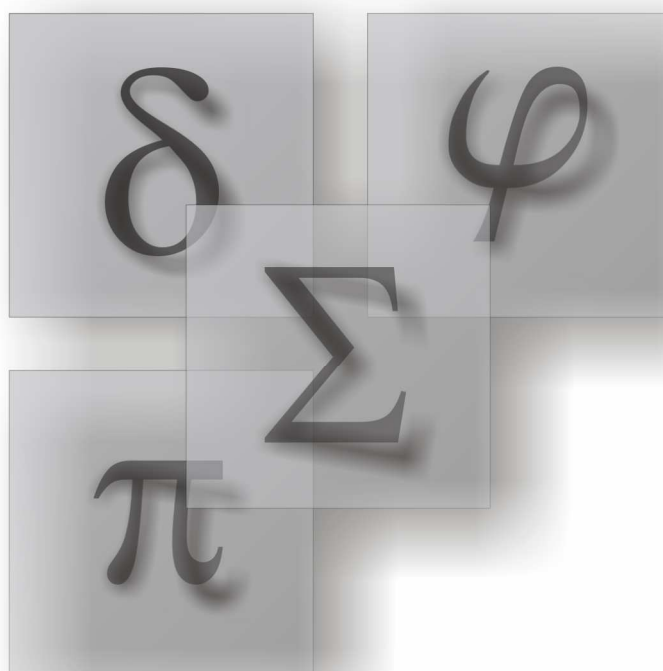


**SOČ 2005 – 2006**  
Středoškolská odborná činnost  
Matematika a matematická informatika  
Softwarové zpracování úloh matematiky a matematické informatiky



## **Smart Counter 2**

**System počítačové algebry**

**Štěpán Kozák**

3. ročník – septima A  
Gymnázium Jeseník,  
Komenského 281.

## Obsah

1. Anotace	2
2. Cíle projektu	3
3. Popis programu	4 – 18
3.1. Hardwarové a softwarové požadavky	4
3.2. Uživatelské rozhraní	4 – 5
3.2.1. Tipy při práci s programem	5
3.3. Možnosti programu	6 – 18
3.3.1. Seznam operátorů a funkcí	6
3.3.2. Symbolické operace s výrazy	7 – 12
3.3.2.1. Simplifikace (zjednodušování) výrazů	7
3.3.2.2. Rozšiřování výrazů	8
3.3.2.3. Substituce výrazů	8 – 9
3.3.2.4. Parciální derivace libovolného stupně	9
3.3.2.5. Řešení vybraných typů rovnic	10
3.3.2.6. Taylorův rozvoj	11
3.3.2.7. Řešení soustav rovnic	11 – 12
3.3.2.8. Dělení mnohočlenů	12
3.3.3. Numerické operace s výrazy	13 – 14
3.3.3.1. Vyhodnocování výrazu	13
3.3.3.2. Numerické řešení rovnic	13 – 14
3.3.3.3. Numerická integrace	14
3.3.3.4. Výpočet nej. spol. dělitele a nej. spol. násobku	15
3.3.3.5. Rozklad na prvočísla	15
3.3.4. Grafické funkce	16 – 17
3.3.4.1. Vykreslení 2D grafu funkce o jedné neznámé	16 – 17
3.3.5. Analytická geometrie	18 – 21
3.3.6. Ostatní funkce	22
3.3.6.1. Převody mezi číselnými soustavami	22
4. Použité postupy a algoritmy	23 – 26
5. Závěr	27
Seznam literatury	28
Seznam příloh	
1. CD s programem	

## 1. Anotace

Smart Counter 2 je komplexní matematický balík (CAS – Computer Algebra System), který je určen především jako pomocník při řešení časově náročných matematických problémů na úrovni středoškolské matematiky. Uplatnění najde nejen u studentů gymnázií a technicky zaměřených škol, kterým pomůže při studiu, ale také u učitelů matematiky, kteří v něm najdou užitečný nástroj pro přípravu do hodin. Program se skládá ze tří částí.

První a zároveň stěžejní částí je rozhraní pro symbolické a numerické operace a výpočty s libovolnými algebraickými výrazy (např. simplifikace výrazů, rozšiřování výrazů, parciální derivace libovolného stupně, řešení různých druhů rovnic a další).

Další, neméně významnou součástí programu, je jeho grafické jádro, díky němuž lze vykreslit graf jakékoli explicitně zadané funkce ve tvaru  $y = f(x)$ . Při vykreslování je použito vyhlazování křivek. Export grafu je možný do moderního vektorového formátu SVG.

Třetí částí je editor matematických výrazů, v němž lze velice snadno a rychle vytvořit jakýkoli matematický zápis. Díky integrované podpoře nejnovějších standardů pro zápis matematických výrazů dokáže editor svůj výstup generovat buďto jako obrázek nebo jako zdrojový kód MathML.

Výhodou oproti ostatním podobně orientovaným aplikacím je především komplexnost programu, přičemž však byl kladen důraz na zachování jednoduchosti a intuitivnosti ovládání. Výhodou je také to, že veškeré matematické zápisy jsou zobrazeny graficky nikoli pouze textově.

## 2. Cíle projektu

Program Smart Counter 2 byl vytvořen k usnadnění studia matematiky na středních školách. Při použití tohoto nástroje se totiž student nemusí zabývat zdlouhavými algebraickými úpravami a numerickými výpočty, ale může se plně soustředit na podstatu věci – tj. postup řešení úlohy, pochopení logických souvislostí, apod. Student se též vyhne zbytečným numerickým chybám, jejichž hledání a následné odstranění by bylo další ztrátou času.

Cílem tedy bylo vytvořit komplexní matematický nástroj dostupný pro všechny (jak cenou tak i přívětivým uživatelským prostředím), který je schopný nejen řešit různé úlohy středoškolské matematiky, ale také pomoci při zápisu úlohy (zabudované vykreslování matematických výrazů).

### 3. Popis programu

#### 3.1. Hardwarové a softwarové požadavky

##### Hardware

CPU: 300 Mhz (800 Mhz doporučeno)  
RAM: 64 MB RAM (256 MB doporučeno)

##### Software

OS: Windows 95 a novější nebo Linux  
Java Virtual Machine 5.0  
Prohlížeč podporující MathML a SVG (např. Mozilla Firefox) \*

\* není nutné pro běh programu

Program byl úspěšně testován na těchto konfiguracích:

Procesor	Operační paměť	Operační systém
AMD K6-II 500 Mhz	256	Windows XP SP1
AMD 64 3000+ (1800 Mhz)	512	Windows XP SP2
Pentium M 1.86 Ghz	512	Windows XP SP2
AMD Duron 700 Mhz	128	Windows 98

#### 3.2. Uživatelské rozhraní

Smart Counter 2

Editor výrazů

$$\frac{x^2 - y^2}{x + y} - \frac{1}{2} \cdot x \cdot y$$

Syntaxe: OK      Kopírovat do schránky:    Obrázek    MathML

`(x^2-y^2)/(x+y)-1/2*x*y`

Aktualizovat automaticky      Vymazat    Zobrazit

Vyhodnocení výrazu    NSD / NSN    Prvočíselný rozklad    Převody    Analytická geometrie  
Taylorův rozvoj    Numerická integrace    Substituce    Dělení mnohočlenů  
Řešení rovnic    Soustavy rovnic    Graf funkce    Úprava výrazu    Derivace

Funkce:

xmin:     xmax:     ymin:     ymax:     Šířka:    

Historie

```
>> taylor[tan(x), x, 9, 0]
[ x + x^3/3 + 2*x^5/15 + 17*x^7/315 + 62*x^9/2835 ]

>> graph[-x^3+2*x+4]
```

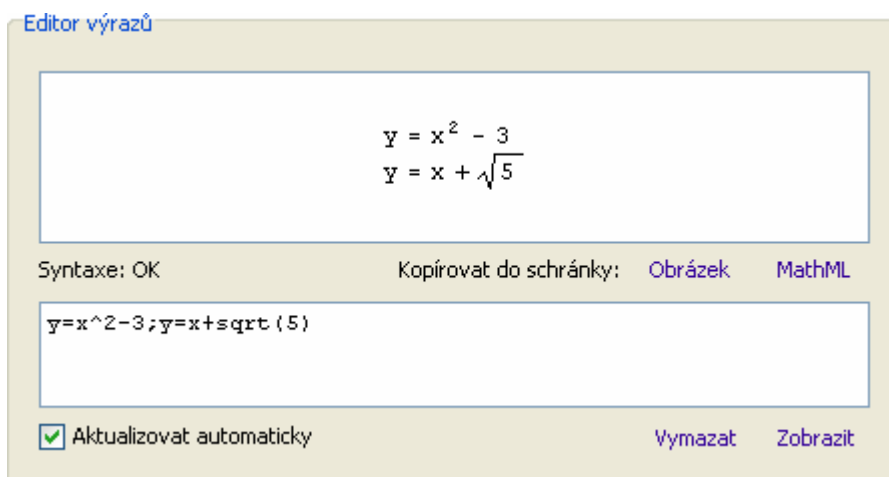
Smart Counter verze 2.0.0 (c) 2005 Štěpán Kozák

### 3.2.1. Tipy při práci s programem

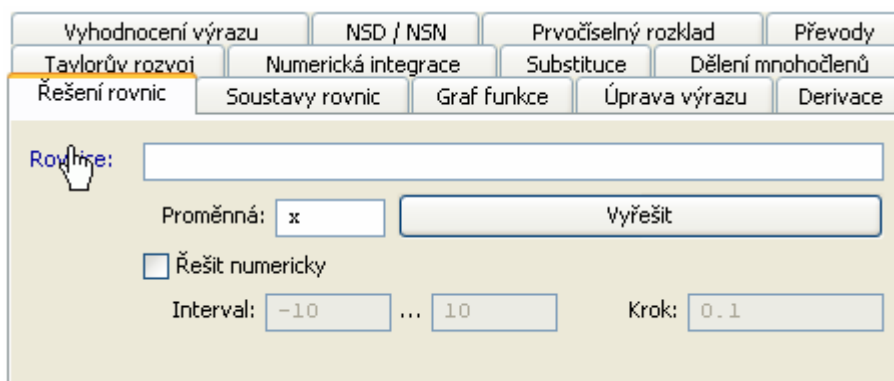
- Pokud nemáte dostatečně rychlý počítač, můžete práci s programem urychlit tím, že volbu „Aktualizovat automaticky“ necháte nezaškrtnutou. Poté se bude obrázek generovat pouze po stisku tlačítka „Zobrazit“.

Aktualizovat automaticky

- Pro zobrazení více výrazů v jednom obrázku (například při zobrazení soustavy rovnic) použijte jako oddělovač těchto výrazů středník.



- Pro rychlé kopírování zadání příkladu z okna Editoru výrazů do jednotlivých textových polí v záložkách pro výpočty klepněte na název textového pole, kam chcete text vložit.



- Pokud chcete získaný výsledek dále použít, nemusíte jej ručně přepisovat, stačí výsledek označit a zkopírovat do schránky, nezkopíruje se daný obrázek, ale jeho syntaktický zápis. V případě grafu se do schránky zkopíruje zdrojový text SVG.
- Pokud nechcete čekat na dokončení dlouhého výpočtu, můžete jej zrušit najetím na dolní lištu na nápis „Probíhá výpočet ...“ a kliknutím na „Zrušit výpočet“.



### 3.3. Možnosti programu

#### 3.3.1. Seznam operátorů a funkcí

Zápis jakéhokoli výrazu se provádí pomocí standardních operátorů a funkcí známých z podobně zaměřených programů.

Operace	Operátor	Příklad použití
Sčítání	+	např. $a + b$
Odčítání	-	např. $a - b$
Násobení	*	např. $a * b$
Dělení (zlomek)	/	např. $a / b$
Umocnění	^	např. $a ^ b$
Unární minus	-	např. $-a$
Plusminus	+-	např. $a +- b$

#### Význam funkce

Sinus
Kosinus
Tangens
Arcussinus
Arcuskosinus
Arcustangens
Hyperbolický sinus
Hyperbolický kosinus
Hyperbolický tangens
Exponenciální fce o základu e ( $e^x$ )
Přirozený logaritmus
Logaritmus
Absolutní hodnota
Druhá odmocnina
Signum

#### Použití

$\sin(\text{výraz})$
$\cos(\text{výraz})$
$\tan(\text{výraz})$
$\text{asin}(\text{výraz})$
$\text{acos}(\text{výraz})$
$\text{atan}(\text{výraz})$
$\sinh(\text{výraz})$
$\cosh(\text{výraz})$
$\tanh(\text{výraz})$
$\exp(\text{výraz})$
$\ln(\text{výraz})$
$\log(\text{výraz}, [\text{základ logaritmu} - \text{defaultně } 10])$
$\text{abs}(\text{výraz})$
$\text{sqrt}(\text{výraz})$
$\text{sign}(\text{výraz})$

#### Předdefinované konstanty

Ludolfovo číslo ( $\pi$ )

#### Název      Hodnota

pi            3.141592653589793

#### Pravidla při vytváření výrazu

- 1) Je možno používat pouze kulaté závorky tj. „(“ a „)“.
- 2) Desetinným oddělovačem je tečka nikoli čárka.
- 3) Pro oddělení parametrů funkcí se používá čárka.
- 4) Pro vytvoření dolního indexu u názvu proměnné se používá podtržítka, např. pro vytvoření  $x_1$  použijete zápis  $x\_1$ .
- 5) Názvy proměnných mohou být libovolně dlouhé, ale mohou obsahovat pouze malá nebo velká písmena bez háčků a čárek, číslice (nesmí však začínat číslicí) a podtržítka (viz bod 4). Ostatní znaky by mohly způsobit chybné chování programu.
- 6) Nikdy se nesmí vyskytovat dva operátory vedle sebe, proto je výraz s unárním minus vždy v závorce tj. např.  $a+(-b)$ .
- 7) Mocnina ve složitějších výrazech musí být vždy v závorce.
- 8) Čitatel i jmenovatel zlomku musí být ve složitějších výrazech v závorce.
- 9) Není možné psát zkrácené zápisy typu  $2x$  nebo  $(a+b)(a-b)$ , vždy je nutné explicitně vyjádřit násobení operátorem „\*“. Správně tedy je  $2*x$  resp.  $(a+b)*(a-b)$ .
- 10) Mocniny vyššího stupně než dva se dělají umocněním s racionálním exponentem tj. např. třetí odmocnina z  $x$  se zapíše jako  $(x)^(1/3)$ .

### 3.3.2. Symbolické operace s výrazy

#### 3.3.2.1. Simplifikace (zjednodušování) výrazů

##### Obecné informace:

Program upraví jakýkoli algebraický výraz do co nejjednodušší podoby.

Při zjednodušování je používáno těchto postupů:

- Rozklad v součin
  - Částečné odmocnění
  - Vytýkání
  - Rozklad pomocí vzorců
    - $a^2 - b^2 = (a + b)(a - b)$
    - $a^3 + b^3 = (a + b)(a^2 - ab + b^2)$
    - $a^3 - b^3 = (a - b)(a^2 + ab + b^2)$
- Krácení zlomků
- Převod na společný jmenovatel
- Převod složeného zlomku na jednoduchý
- Usměrnění zlomku

##### Postup:

Do textového pole „Výraz“ zadejte výraz, který chcete zjednodušit a klepněte na tlačítko „Zjednodušit“.

##### Příklady:

Matematický zápis	Zápis v programu	Výsledek
$\frac{\frac{a^2 + b^2}{a} + b}{\left(\frac{1}{a^2} + \frac{1}{b^2}\right) \cdot \frac{a^3 - b^3}{a^2 + b^2}}$	$((a^2+b^2)/a+b)/((1/(a^2)+1/(b^2))*((a^3-b^3)/(a^2+b^2)))$	$\frac{a \cdot b^2}{a - b}$
$\frac{\frac{x}{x-1} - \frac{x+1}{x}}{\frac{x}{x+1} - \frac{x-1}{x}}$	$\frac{(x/(x-1) - (x+1)/x)}{(x/(x+1) - (x-1)/x)}$	$\frac{x+1}{x-1}$



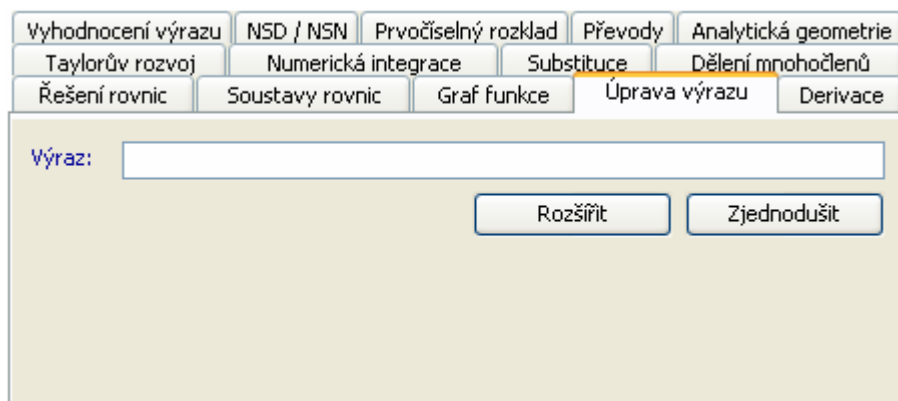
### 3.3.2.2. Rozšiřování výrazů

#### Obecné informace:

Rozšiřování výrazů spočívá v rozvinutí algebraických výrazů do řady součtu jeho podvýrazu (rozvinutí výrazu do podoby běžného polynomu). Po roznásobení a umocnění všech výrazů dojde ke sloučení (sečtení, resp. odečtení) výrazů, u kterých je to možné.

#### Postup:

Do textového pole „Výraz“ zadejte výraz, který chcete rozvinout a klepněte na tlačítko „Rozšířit“.



#### Příklady:

Matematický zápis	Zápis v programu	Výsledek
$(x - y)^2$	$(x-y)^2$	$x^2 - 2 \cdot x \cdot y + y^2$
$(x - y) \cdot (x + y)$	$(x-y) \cdot (x+y)$	$x^2 - y^2$

### 3.3.2.3. Substituce výrazů

#### Obecné informace:

Program nahradí veškerý výskyt dané proměnné jakýmkoli jiným zadaným výrazem nebo číselnou hodnotou.

#### Postup:

Do textového pole „Výraz“ zadejte výraz, na který chcete aplikovat substituci. Do pole proměnná zadejte jméno proměnné, kterou chcete nahradit. Zástupný výraz vepište do pole „Výrazem“. Pro potvrzení operace klepněte na tlačítko „Nahradit“.

Řešení rovnic	Soustavy rovnic	Graf funkce	Úprava výrazu	Derivace
Vyhodnocení výrazu	NSD / NSN	Prvočíselný rozklad	Převody	Analytická geometrie
Taylorův rozvoj	Numerická integrace	Substituce	Dělení mnohočlenů	

Výraz:

Proměnná:  Výrazem:

### Příklady:

Matematický zápis	Zápis v programu	Výsledek
$a + b^2 \cdot (a^3 - 2)$	Výraz: a+b^2*(a^3-2) Proměnná: a Výrazem: y-1	$y - 1 + b^2 \cdot ((y - 1)^3 - 2)$

### 3.3.2.4. Parciální derivace libovolného stupně

#### Obecné informace:

Program provede výpočet derivace dané funkce s ohledem na zadanou proměnnou.

#### Postup:

Do textového pole „Funkce“ zadejte funkci, kterou chcete derivovat. Do pole „Derivovat podle“ zadejte název proměnné, podle které se bude derivovat (všechny ostatní proměnné ve funkci budou brány jako konstanty). Zadejte stupeň derivace a klepněte na tlačítko „Derivovat“.

Vyhodnocení výrazu	NSD / NSN	Prvočíselný rozklad	Převody	Analytická geometrie
Taylorův rozvoj	Numerická integrace	Substituce	Dělení mnohočlenů	
Řešení rovnic	Soustavy rovnic	Graf funkce	Úprava výrazu	Derivace

Funkce:

Derivovat podle:  Stupeň:

### Příklady:

Matematický zápis	Zápis v programu	Výsledek
$\tan x$	Výraz: tan(x) Podle: x Stupeň: 3	$\frac{2 \cdot (\cos^2 x + 3 \cdot \sin^2 x)}{\cos^4 x}$

### 3.3.2.5. Řešení vybraných typů rovnic

#### Obecné informace:

Program dokáže symbolicky (řešením je symbolický výraz, nikoli pouze číslo) vyřešit tyto typy rovnic (v oboru reálných čísel):

- Lineární rovnice
- Kvadratické rovnice
- Kubické rovnice
- Binomické rovnice (algebraické řešení)
- Trinomické rovnice
- Iracionální rovnice (rovnice s neznámou pod odmocninou)
- Rovnice vyšších stupňů v součinném nebo podílovém tvaru, kde v součinu resp. podílu jsou některé z výše jmenovaných typů rovnic

#### Postup:

Do textového pole „Rovnice“ zadejte rovnici. Do pole „Proměnná“ zadejte název proměnné, kterou chcete vyjádřit a klepněte na tlačítko „Vyřešit“.

#### Příklady:

Matematický zápis	Zápis v programu	Výsledek
$4 \cdot \sqrt{8-x} - \sqrt{6 \cdot x + 150} = 0$	$4 * (\text{sqrt}(8-x)) - \text{sqrt}(6 * x + 150) = 0$	$x = -1$
$\sqrt{2 \cdot x - a} = 2$	$\text{sqrt}(2 * x - a) = 2$	$x = \frac{a + 4}{2}$

#### Tip:

Řešení rovnic lze použít např. ve fyzice při vyjadřování veličin ze vzorců.

Matematický zápis	Zápis v programu	Výsledek
$\frac{1}{R} = \frac{1}{R_1} + \frac{1}{R_2}$	$1/R = 1/R_1 + 1/R_2$	$R = \frac{R_1 \cdot R_2}{R_1 + R_2}$

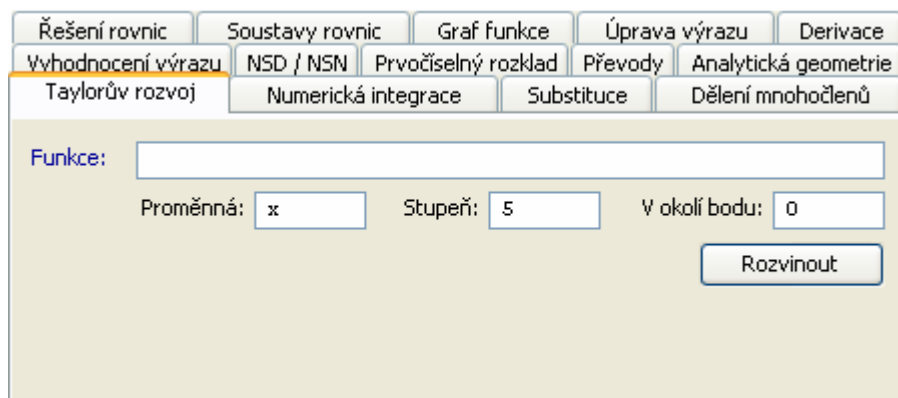
### 3.3.2.6. Taylorův rozvoj

#### Obecné informace:

Program vyjádří Taylorův rozvoj dané funkce do určeného stupně.

#### Postup:

Do textového pole „Funkce“ zadejte výraz, který chcete rozvinout. Do pole „Proměnná“ zadejte název proměnné, která je ve výrazu použita. Zadejte stupeň rozvinutí a okolí bodu a klepněte na tlačítko „Rozvinout“.



#### Příklady:

Matematický zápis	Zápis v programu	Výsledek
$\tan x$	Funkce: $\tan(x)$ Proměnná: $x$ Stupeň: 5 Bod: 0	$x + \frac{x^3}{3} + \frac{2 \cdot x^5}{15}$

### 3.3.2.7. Řešení soustav rovnic

#### Obecné informace:

Program vyřeší soustavu zadaných rovnic pomocí dosazovací metody.

#### Postup:

Do textového pole „Rovnice“ zadejte rovnice oddělené středníkem, které tvoří danou soustavu. Do pole „Proměnné“ napište názvy proměnných oddělené středníkem, pro které má být soustava vyřešena.

Vyhodnocení výrazu	NSD / NSN	Prvočíselný rozklad	Převody	Analytická geometrie
Taylorův rozvoj	Numerická integrace	Substituce	Dělení mnohočlenů	
Řešení rovnic	Soustavy rovnic	Graf funkce	Úprava výrazu	Derivace

Rovnice:

Proměnné:

**Příklady:**

Matematický zápis	Zápis v programu	Výsledek
$2 \cdot x - y = 4$ $3 \cdot x - 4 \cdot y = 1$	Rovnice: $2 \cdot x - y = 4; 3 \cdot x - 4 \cdot y = 1$ Proměnné: $x, y$	$x = 3$ $y = 2$

**3.3.2.8. Dělení mnohočlenů**

**Obecné informace:**

Program vydělí zadaný mnohočlen (o jedné proměnné) jiným mnohočlenem.

**Postup:**

Do textového pole „Dělenec“ zadejte polynom, který chcete vydělit. Do pole „Dělitel“ zadejte polynom, kterým chcete dělit. Do pole „Proměnná“ zadejte název proměnné použité ve výrazech.

Řešení rovnic	Soustavy rovnic	Graf funkce	Úprava výrazu	Derivace
Vyhodnocení výrazu	NSD / NSN	Prvočíselný rozklad	Převody	Analytická geometrie
Taylorův rozvoj	Numerická integrace	Substituce	Dělení mnohočlenů	

Dělenec:

Proměnná:  Dělitel:

**Příklady:**

Matematický zápis	Zápis v programu	Výsledek
$\frac{x^5 - 1}{x - 1}$	Dělenec: $x^5 - 1$ Dělitel: $x - 1$ Proměnná: $x$	$x^4 + x^3 + x^2 + x + 1$

### 3.3.3. Numerické operace s výrazy

#### 3.3.3.1. Vyhodnocování výrazu

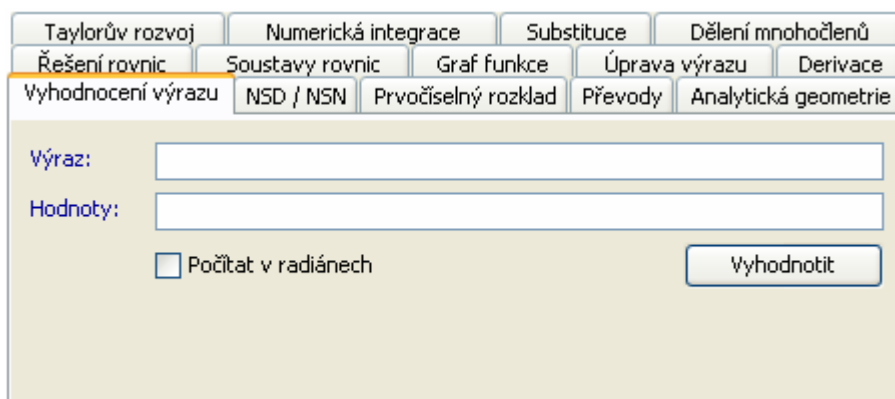
##### Obecné informace:

Program dokáže vyhodnotit jakýkoli výraz obsahující libovolný počet proměnných, tj. zjistí jeho numerickou hodnotu.

##### Postup:

Do textového pole „Výraz“ zadejte výraz, který chcete vyhodnotit. Do pole „Hodnoty“ zadejte názvy a číselné hodnoty proměnných ve tvaru „proměnná=hodnota“ oddělené středníkem. Pokud chcete počítat s úhlovou jednotkou radiány, zaškrtněte pole „Počítat v radiánech“ (V opačném případě bude úhlovou jednotkou stupeň). Pro vyhodnocení klepněte na tlačítko „Vyhodnotit“.

Pokud se ve výrazu objeví proměnná, jejíž hodnota nebyla definována v poli „Hodnoty“, dojde k zjednodušení výrazu – tedy nedostáváme numerickou hodnotu, ale opět výraz.



##### Příklady:

Matematický zápis	Zápis v programu	Výsledek
$\frac{\sqrt{5} + \sqrt{3} \cdot x}{2}$	Výraz: (sqrt(5)+sqrt(3)*x)/2 Hodnoty: x=4	4.582135603887
$\frac{1}{\sqrt[3]{3} + 2}$	Výraz: 1/(3^(1/3)+2) Hodnoty:	0.290507698403

#### 3.3.3.2. Numerické řešení rovnic

##### Obecné informace:

Program dokáže numericky (výsledkem je desetinné číslo s deseti platnými číslicemi) vyřešit jakoukoli rovnici o jedné neznámé v zadaném intervalu pomocí Newtonovy metody tečen.

## Postup:

Do textového pole „Rovnice“ zadejte rovnici. Do pole „Proměnná“ zadejte název proměnné. Zadejte interval, ve kterém chcete zjistit kořeny. Do pole krok zadejte velikost „podintervalu“. V každém podintervalu bude nalezen nejvýše jeden kořen, takže pokud má rovnice více kořenů, jejichž hodnota se liší o méně než je hodnota v tomto poli, bude nalezen jen jeden z těchto kořenů. Čím je tedy hodnota kroku menší, tím je větší pravděpodobnost, že budou nalezeny všechny kořeny dané rovnice (také se tím ale prodlouží doba potřebná pro výpočet).

Navigation: Vyhodnocení výrazu, NSD / NSN, Prvočíselný rozklad, Převody, Analytická geometrie  
Taylorův rozvoj, Numerická integrace, Substituce, Dělení mnohočlenů  
Řešení rovnic, Soustavy rovnic, Graf funkce, Úprava výrazu, Derivace

Rovnice:

Proměnná:

Řešit numericky

Interval:  ...  Krok:

## Příklady:

Matematický zápis	Zápis v programu	Výsledek
$6 \cdot x^5 - 3 \cdot x + 5$	Rovnice: $6 \cdot x^5 - 3 \cdot x + 5$ Proměnná: x Interval: -10 ... 10 Krok: 0.1	$x = -1.0642819765$

### 3.3.3.3. Numerická integrace

#### Obecné informace:

Program vypočte určitý integrál dané funkce (obsah plochy pod křivkou funkce).

#### Postup:

Zadejte funkci a proměnnou, rozsah intervalu a klepněte na tlačítko „Integrovat“.

Navigation: Řešení rovnic, Soustavy rovnic, Graf funkce, Úprava výrazu, Derivace  
Vyhodnocení výrazu, NSD / NSN, Prvočíselný rozklad, Převody, Analytická geometrie  
Taylorův rozvoj, Numerická integrace, Substituce, Dělení mnohočlenů

Funkce:

Proměnná:  Interval:  ...

### 3.3.3.4. Výpočet největšího spol. dělitele a nejmenšího spol. násobku

#### Obecné informace:

Program vypočítá největší společný dělitel, resp. nejmenší společný násobek libovolného množství čísel.

#### Postup:

Zadejte čísla oddělená středníkem, pro která chcete spočítat NSD resp. NSN.

Taylorův rozvoj	Numerická integrace	Substituce	Dělení mnohočlenů	
Řešení rovnic	Soustavy rovnic	Graf funkce	Úprava výrazu	Derivace
Vyhodnocení výrazu	NSD / NSN	Prvočíselný rozklad	Převody	Analytická geometrie

Čísla:

### 3.3.3.5. Rozklad na prvočísla

#### Obecné informace:

Program rozloží číslo na součin prvočísel.

#### Postup:

Zadejte číslo, které chcete rozložit.

Taylorův rozvoj	Numerická integrace	Substituce	Dělení mnohočlenů	
Řešení rovnic	Soustavy rovnic	Graf funkce	Úprava výrazu	Derivace
Vyhodnocení výrazu	NSD / NSN	Prvočíselný rozklad	Převody	Analytická geometrie

Číslo:



### 3.3.4. Grafické funkce

#### 3.3.4.1. Vykreslení 2D grafu funkce o jedné neznámé

##### Obecné informace:

Program vykreslí graf jakékoli zadané funkce v zadaném rozmezí hodnot na osách.

##### Postup:

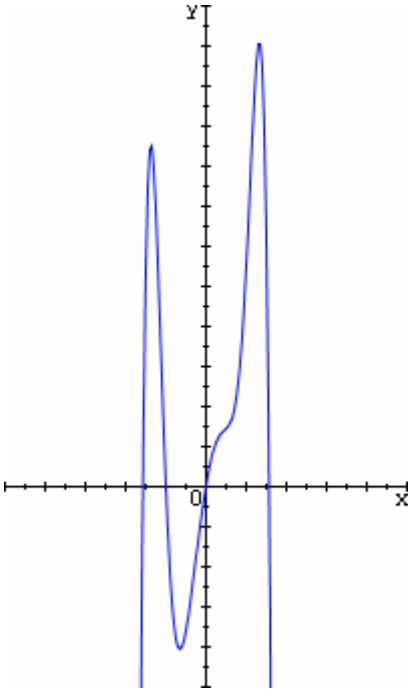
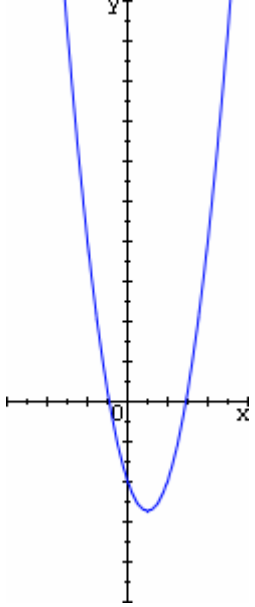
Do textového pole „Funkce“ zadejte funkci  $f(x)$ , kterou chcete vykreslit (proměnná je vždy  $x$ ). Dále zadejte rozmezí na ose  $x$  ( $x_{\min}$  –  $x_{\max}$ ) a na ose  $y$  ( $y_{\min}$  –  $y_{\max}$ ) a klepněte na tlačítko „Vykreslit“.

Vyhodnocení výrazu	NSD / NSN	Prvočíselný rozklad	Převody	Analytická geometrie
Taylorův rozvoj	Numerická integrace	Substituce	Dělení mnohočlenů	
Řešení rovnic	Soustavy rovnic	Graf funkce	Úprava výrazu	Derivace

Funkce:

xmin:  xmax:  ymin:  ymax:  Šířka:

##### Příklady:

Vykreslete graf funkce: $(6 \cdot x^5 - 3 \cdot x + 3) \cdot \sin(2 \cdot x)$	Řešte graficky rovnici: $3 \cdot x^2 - 3 \cdot x - 2 = 0$
	

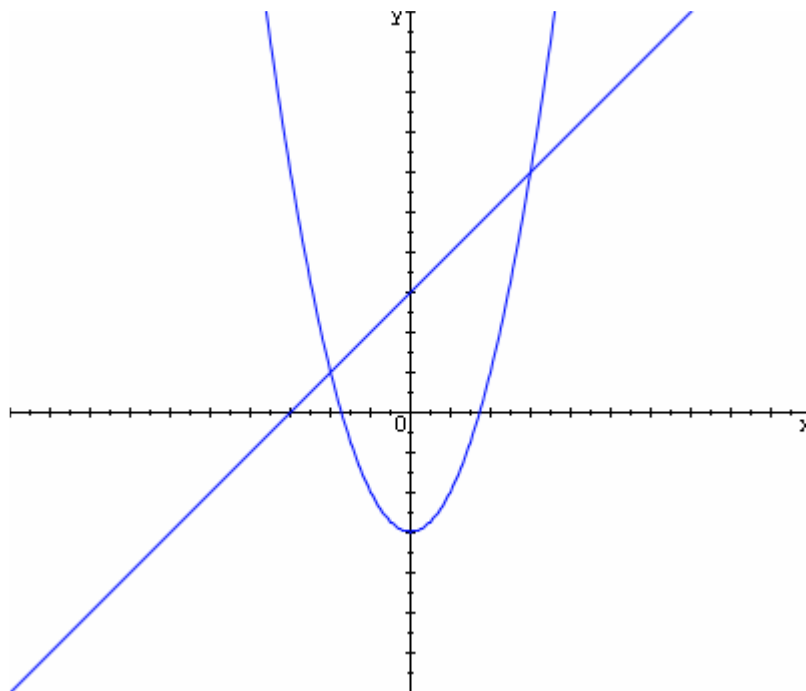
**Tip:**

Vykreslení grafu se dá také použít pro grafické řešení soustav rovnic o dvou neznámých. Pro vykreslení více grafů do jednoho obrázku zadejte funkce oddělené středníkem (maximální počet funkcí v jednom obrázku je 5).

Např. Řešte graficky soustavu rovnic:

$$y = x^2 - 3$$

$$y = x + 3$$



Řešení: { [-2, 1] [3, 6] }

**Export do SVG**

Pro získání zdrojového kódu SVG grafu stačí zkopírovat obrázek grafu z Historie do schránky. Do schránky se vloží zdrojový text.

### 3.3.6. Analytická geometrie

Pro výpočty z oboru analytické geometrie je v programu implementován primitivní skriptovací jazyk. Prozatím jsou v něm zahrnuty body, vektory a lineární útvary (přímka, rovina) v prostoru i rovině a polohové i metrické úlohy s těmito útvary.

#### Pravidla:

- Každý řádek znamená další příkaz
- Vytváření objektů může být libovolně vnořené
- Oddělovačem parametrů je středník
- Jazyk je case-insensitive; Vector3D je totéž co vector3d
- Existují pouze dva typy příkazů:
  - Uložení objektu / hodnoty do proměnné
  - Vypsání objektu / hodnoty do Historie
- Každá proměnná musí mít jedinečný název
- Název proměnné nesmí být rezervovaným (klíčovým) slovem, tj. {"print", "line2d", "line3d", "plane3d", "point2d", "point3d", "vector2d", "vector3d", "position", "deviation", "distance", "size", "scalarproduct", "vectorproduct"};
- Komentáře se dělají pomocí dvojitého lomítka //

#### Uložení objektu do proměnné:

**Syntaxe:** jméno\_proměnné=typ\_objektu(parametry)

**Příklady:** u=vector3d("1;2;3")  
p=line2d("2\*x+y-3";"universal")

#### Seznam objektů a jejich konstruktorů:

Objekt	Popis	Konstruktory	Vysvětlivky
<b>Point2D</b>	Bod v rovině	point2d ("x; y")	x, y ... souřadnice bodu
<b>Point3D</b>	Bod v prostoru	point3d ("x; y; z")	x, y, z ... souřadnice bodu
<b>Vector2D</b>	Vektor v rovině	vector2d ("x; y") vector2d (A; B)	x, y ... souřadnice vektoru A, B ... dva různé body (Point2D)
<b>Vector3D</b>	Vektor v prostoru	vector3d ("x; y; z") vector3d (A; B)	x, y, z ... souřadnice vektoru A, B ... dva různé body (Point3D)
<b>Line2D</b>	Přímka v rovině	line2d (A; B) line2d (A; u) line2d("obecné/směrnice/parametrické vyjádření"; "typ_vyjádření"; "parametr")	A, B ... dva různé body (Point2D) u ... směrový vektor přímky (Vector2D)  typ_vyjádření ... nabývá hodnot „universal“ (obecná rovnice), „parametric“ (parametrické rovnice) nebo „directional“ (směrnice rovnice)  parametr ... název parametru (bere se v potaz jen pokud je zadán typ rovnice „parametric“, zapsán musí být vždy)

Objekt	Popis	Konstruktory	Vysvětlivky
<b>Line3D</b>	Přímka v prostoru	<pre>line2d (A; B) line2d (A; u) line3d ("parametrické vyjádření"; "parametr")</pre>	<p><i>A, B ... dva různé body (Point3D)</i>  <i>u ... směrový vektor přímky (Vector3D)</i></p> <p><i>parametr ... název parametru</i></p>
<b>Plane3D</b>	Rovina v prostoru	<pre>plane3d (A; B; C) plane3d (u; v; A) plane3d (n; A) plane3d (p; q) plane3d ("parametrické vyjádření"; "parametr") plane3d ("obecné vyjádření")</pre>	<p><i>A, B, C ... tři různé body (Point3D)</i></p> <p><i>u, v ... směrové vektory roviny</i>  <i>n ... normálový vektor roviny</i></p> <p><i>p, q ... dvě různoběžné nebo rovnoběžné různé přímky určující rovinu (Line3D)</i></p> <p><i>parametr ... název parametru</i></p>
<b>Position</b>	Vzájemná poloha dvou objektů	<pre>position (line2d; line2d) position (line3d; line3d) position (line3d; plane3d) position (plane3d; plane3d)</pre>	<i>Vrátí vzájemnou polohu daných dvou objektů, pokud se objekty prolínají, vrátí i jejich průsečík, resp. průsečnici.</i>
<b>Deviation</b>	Vzájemná odchylka dvou objektů	<pre>deviation (line2d; line2d) deviation (line3d; line3d) deviation (vector2d; vector2d) deviation (vector3d; vector3d) deviation (line3d; plane3d) deviation (plane3d; plane3d)</pre>	<i>Vrátí odchylku dvou objektů.</i>
<b>Distance</b>	Vzájemná vzdálenost dvou objektů	<pre>distance (point2d; line2d) distance (point3d; line3d) distance (point3d; plane3d)</pre>	<i>Vrátí vzdálenost bodu od objektu.</i>
<b>Size</b>	Velikost objektu	<pre>size (vector2d) size (vector3d)</pre>	<i>Vrátí velikost vektoru.</i>
<b>ScalarProduct</b>	Skalární součin dvou vektorů	<pre>scalarproduct (vector2d, vector2d) scalarproduct (vector3d, vector2d)</pre>	<i>Vrátí skalární součin dvou vektorů.</i>
<b>VectorProduct</b>	Vektorový součin dvou vektorů	<pre>vectorproduct (vector3d; vector3d)</pre>	<i>Vrátí vektorový součin dvou vektorů.</i>
<b>NormalLine</b>	Kolmá přímka	<pre>normalline (point2d; line2d) normalline (point3d; line3d) normalline (point3d; plane3d)</pre>	<i>Vrátí kolmou přímku vedenou bodem k danému objektu.</i>

Objekt	Popis	Konstruktory	Vysvětlivky
<b>ParallelLine</b>	Rovnoběžná přímka	parallelline (point2d; line2d) parallelline (point3d; line3d)	Vrátí rovnoběžnou přímku vedenou bodem k jiné přímce.
<b>CrossPoint</b>	Průsečík	crosspoint (line2d; line2d) crosspoint (line3d; line3d) crosspoint (line3d; plane3d)	Vrátí průsečík dvou různoběžných přímek nebo přímky a roviny.
<b>CrossLine</b>	Průsečnice dvou rovin	crossline (plane3d; plane3d)	Vrátí průsečnici dvou různoběžných rovin.

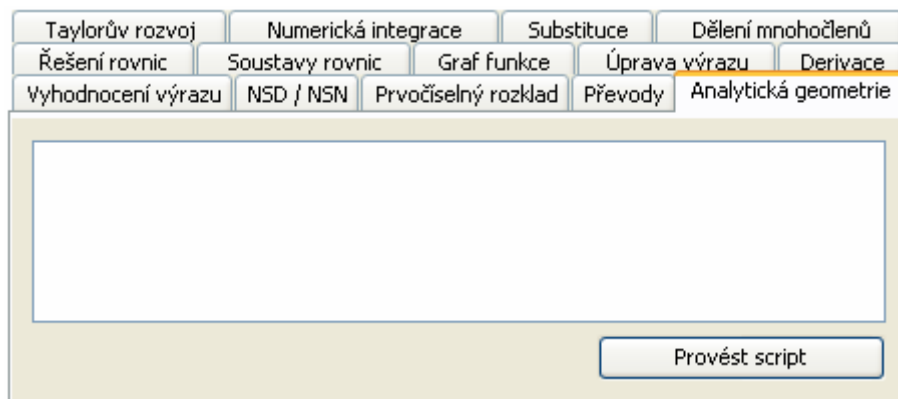
### Výpis objektu do Historie:

**Syntaxe:** print(objekt)

**Příklady:** print(vector3d("1;2;3"))  
print(p)

### Postup:

Do textového pole zapište skript a klepněte na tlačítko „Provést script“



### Příklady:

```
// Zjistí vzájemnou polohu dvou přímek v rovině
p=line2d("3*x+2";"universal";"")
q=line2d("2*x-y+1";"universal";"")
print(position(p;q))
```

```
// Zjistí vzájemnou polohu a odchylku dvou přímek
a=point3d("1;2;-1")
b=point3d("3;0;1")
c=point3d("2;-1;2")
d=point3d("5;-6;7")
p=line3d(a;b)
q=line3d(c;d)
print(position(p;q))
print(deviation(p;q))
```

```
// Z parametrického vyjádření získá obecnou rovnici roviny
ro=plane3d("1-t;-3+s;t-s";"t";"s")
print(ro;"universal")
```

```
// Z obecné rovnice přímky získá parametrické a směrnivé vyjádření
p=line2d("2*x+5*y-1";"universal";"")
print(p;"parametric")
print(p;"directional")
```

**Tip:**

Pokud na začátek skriptu dáte příkaz 3d resp. 2d pak nemusíte u deklarace objektů psát vector3d, point3d, ..., ale stačí jen vector, point ...

```
3d
v=vector("1;3;5")
a=point("4;-2;5")
print(line(a;v))
```

### 3.3.6. Ostatní funkce

#### 3.3.6.1. Převody mezi číselnými soustavami

##### Obecné informace:

Program převede jakékoli číslo (i desetinné) z jedné soustavy do druhé. Pokud nemá desetinné číslo v cílové soustavě konečný rozvoj je vypočítáno 20 platných desetinných míst.

##### Postup:

Do textového pole „Číslo“ zadejte číslo, které chcete převádět. Do pole „Ze soustavy“ zadejte číselný základ původní soustavy. Do pole „Do soustavy“ zadejte číselný základ cílové soustavy. Klepněte na tlačítko „Převést“.

Taylorův rozvoj	Numerická integrace	Substituce	Dělení mnohočlenů	
Řešení rovnic	Soustavy rovnic	Graf funkce	Úprava výrazu	Derivace
Vyhodnocení výrazu	NSD / NSN	Prvočíselný rozklad	Převody	Analytická geometrie

Číslo:	<input type="text"/>			
Ze soustavy:	<input type="text" value="10"/>	Do soustavy:	<input type="text" value="2"/>	<input type="button" value="Převést"/>

## 4. Použité postupy a algoritmy

Celá aplikace je programována v jazyce Java 5.0 (použita byla Java Virtual Machine 1.5 update 4) ve vývojovém prostředí NetBeans 4.1 a 5.0 beta.

### Jádro aplikace

Jádro aplikace tvoří abstraktní třída pro symbolickou reprezentaci výrazu (jedná se o dynamickou datovou strukturu binární strom).

```
public abstract class Expr {}
```

Od této třídy jsou odvozeny dvě abstraktní třídy určující binaritu nebo unaritu operátoru.

```
abstract class BinaryExpr extends Expr {}
abstract class UnaryExpr extends Expr {}
```

Od těchto tříd jsou odvozeny vlastní třídy reprezentující uzle stromu (operátory).

```
class AddExpr extends BinaryExpr {}
class MulExpr extends BinaryExpr {}
...
class NegExpr extends UnaryExpr {}
```

Jednotlivé listy stromu jsou reprezentované třídami pro proměnné a konstanty.

```
class VarExpr extends Expr {}
class ConstExpr extends Expr {}
```

### Parsování výrazu:

Zadaný textový řetězec (String) je procházen zleva, a pokud se narazí na operátor (ne vnořený v závorce), který má v dané fázi průchodu nejnižší prioritu, řetězec se rozdělí na dvě části (podle operátoru) a rekurzivním voláním stejné metody na levou a pravou část se dosáhne rozparsování celého výrazu do binárního stromu.

```
public static Expr parse (String expression) {

    char operator[] = {'+', '-', '*', '/', '^'};

    for (int i = 0; i < operator.length; i++) {
        for (int j = 0; j < expression.length(); j++) {
            if (expression.charAt(j) == operator[i] && notInBracket) {
                if (expression.charAt(j) == '+') {
                    // Vytvoří součet
                    // levá větev = parse(vlevo od +)
                    // pravá větev = parse(vpravo od +)
                } else if (expression.charAt(j) == '-') {
                    // Vytvoří rozdíl
                    // levá větev = parse(vlevo od -)
                    // pravá větev = parse(vpravo od -)
                } else if ...
            }
        }
    }
}
```



## Zjednodušování výrazu:

Zjednodušování výrazu je prováděno „ze spoda“ opět pomocí rekurze. Nejdříve zajistíme, aby obě větve kořenu, ve kterém se právě nacházíme, byly maximálně zjednodušené, pak teprve začneme provádět vlastní zjednodušení.

```
public Expr simplify (String expression) {  
  
    Expr leftSide = this.getLeft().simplify();  
    Expr rightSide = this.getRight().simplify();  
  
    // Vlastní zjednodušení daného výrazu  
    // Sčítáme, co se sečíst dá,  
    // Snažíme se co nejvíce o součin (vytýkaní, vzorce, ...), abychom mohli  
    // krátit ve zlomku a provádět další úpravy  
  
}
```

## Rozšiřování výrazu:

Rozšiřování výrazu pracuje na stejném principu jako zjednodušování – tedy rekurzivně „od spoda“. Rozdíl je pouze ve vlastním těle funkce, kde se naopak snažíme součin a mocniny odstranit.

## Substituce:

Rekurzivně procházíme strom a testujeme každý list, zda se nejedná o nahrazovaný výraz. Pokud ano, pak tento list nahradíme požadovanou hodnotou.

```
public Expr substitute (Expr what, Expr with) {  
    if (this.equals(what)) {  
        return with.clone();  
    } else {  
        // Pokud jsme se dostali sem, pak jsme nenašli to, co hledáme,  
        // takže budeme hledat v dalších větvích stromu.  
        // Binární operátor:  
        //     return new BinaryExpr(left.substitute(what, with),  
        //                             right.substitute(what, with));  
        // Unární operátor:  
        //     return new UnaryExpr(left.substitute(what, with));  
    }  
}
```

## Derivace:

Aplikací vět o derivaci součtu, rozdílu, součinu, podílu a složených funkcí dosáhneme rozdělení celé funkce na několik elementárních funkcí, jejichž derivace jsou známé. Tyto známé hodnoty dosadíme a rekurzí dosáhneme derivace celé funkce.

Ohled k některé proměnné znamená, že všechny ostatní proměnné ve výrazu jsou brány jako konstanty a tudíž jejich derivace je 0.

## Taylorův rozvoj:

Taylorův rozvoj n-tého stupně zjišťujeme dosazením do vzorce:

$$f(z) = \sum_{n=0}^{\infty} (z - z_0)^n \frac{f^{(n)}(z_0)}{n!}.$$

## Symbolické řešení rovnic:

Řešení spočívá v tom, že vše převedeme na levou stranu, na kterou nejdříve aplikujeme simplifikaci, abychom zajistili vyřešení i rovnic vyšších stupňů tím, že je dostaneme do součinnového nebo podílového tvaru. Pokud tedy dostaneme podílový nebo součinnový tvar, řešíme jednotlivé činitele zvlášť (rekurzivně zavoláme metodu pro řešení rovnic).

Na rovnice, které ani po simplifikaci nejsou v součinnovém nebo podílovém tvaru, nyní aplikujeme rozšíření výrazů – tím zajistíme úplné odstranění všech závorek a dostáváme polynomický tvar rovnice, který nyní standardními postupy řešíme (lineární, kvadratická, kubická, iracionální rovnice).

```
public List<Expr> solve (String variable) {  
  
    //Aplikujeme simplifikaci  
    Expr left = this.simplify();  
  
    if (left.getClass() == FraExpr.class) {  
  
        // Podílový tvar  
        // Zlomek je roven nule, pokud je čitatel roven nule  
  
    } else if (left.getClass() == MulExpr.class) {  
  
        // Součinnový tvar  
        // Součin je roven nule, pokud je aspoň jeden z činitelů roven nule  
  
    }  
  
    // Pokud jsem se dostali sem, už nepotřebujeme součin, tak ho odstraníme  
    left = left.expand();  
  
    // Nyní máme polynomický tvar rovnice  
    // Chod rozdělíme podle typu rovnice a pomocí vztahů pro kořeny daného  
    // typu tyto kořeny spočítáme a vrátíme jako seznam.  
  
}
```

## Numerické řešení rovnic:

Pro numerické řešení rovnic je použita tzv. Newtonova metoda tečen.

Hledáme všechny kořeny v zadaném intervalu, který je rozkouskován na podintervaly podle velikosti proměnné „step“ (krok). V každém tomto podintervalu jsme schopni pomocí Newtonovy metody nalézt kořen v požadované přesnosti (je zvolena pevně na  $10^{-10}$ ).

První aproximaci kořene volíme ve středu právě prohledávaného podintervalu. Každou další aproximaci počítáme podle vzorce:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Tedy od předchozího kořene odečítáme podíl funkční hodnoty původní funkce a funkční hodnoty první derivace této funkce.

## Numerická integrace:

Pro výpočet numerického integrálu funkce (obsah plochy pod křivkou grafu funkce) byla použita tzv. lichoběžníková metoda.

```

public Expr nintegrate(String variable, double lowerBound, double upperBound) {

    // Čím je větší, tím je výpočet přesnější
    long precision = 100000;

    double trape = 0;

    double interval = (upperBound - lowerBound) / precision;

    // Lichoběžníková metoda
    for (int i = 0; i < precision; i++) {
        // f(x) = funkční hodnota v bodě x
        trape += ((f(lowerBound + i) + f(lowerBound + i + 1)) / 2) * interval
    }

    return trape;
}

```

## Vyhodnocování výrazů

Nejdříve pomocí substituce nahradíme zadané proměnné jejich hodnotami a pak rekurzí získáváme hodnoty „od spoda“.

V základní třídě Expr zavedeme abstraktní metodu eval(), kterou poté implementujeme v každé třídě, která dědí od třídy Expr.

```

// Zavedení metody v hlavní třídě Expr
public abstract double eval ()

// Implementace metody v třídách pro operace
public double eval() {
    // return left.eval() (operátor např. +) right.eval();
}

// Implementace metody v třídě ConstExpr (reprezentuje číslo)
public double eval() {
    return this.value;
}

```

## Výpočet největšího společného dělitele

Výpočet je prováděn pomocí Euklidova algoritmu.

```

public static long gcd(long number1, long number2) {

    while (true) {
        if (number1 > number2) {
            number1 = number1 % number2;
            if (number1 == 0) {
                return number2;
            }
        } else {
            number2 = number2 % number1;
            if (number2 == 0) {
                return number1;
            }
        }
    }
}

```

## 5. Závěr

V porovnání s ostatními, podobně zaměřenými aplikacemi, má Smart Counter výhodu především v tom, že je šířen jako freeware, takže jej může kdokoli legálně používat jak pro domácí tak i pro komerční účely. Toto je důležité především pro cílovou skupinu tohoto programu – studenty, kteří si drahé placené programy nemohou dovolit. Další velkou výhodou je naprosto jednoduché a intuitivní uživatelské rozhraní, díky němuž se Smart Counterem dokáže pracovat i naprostý laik v oboru informačních technologií. Toto rozhraní bylo navrženo především na základě ohlasů k první verzi Smart Counteru (je k dispozici na internetu od prosince 2004, odkud si ji stáhlo několik tisíc uživatelů) od skutečných uživatelů, kteří si mnohdy stěžovali na složitost první verze.

Projekt je neustále živý a na dalším vývoji se pracuje. Nová verze, která je nyní ve vývoji, je zaměřena především na rozvoj části Analytické geometrie, kde přibude mnoho dalších objektů (především kvadratických – kuželosečky). Dále přibude možnost vykreslení celé scény (2D / 3D / Mongeova projekce), což znatelně pomůže slabším studentům při řešení úloh, neboť nebudou mít problém si situaci představit. Na základě několika návrhů k nové verzi (je vystavena na většině velkých českých download serverech od února 2006), především ze strany pokročilejších uživatelů, přibude v programu k jednoduchému rozhraní ještě pokročilé uživatelské rozhraní s příkazovým řádkem a dialogovými okny, které umožní velmi rychlou a efektivní práci s programem těm, kteří jsou na tento typ rozhraní zvyklí.

## Seznam literatury

WEISSTEIN, Eric W. *MathWorld*. Dostupné z URL: <http://mathworld.wolfram.com/>

BARTSCH, Hans-Jochen: *Matematické vzorce*. 3. vyd. Praha: Mladá fronta, 1996.

HRUBÝ, Dag. KUBÁT, Josef: *Matematika pro gymnázia – Diferenciální a integrální počet*. 1. vyd. Praha: Prométheus, 1997.

KOČANDRLE, Milan. BOČEK, Leo: *Matematika pro gymnázia – Analytická geometrie*. 2. vyd. Praha: Prométheus, 1997.

WRÓBLEWSKI, Piotr: *Algoritmy – Datové struktury a programovací techniky*. 1.vyd. Brno: Computer Press, 2004.

BENEŠ, Miroslav: *Programovací techniky*. Dostupné z URL: <http://www.cs.vsb.cz/benes/vyuka/pte/texty/index.php>

W3C, *Mathematical Markup Language (MathML) Specification*. Dostupné z URL: <http://www.w3.org/TR/MathML2/>

W3C, *Scalable Vector Graphics (SVG) Specification*. Dostupné z URL: <http://www.w3.org/TR/SVG/>

W3C, *An XHTML + MathML + SVG Profile Specification*. Dostupné z URL: <http://www.w3.org/TR/2002/WD-XHTMLplusMathMLplusSVG-20020809/>