

# FRACTIONS 2.0

$$\frac{\frac{1}{2} + \frac{3}{2}}{\frac{123}{456} \frac{456}{123}} = \underline{\underline{2}}$$

written by Michal Hrušecký

17. listopadu 2004

# Obsah

<b>I</b>	<b>Uživatelský manuál</b>	<b>3</b>
<b>1</b>	<b>Slovo úvodem</b>	<b>4</b>
<b>2</b>	<b>Instalace</b>	<b>5</b>
2.1	Požadavky . . . . .	5
2.2	Instalace (MS Windows) . . . . .	5
2.2.1	Vyvolání příkazové řádky . . . . .	5
2.3	Instalace (MS-DOS) . . . . .	7
2.4	Spouštění programu (MS Windows) . . . . .	7
2.5	Spouštění programu (MS-DOS) . . . . .	7
<b>3</b>	<b>Ovládání</b>	<b>9</b>
3.1	Volby v menu . . . . .	9
3.1.1	Zobrazovat mezivýsledky . . . . .	9
3.1.2	Vstup ze souboru . . . . .	9
3.1.3	Výstup do souboru . . . . .	9
3.1.4	Export do T <sub>E</sub> Xu . . . . .	9
3.1.5	Počítat . . . . .	10
3.1.6	Konec . . . . .	10
3.2	Pravidla pro zadávání výrazů . . . . .	10
3.2.1	Operátory . . . . .	10
3.2.2	Zlomky . . . . .	10
3.2.3	Neurčité výrazy . . . . .	10
3.2.4	Proměnné . . . . .	10
3.2.5	Zaokrouhlování . . . . .	11
3.2.6	Funkce . . . . .	11
<b>4</b>	<b>Programování vlastních funkcí</b>	<b>12</b>
4.1	Příkazy . . . . .	12
4.1.1	<code>exit</code> . . . . .	12
4.1.2	<code>if</code> . . . . .	12
4.1.3	<code>goto</code> . . . . .	13
4.1.4	Práce s proměnnými . . . . .	13

<b>II</b>	<b>Programátorská dokumentace</b>	<b>14</b>
<b>1</b>	<b>Globální proměnné</b>	<b>15</b>
1.1	MAXCISLO . . . . .	15
1.2	nula, jedna, deset . . . . .	15
1.3	pozice . . . . .	15
1.4	rounded . . . . .	15
1.5	rozpoznavam . . . . .	16
1.6	ans . . . . .	16
<b>2</b>	<b>Postupy</b>	<b>17</b>
2.1	Ukládání zlomků . . . . .	17
2.2	Krácení . . . . .	17
2.3	Načítání . . . . .	18
2.3.1	Vstupy . . . . .	18
2.3.2	Výrazy . . . . .	18
2.3.3	Zlomky . . . . .	18
2.3.4	Čísla . . . . .	18
2.4	Ostatní . . . . .	18

Část I

**Uživatelský manuál**

# Kapitola 1

## Slovo úvodem

Program Fractions je určen pro počítání se zlomky. Umožňuje jednoduše zadávat matematické výrazy a počítat výsledky. Umožňuje zapsat několik výrazů do souboru a tento soubor poté nechat spočítat. Je podporován i výstup ve formátu vhodném pro následné zpracování sázecím programem L<sup>A</sup>T<sub>E</sub>X. Takto získáte přehledně zpracované výsledky.

Program Fractions je určen především pro nekomerční využití. Z toho plynou i nízké nároky na hardware a software. Naopak již zmiňovaný vstup i výstup do souboru umožňuje využití i při složitějších výpočtech.

Program je navíc vybaven velmi intuitivním a userfriendly ovládáním. Běží v DOSovém okně, což u mnohých zajisté vzbudí nostalgické vzpomínky. A ostatním připomene, jak bylo v dávných dobách ovládání jednoduché a přívětivé.

Doufám, že tento program splní Vaše očekávání a dopomůže Vám k lepším výsledkům

*Michal Hrušecký*  
autor programu

# Kapitola 2

## Instalace

Zkontrolujte prosím nejdříve, zda váš počítač splňuje minimální požadavky.

### 2.1 Požadavky

Zde je seznam požadavků. Uváděny jsou minimální požadavky i optimální konfigurace pro běh programu.

	minimální	optimální
Operační systém	MS Windows	MS-DOS
Další software	unzip	unzip, L <sup>A</sup> T <sub>E</sub> X
Volné místo na pevném disku	500Kbytů	1Mbyte

Tabulka 2.1: minimální požadavky

### 2.2 Instalace (MS Windows)

#### 2.2.1 Vyvolání příkazové řádky

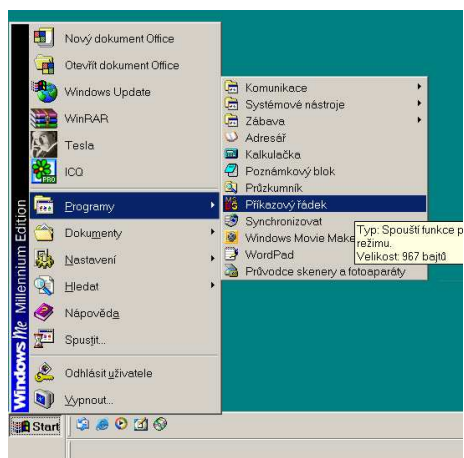
Vyvolejte nabídku **START**<sup>1</sup>. Dále vyvolejte podnabídku programy<sup>2</sup>. Pokud se v této nabídce nachází **Prompt MS-DOS** nebo **Příkazový řádek**, klikněte na něj. Pokud ne, vyvolejte podnabídku příslušenství. Pokud se nenachází ani tam, kontaktujte administrátora.

Po vyvolání příkazové řádky by se na obrazovce mělo objevit černé okno příkazového řádku **MS-DOS**. V tomto okně pokračujte podle popisu instalace pro uživatele systému **MS-DOS**.

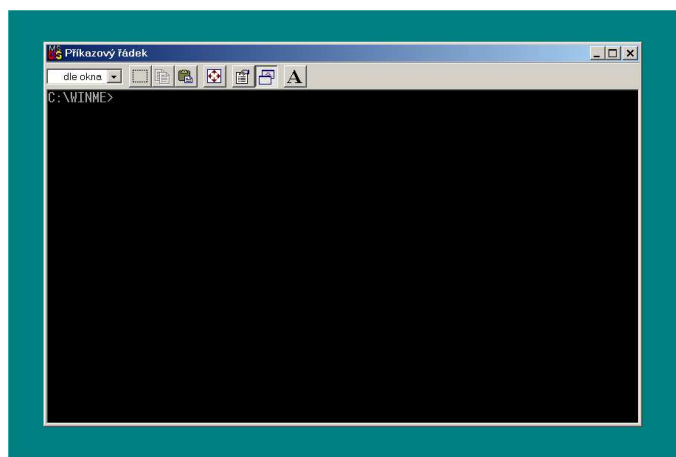
---

<sup>1</sup>Tento úkon se obvykle provádí stiskem levého tlačítka myši nad obrázkem v levém dolním rohu obrazovky nebo stiskem speciálního tlačítka na klávesnici

<sup>2</sup>najedete myší nad nápis Programy, který se objevil po vyvolání nabídky Start



Obrázek 2.1: nabídka START



Obrázek 2.2: příkazový řádek MS-DOS

## 2.3 Instalace (MS-DOS)

Vytvořte adresář, v němž chcete mít program nainstalovaný a rozbalte do něj obsah souboru frac.zip. Například, chci-li mít program nainstalovaný v adresáři C:\bin\frac a mám-li zip v adresáři D:\Software provedu to následujícím způsobem:

```
C:\> cd bin
C:\BIN> mkdir frac
C:\BIN> cd frac
C:\BIN\FRAC> copy d:\Software\frac.zip
          1 zkopírovaných souborů
C:\BIN\FRAC> unzip frac.zip
Archive: frac.zip
   inflating: frac.exe
   inflating: readme.pdf
C:\BIN\FRAC> del frac.zip
```

Obrázek 2.3: Postup instalace

Po úspěšném nainstalování by měl být program připraven k použití. Můžeme tedy přejít k vlastnímu spouštění.

## 2.4 Spouštění programu (MS Windows)

Vyvoláme příkazový řádek MS-DOS<sup>3</sup> a postupujeme stejně jako uživatelé operačního systému MS-DOS.

## 2.5 Spouštění programu (MS-DOS)

Program spustíme zadáním C:\>BIN\FRAC> frac.exe. Po spuštění se na obrazovce objeví úvodní menu.

---

<sup>3</sup>postup pro vyvolání příkazového řádku je popsán v části 2.2.1





Obrázek 2.4: Úvodní menu

## Kapitola 3

# Ovládání

### 3.1 Volby v menu

#### 3.1.1 Zobrazovat mezivýsledky

Je-li volba zapnutá [X], program bude při počítání zobrazovat mezivýsledek. To jest vzorec, ve kterém jsou již spočítány všechny násobení, dělení a kde je spočítán výsledek všech závorek.

#### 3.1.2 Vstup ze souboru

Je-li volba zapnutá [X], program se po zadání příkazu k počítání nebude dotazovat na výraz, který chcete počítat, ale načte si výraz ze souboru `input.txt`. Do tohoto souboru můžete zapsat výrazy, jež chcete spočítat. Jednotlivé výrazy musí být každý na samostatné řádce a musí splňovat podmínky pro zadávání výrazů(viz. 3.2)

#### 3.1.3 Výstup do souboru

Je-li volba zapnutá [X], program se bude po zadání příkazu k počítání zapisovat výsledky i do souboru `output.txt`.

#### 3.1.4 Export do $\text{\TeX}$

Obsah souboru `output.txt` převede na vstupní soubor pro sázecí software  $\text{\LaTeX}$ . Výsledek se uloží do souboru `Vysledky.tex` a obsah souboru `output.txt` se smaže. Vypíše se informační hláška:

```
Soubor byl zapsan jako Vysledky.tex
```

a vyčká se stisku klávesy. Po stisku libovolné klávesy je program opět připraven přijímat vaše příkazy.

### 3.1.5 Počítat

Umožní počítat. Není-li nastaven vstup ze souboru, program se zeptá na výraz, jenž chcete spočítat - Zadejte výraz. Zadejte tedy výraz<sup>1</sup> a potvrďte enterem. Po vypsání výsledku budete dotázáni na další zadání. Nechcete-li už dále pokračovat, zadejte #exit a potvrďte klávesou Enter. Program se vrátí do úvodní obrazovky.

### 3.1.6 Konec

Ukončí program.

## 3.2 Pravidla pro zadávání výrazů

### 3.2.1 Operátory

Ve výrazu se nesmí vyskytovat dva operátory hned po sobě. Chcete-li násobit či dělit záporným číslem, nebo chcete-li použít ve jmenovateli zlomku záporné číslo, musí být uzavřeno do závorek. Program podporuje tyto operandy:

+	sčítání
-	odčítání
*	násobení
/	dělení

Tabulka 3.1: podporované operátory

### 3.2.2 Zlomky

Zlomky se zadávají v tomto tvaru: čítelel\_jmenovatel. Čítelel i jmenovatel může být nahrazen uzavřeným výrazem. Např.  $((1+4)*4)/(5-4)$ .

### 3.2.3 Neurčité výrazy

Bude-li ve jmenovateli nula, nahradí se jedničkou a výsledek se označí jako zaokrouhlený (viz. 3.2.5)

### 3.2.4 Proměnné

Ve výrazech lze používat i proměnné. Po spuštění programu je k dispozici proměnná ans, do níž se ukládá hodnota posledního výrazu. Použití proměnné ve výrazu je možné zadáním znaku \$ a bezprostředně za ním musí

<sup>1</sup> přečtete si nejdříve pravidla pro zadávání výrazů v části 3.2

následovat název proměnné (např. `$ans+1`). Názvy proměnných mohou obsahovat jen písmena. Výchozí hodnota všech proměnných je nula. Zadáte-li do výrazu proměnnou, jejíž hodnota zatím nebyla definována, dosadí se místo ní nula. O práci s proměnnými se můžete více dozvědět v části 4.1.4.

### 3.2.5 Zaokrouhlování

Bude-li číselný výraz větší než 32000, bude výsledek (i mezi výsledkem) zaokrouhlen. Tuto skutečnost indikuje nápis `cca`, jenž se zobrazí před výsledkem. V takovémto případě výsledek není přesný a může se od skutečného výsledku značně lišit. Proto je lepší pokusit se přeformulovat zadaný výraz tak, aby se všechny mezivýsledky vešly do daných mezí.

### 3.2.6 Funkce

Do zadávaných výrazů lze používat také uživatelem definované funkce. Jejich použití je velmi snadné. Stačí obdobně jako při zadávání proměnných napsat znak `&` a bezprostředně za něj název funkce. Má-li funkce nějaký parametr zadá se do závorek, jež bezprostředně následují (např. `&mocni(2)`). Pokud žádný parametr funkce nemá, nebo jej nechcete zadávat, závorky nepište. Je důležité si uvědomit, že funkce mají přístup ke všem proměnným a mohou je měnit. Tak mohou poskytovat více výsledků, nebo požadovat více parametrů. Více se dozvíte v kapitole 4.

## Kapitola 4

# Programování vlastních funkcí

Do programu lze doprogramovat vlastní funkce a pak je dále používat. Funkce se ukládají do souboru, jehož jméno je zároveň i názvem funkce. Nová funkce se dá vložit tak, že pomocí textového editoru vytvoříme v pracovním adresáři programu soubor, jehož jméno je názvem funkce. Z toho plynou i jistá omezení a to ta, že v systému MS-DOS může být maximální délka názvu souboru pouhých 8 znaků. To je také maximální délka názvu funkce. Soubor se ukládá bez jakékoliv přípony.

### 4.1 Příkazy

Řádky obsahující příkazy začínají vždy znakem `#`. Některé z příkazů jako svůj parametr požadují nějakou hodnotu. Tou může být zlomek (případně obyčejné číslo), proměnná (např. `$ans`), nebo i celý výraz. Pokud to ale má být výraz, musí být z obou stran uzavřen do závorek.

#### 4.1.1 `exit`

Tento příkaz slouží pro ukončení vykonávání současné funkce (případně pro ukončení zadávání výrazů). Musí stát na samostatném řádku a má přednost před všemi ostatními.

#### 4.1.2 `if`

Tento příkaz slouží k větvení. Přesná syntaxe je:

*if hodnota operátor hodnota příkaz.*

Operátorem musí být jeden ze znaků `<`, `>`, `=`. Pokud nerovnost (případně rovnost) platí, provede se následující příkaz. To může být `goto`, `set` nebo

`get` se svými parametry (např. `# if $ans > 10 goto end`). Pokud bude příkaz zadán špatně, vyhodnotí se, jako kdyby nerovnost (rovnost neplatila).

### 4.1.3 `goto`

Tento příkaz umožní skok na návěští. Syntaxe je “`goto návěští`”, kde návěští je označená řádka programu. Návěští je jednoslovný název a zadává se takto: “`# navesti:`”, kde mezi názvem návěští a dvojtečkou nesmí být mezera.

### 4.1.4 Práce s proměnnými

Pomocí příkazů lze také nastavit nebo přecíst hodnotu proměnné. Hodnotu lze nastavit pomocí příkazu “`set jméno hodnota`” (např. `#set result $ans`). Tento příkaz lze použít i po příkazu `if`. Dále lze použít příkaz “`get název`” pro zjištění hodnoty proměnné (vypíše se na obrazovku). Pokud vaše funkce má podporovat vstupní parametr, předává se v proměnné `input`. Výsledek práce funkce by se měl zapsat do proměnné `result`. Tato proměnná se po skončení běhu funkce použije jako výsledek.