

Vědecká kalkulačka

KIV/MKZ – semestrální práce

Příjmení a jméno:	Kozler Petr
Univerzitní číslo:	A13B0359P
E-mailová adresa:	pkozler@students.zcu.cz
Datum vypracování:	3. 4. 2017

1. Zadání

Cílem této semestrální práce bude vytvoření mobilní aplikace pro operační systém Android. Aplikace bude sloužit jako **Vědecká kalkulačka** – bude umožňovat vyhodnocování výrazů zahrnujících běžně používané matematické funkce, zobrazování průběhu funkce jedné reálné proměnné do grafu (ve kterém bude možné se pohybovat pomocí přesunů po obrazovce a měnit měřítko zobrazení) a hledání řešení nelineární rovnice pomocí numerické metody s možností nastavení intervalu a přesnosti hledání řešení.

2. Programátorská dokumentace

Program byl vytvořen pro operační systém Android ve verzi 6.0 (API level 23).

2.1. Použité třídy a metody

- `Evaluator` – hlavní třída jádra aplikace; slouží pro zpracovávání a vyhodnocování řetězcových reprezentací zadaných matematických výrazů
- `CalculatorContext` – třída představující aplikační kontext; uchovává komponenty pro manipulaci s ovládacími prvky grafického uživatelského rozhraní, ke kterým je přístupováno globálně z většího počtu aktivit aplikace
- `BaseActivity` – abstraktní třída představující obecnou aktivitu; poskytuje základní funkcionalitu a ovládací prvky (menu, chybové hlášky) společné pro většinu aktivit
- `CalcActivity` – aktivita, představující sekci pro výpočet zadaných matematických výrazů
- `PlotActivity` – aktivita, představující sekci pro vizualizaci průběhu zadaných matematických funkcí do grafu
- `SolveActivity` – aktivita, představující sekci pro hledání řešení zadané obecné rovnice pomocí numerické metody se zadaným intervalem a počtem kroků pro hledání řešení
- `PlotView` – třída, představující vlastní grafickou komponentu; slouží jako plátno pro kreslení grafu matematické funkce
- `BisectionEquationSolver` – třída, sloužící k provádění algoritmu hledání řešení rovnice metodou půlení intervalu

2.2. Použité algoritmy

Pro snadné vyhodnocení jsou zadané matematické výrazy nejprve převáděny z běžného zápisu, ve kterém se operátory nacházejí mezi operandy (tzv. in-fixový zápis) do zápisu, při kterém se operátory umísťují za operandy (post-fixový zápis). K tomuto převodu se používá tzv. **Shunting-Yard algoritmus**, jehož průběh je následující:

- pro každý token ze vstupní fronty:
 - pokud je token číslo nebo proměnná:
 - vlož ho do výstupní fronty
 - jinak pokud je token funkce:
 - vlož ho na zásobník
 - jinak pokud je token oddělovač argumentů funkce:

- dokud není na vrcholu zásobníku levá závorka:
 - vyjmi z něj token a vlož ho do výstupní fronty
- pokud je zásobník prázdný (nebyla nalezena levá závorka):
 - chyba: oddělovač byl na nesprávném místě, nebo ve vstupu chybí otevírací závorka
- jinak pokud je token operátor o1:
 - dokud je na vrcholu zásobníku operátor o2 a dokud zároveň platí, že o1 je asociativní zleva a jeho priorita je menší nebo stejná jako priorita o2, nebo že o1 je asociativní zprava a jeho priorita je nižší než priorita o2:
 - vyjmi o2 ze zásobníku a vlož ho do výstupní fronty
 - vlož o1 na zásobník
- jinak pokud je token levá závorka:
 - vlož ho na zásobník
- jinak pokud je token pravá závorka
 - dokud není na vrcholu zásobníku levá závorka:
 - vyjmi operátor ze zásobníku a vlož ho do výstupní fronty
 - pokud je zásobník prázdný (nebyla nalezena levá závorka):
 - chyba: ve výrazu se nacházejí neotevřené závorky
 - vyjmi ze zásobníku levou závorku (nevkládej do výstupní fronty)
 - pokud je na vrcholu zásobníku funkce:
 - vlož ji do výstupní fronty
- dokud jsou tokeny v zásobníku:
 - pokud je token operátor nebo funkce:
 - vyjmi ho ze zásobníku a vlož do výstupní fronty
 - jinak (token na vrcholu zásobníku je závorka):
 - chyba: ve výrazu se nacházejí neuzavřené závorky

Pro samotné **vyhodnocení výrazu převedeného do post-fixové notace** (také nazývána jako RPN – reverzní polská notace) se používá následující algoritmus:

- pro každý token ze vstupní fronty:
 - pokud je token číslo:

- vlož ho na zásobník
- jinak pokud je token proměnná:
 - dosad' hodnotu za proměnnou a vlož ji na zásobník
- jinak pokud je token (binární) operátor:
 - pokud nejsou na vrcholu zásobníku alespoň 2 hodnoty:
 - chyba: nebyl zadán dostatečný počet operandů
 - vyjmi ze zásobníku dvě čísla, proved' odpovídající výpočet a výsledné číslo vlož na zásobník
- jinak pokud je token funkce:
 - pokud je na zásobníku méně čísel, než je minimální povolený počet argumentů funkce:
 - chyba: nebyl zadán dostatečný počet parametrů
 - dokud jsou na zásobníku čísla a velikost pomocného zásobníku nepřesáhne maximální povolený počet argumentů funkce:
 - vyjmi číslo z hlavního zásobníku a vlož jej do pomocného zásobníku (slouží ke shromáždění argumentů funkce)
 - vypočti hodnotu dané funkce s parametry předanými z pomocného zásobníku a výsledné číslo vlož na hlavní zásobník
- pokud je na zásobníku více než jedno číslo:
 - chyba: byl zadán přebytečný počet hodnot
- vyjmi číslo ze zásobníku (jeho hodnota je výsledek výpočtu)

Vykreslení průběhu funkce jedné reálné proměnné do grafu v kartézské soustavě souřadnic je použita **metoda elementárních úseček**, jejíž algoritmus vypadá následovně:

- ulož souřadnici x_0 počátečního bodu spojnice na levém okraji zobrazené části grafu
- vypočítej souřadnici y_0 počátečního bodu spojnice
- ulož souřadnici x (posunutou podle aktuálního měřítka) koncového bodu spojnice
- dokud se souřadnice x koncového bodu spojnice nenachází za pravým okrajem zobrazené části grafu, opakuj:
 - vypočítej souřadnici y koncového bodu spojnice
 - nakresli aktuální spojnicí
 - ulož souřadnici x jako novou souřadnici x_0

- ulož souřadnici y jako novou souřadnici y_0

Pro numerické hledání řešení nelineárních rovnic byla použita snadno implementovatelná **metoda půlení intervalu** (jinak též známa jako metoda bisekce) – tato metoda očekává nelineární rovnici převedenou z obecného tvaru $L(x)=R(x)$ do tvaru $L(x)-R(x)=0$; algoritmus poté probíhá následovně:

- vypočítej funkční hodnotu $f(a)$ dolní meze intervalu hledání řešení a
- vypočítej funkční hodnotu $f(b)$ dolní meze intervalu hledání řešení b
- dokud není rozdíl mezi hodnotami a a b menší než stanovená maximální odchylka ε :
 - ulož aritmetický průměr hodnot a a b jako hodnotu c
 - vypočítej funkční hodnotu $f(c)$ z hodnoty c
 - je-li hodnota $f(c)$ rovna 0:
 - vrať hodnotu c jako přesné řešení rovnice x
 - je-li hodnota $f(c)$ větší než 0:
 - ulož hodnotu c jako novou dolní mez a
 - je-li hodnota $f(c)$ menší než 0:
 - ulož hodnotu c jako novou horní mez b
- vrať aritmetický průměr aktuálních hodnot a a b jako odhad řešení rovnice x

3. Uživatelská dokumentace

Aplikace obsahuje úvodní (uvítací) aktivitu a 3 aktivity představující sekce, v nichž jsou prováděny jednotlivé druhy podporovaných matematických operací. Mezi sekcemi pro výpočty je možné se přesouvat pomocí výsuvného menu v záhlaví uživatelského rozhraní aplikace. Každá z těchto sekcí je rozdělena na horní část (nalevo při otočení displeje na horizontální orientaci) s textovými poli pro zadávání vstupu a komponentami pro zobrazování výstupu, a dolní část (napravo při horizontální orientaci) s klávesnicí pro rychlé zadávání vstupu, která je pro všechny tyto sekce společná. Klávesnice obsahuje několik režimů zadávání, mezi kterými je možné se přepínat pomocí vyhrazených tlačítek. K přepínání mezi zadáváním čísel a operátorů a zadáváním funkcí slouží tlačítko označené $f(x)/\#$. K přepínání částí pro zadávání dostupných typů funkcí v režimu zadávání funkcí slouží tlačítko $1/2/2/2$. Část pro zadávání goniometrických funkcí obsahuje dále tlačítko pro přepínání úhlových jednotek, se kterými tyto funkce počítají, a které je označeno podle aktuálně zvolených jednotek popisem DEG/RAD/GRAD.

3.1. Ovládání aplikace

Úvodní obrazovka: Obsahuje 3 tlačítka pro přesun do jedné ze sekcí pro matematické operace.

Sekce pro výpočet výrazů: Obsahuje textové pole pro zadávání matematického výrazu, který má být vyhodnocen, a dále výstupní textové pole pro zobrazení výstupu. Výrazy jsou vyhodnocovány automaticky při zadávání vstupu a výsledky jsou ihned vypisovány do vyhrazeného výstupního textového pole.

Sekce pro zobrazení funkcí: Obsahuje taktéž vstupní textové pole pro zadání funkce jedné reálné proměnné x , dále tlačítko *Zobrazit* pro ruční spuštění vyhodnocování a vizualizace průběhu zadané funkce, výstupní textové pole pro výpis výsledku operace a plátno pro současné zobrazení grafu, pokud je zadaná funkce platná. Po zobrazení grafu je možné se v něm přesouvat pohybem po obrazovce. Vedle grafu se nacházejí 2 tlačítka, kterými lze změnit měřítko. Vycentrování a nastavení původního měřítka lze docílit opětovným stiskem tlačítka *Zobrazit*.

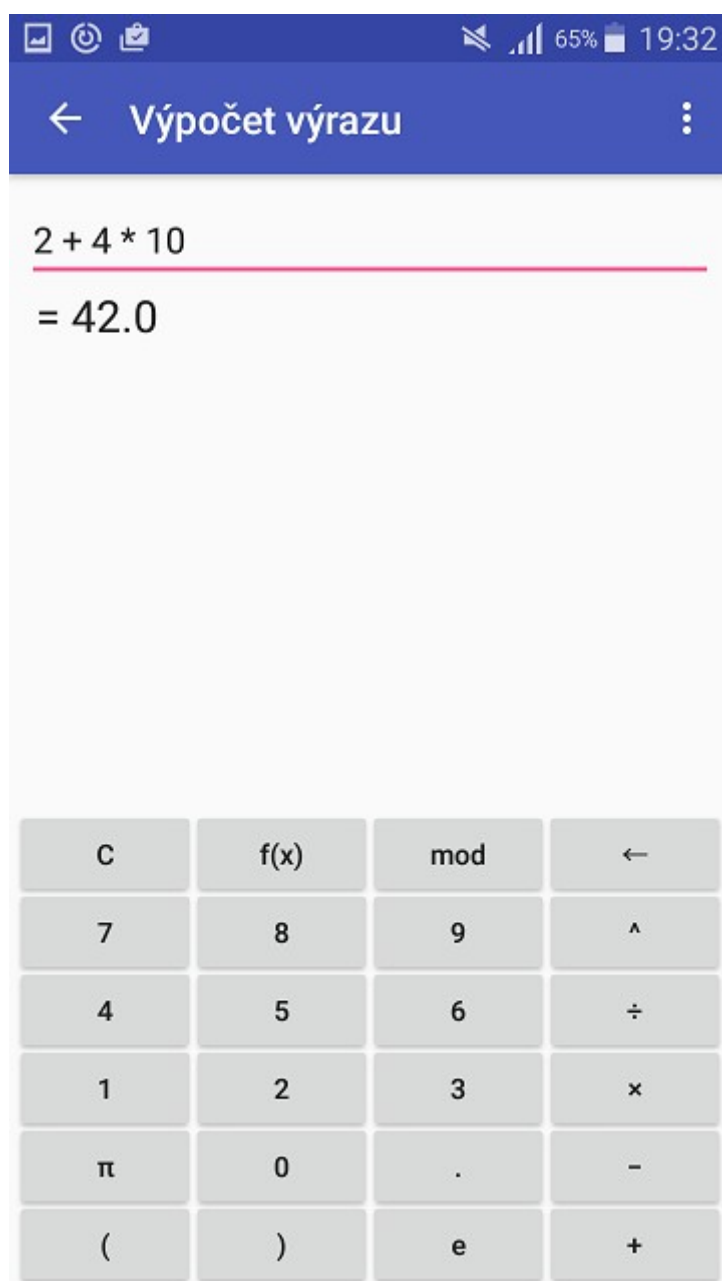
Sekce pro řešení rovnic: Obsahuje v horní části řádek s 2 vstupními textovými poli pro zadání jednotlivých stran obecné rovnice o jedné proměnné x . Pod ním se nachází řádek s dalšími 3 vstupními textovými poli pro zadání jednotlivých číselných parametrů. První 2 textová pole slouží k zadání desetinných čísel představujících dolní a horní meze celkového intervalu, v němž jsou hledána řešení zadané rovnice. Poslední textové pole slouží k zadání přirozeného čísla představujícího „jemnost“ hledání řešení – počet pod-intervalů stejné délky, do kterých je rozdělen celkový interval, přičemž hledání řešení je spuštěno v každém zvlášť (zadané číslo tedy odpovídá počtu opakování algoritmu hledání řešení), což umožňuje nalezení vícero nebo všech možných řešení u rovnic, které mají více než jedno jednoznačné řešení. Následuje řádek s tlačítkem *Řešit* pro ruční spuštění hledání řešení s nastavenými parametry a výstupním polem pro výpis výsledku operace. V dolní části se nachází komponenta pro výpis seznamu jednotlivých nalezených jednoznačných řešení zadané rovnice (pokud byly v zadaném intervalu nalezeny).

3.2. Screenshots aplikace

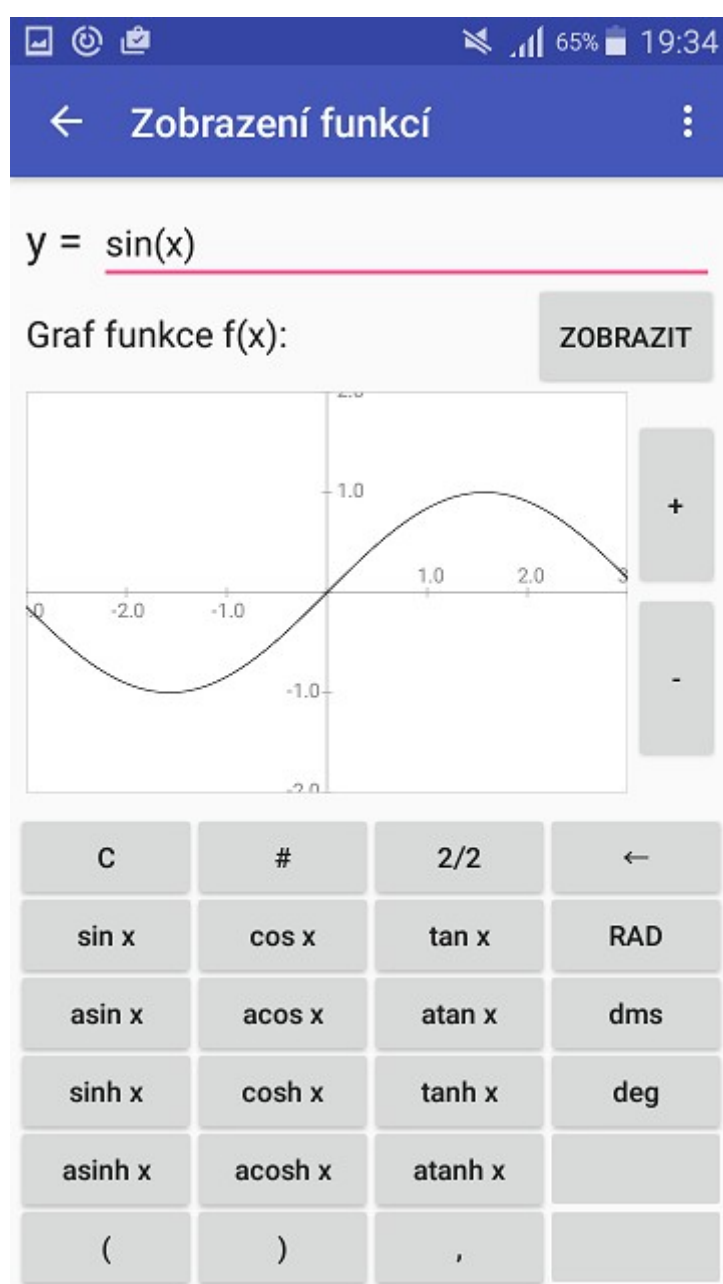
Úvodní obrazovka:



Sekce pro výpočet výrazů:



Sekce pro zobrazení funkcí:



Sekce pro řešení rovnic:

← Řešení rovnic

$x^2 = 4$

Interval od -4 do 4, kroků: 2

ŘEŠIT Nalezená řešení x:

-2.0

2.0

C	#	1/2	←
x^2	x^3	x^y	[x]
\sqrt{x}	$\sqrt[3]{x}$	$\sqrt[y]{x}$	$x-[x]$
x^{-1}	$\ln x$	$\log_a x$	[x]
x	e^x	n!	$\lceil x \rceil$
()	,	sgn x

4. Řešené problémy

Během vytváření aplikace bylo nutné čelit několika ne zcela triviálním problémům s realizací numerických metod výpočtů a ovládání prvků grafického uživatelského rozhraní.

4.1. Výpočty

Úprava řetězce výrazu pro zpracování a vyhodnocení:

Matematické výrazy je možné zadávat rozdílnými způsoby – s mezerami nebo bez nich, s velkými nebo malými písmeny (u názvů funkcí), lze vynechávat operátory násobení mezi čísly a závorkami apod. Je vhodné tyto rozdíly odstranit pro zjednodušení dalšího zpracování. Z tohoto důvodu jsou zadané řetězcové reprezentace matematických výrazů předzpracovány. Předzpracování se skládá z několika jednoduchých kroků:

- převod velkých písmen na malá
- odstranění bílých znaků
- přepis záporných znamének před závorkami na výraz násobení číslem 1
- doplnění operátorů násobení mezi číslem a závorkou nebo mezi závorkami
- odstranění přebytečných kladných znamének u hodnot nebo před závorkami
- přepis záporných znamének na symbol představující funkci „negace“ pro odlišení od operátoru odčítání

Ošetření singulárních bodů při vizualizaci průběhu funkce:

Při vykreslování průběhu matematické funkce metodou elementárních úseček nastává problém v singulárních bodech, kde funkce není definována – mezi sousedními body je vytvořena svislá „spojnice“.

Z tohoto důvodu je zabráněno zobrazení (barva úsečky v tomto místě je nastavena na průhlednou) v singulárních bodech, kde počáteční i koncový bod elementární úsečky mají stejnou souřadnici y . Řešení není dokonalé, neboť je problematické rozlišení mezi případy singulárních bodů a takových částí funkce, které se v grafu svým průběhem svislým čarám pouze podobají.

Identifikace rovnic s nulovým nebo nekonečným počtem řešení:

Při řešení rovnic je možné narazit na speciální případy, kdy zadaná rovnice nemá v prohledávaném intervalu žádné řešení, nebo jich je naopak nekonečný počet.

Detekce případu žádného řešení je triviální – rovnice na daném intervalu nemá řešení, pokud pro každý dílčí interval platí, že součin funkčních hodnot jeho mezí není záporný, tj. $f(a) \cdot f(b) > 0$.

Detekce nekonečně mnoha řešení je problematictější – nelze ověřit nekonečný počet kořenů a zároveň je nutné odlišit situaci, kdy má funkce nekonečný počet kořenů od situace, kdy jich má pouze „větší počet“. Pravděpodobný nekonečný počet kořenů je detekován v případě, kdy ve všech dílčích intervalech jsou funkční hodnoty obou mezi velmi blízké nule (nižší než stanovená maximální odchylka ε od přesného jednoznačného řešení x). Řešení je samozřejmě pouze přibližné, a jeho spolehlivost je závislá na zvolených parametrech hledání řešení.

4.2. Ovládání grafických prvků

Současný posun, změna měřítka a centrování grafu pomocí dotyků na obrazovce:

Při implementaci posluchačů událostí nad ovládacími prvky byly zkoumány možnosti zachytávání více druhů gest současně – například přesunu po obrazovce po pohyb v grafu, multi-touch pro změnu měřítka a poklepání na obrazovce pro vycentrování grafu. Pro obtížnost implementace bylo nakonec zvoleno pouze odchyťávání přesunu. Ke změně měřítka a centrování grafu slouží vyhrazená tlačítka.

5. Testování

Testování aplikace bylo prováděno pomocí emulátoru na virtuálním zařízení Nexus 5 API 23 a na reálném zařízení Samsung Galaxy S5 Mini s verzí Androidu 6.0.1.

6. Závěr

Podařilo se splnit všechny základní body zadání. V těchto bodech nebyly při testování aplikace odhaleny žádné chyby ve funkčnosti. Při testování na reálném zařízení nedocházelo k problémům se spouštěním. Zároveň byl běh aplikace o poznání rychlejší na reálném zařízení oproti emulátoru, což odpovídalo původním očekáváním. Zdrojový kód byl psán s ohledem na rozšiřitelnost a znovupoužitelnost některých komponent (např. pro zpracování a vyhodnocování výrazů) v jiných budoucích aplikacích. Samotná vytvořená vědecká kalkulačka poskytuje velké množství prostoru k rozšíření o další podporované matematické operace, např. numerické integrování výpočty s maticemi a další.