercová, funkce Determinant(var TZlomek) vrátí false).

Dále tato jednotka obsahuje pomocné funkce na hledání maximální hodnoty ve sloupci matice, prohazování řádků, apod.

Poslední jednotkou používanou v programu je jednotka UZlomky. Ta zajišťuje práci se zlomky ve tvaru , kde , konkrétně např. o krácení, sčítání, odčítání a násobení zlomků. Nejdůležitějšími funkcemi této jednotky jsou:

* Zkrat – zkrátí zlomek na jeho základní tvar (tj. jsou nesoudělná čísla),
* Secti(TZlomek), Odecti(TZlomek), Nasob(TZlomek) – sečte, resp. odečte, reps. vynásobí vlastní zlomek se zlomkem předaným v parametru,
* Vysledek – vrátí hodnotu vlastního zlomku jako reálné číslo.

Dále tato jednotka obsahuje pomocné funkce pro práci se zlomky, např. převrácení hodnoty zlomku, určení absolutní hodnoty zlomku, porovnání dvou zlomků, apod.

## Alternativní programová řešení

Algoritmy použité v implementaci jsou co nejpřímočařejší. Některé algoritmy by pravděpodobně šlo nahradit výkonnějšími, nicméně nepřispělo by to k přehlednosti zdrojového kódu.

Funkce formulářové jednotky Unit1 mají defaultní názvy, což trochu znepříjemňuje čtení kódu, nicméně u každé z těchto funkcí je stručný komentář, k čemu daná funkce slouží, aby bylo na první pohled jasné, o jakou funkci se jedná.

## Reprezentace vstupních dat a jejich příprava

Uživatel komunikuje s programem pomocí tlačítkového formuláře, kde každé tlačítko obsahuje popis své funkcionality. Vyskytují se také přepínače, které slouží k tomu, aby bylo jednoznačně určeno, se kterou maticí se má pracovat. Zadávat matice může uživatel přímo ve formuláři prostřednictvím připravené mřížky (která je omezená maximálními rozměry ) nebo načtením matice z textového souboru, ve kterém je na prvním řádku počet řádků a počet sloupců matice oddělený mezerami a na dalších řádcích jsou po řádcích vypsané prvky matice daného rozměru. Pokud vstupní soubor není ve správném formátu, program uživatele uvědomí a načtení neproběhne. Matici soustavy lineárních rovnic uživatel zadává přímo ve formuláři do připravené mřížky, jejíž velikost si před tím sám nastaví.

Program nabízí také lištové menu v horní části formuláře, které ale poskytuje stejné možnosti jako tlačítkový formulář.

## Reprezentace výstupních dat a jejich příprava

Výstupní data jsou graficky reprezentována ve formuláři, kde uživatel vidí výsledky svých operací s maticemi a vyhodnocování soustav rovnic v připravených mřížkách. Uživatel také může výsledek svých operací a vyhodnocování soustav rovnic exportovat do výstupního textového souboru, který má stejný formát jako platný vstupní soubor programu a může tak být použit pro další načtení matice ze souboru.

## Průběh práce

Práce na programu probíhala relativně hladce, z počátku nejvíce času zabral grafický návrh formuláře, posléze byly nejnáročnějšími funkcemi na implementaci funkce pro převod do Gaussova tvaru a vyhodnocení matice soustavy lineárních rovnic. Jednotka pro práci se zlomky byla napsána samostatně již dříve pro jiný můj program, a tedy mi přišlo výhodné ji využít i v tomto programu (proto se v jednotce vyskytují některé funkce a procedury, které nejsou nikde v programu použity).

## Co nebylo doděláno

Nebylo doděláno lepší počítání soustav lineárních rovnic, kdy při nekonečném počtu řešení program pouze vypisuje informaci o tomto stavu, přičemž by mohl soustavu vyřešit parametrizací nějakých proměnných a vrátit vektor výsledků.

Za zamyšlení by stála také historie provedených operací a s tím se vázající funkce Undo a Redo.

Dalším nápadem, který by šlo realizovat by byla jakási příkazová řádka, kde by uživatel mohl po načtení matic (ne nutně nejvýše dvou zároveň – také by šlo dodělat) zadávat příkazy, které by s maticemi chtěl provést, např. ve tvaru ((A - B) \* (1/3) \* C) – (D \* (E + I\_6)) a tyto operace by se následně provedly. Jednalo by se tedy o připsání nějakého interpreteru pro tuto příkazovou řádku.