

## Dokumentácia zápočtového programu

### 1. Anotácia

Zápočtový program v podobe jednoduchkej kalkulačky pracuje v grafickom prostredí tkinter. Výraz do kalkulačky zadávame prostredníctvom tlačidiel vykreslených v grafickej ploche alebo prostredníctvom klávesnice. Kalkulačka vykonáva základné operácie (+ - \* / ^) a vyhodnocuje výraz pomocou prevodu do postfixovej notácie. Dokáže pracovať s viaccifernými aj desatinnými číslami. Zadávaný výraz podobne ako výsledok výrazu po vyhodnutení vidíme v grafickej ploche.

### 2. Algoritmus

Na vyhodnocovanie výrazu som zvolila algoritmus prevodu výrazu z infixovej do postfixovej notácie a následné vyhodnotenie výrazu v postfixovej notácii. Prvá časť sa prevedie v čase  $O(N)$  a druhá časť tiež v čase  $O(N)$ . Celková zložitosť algoritmu je teda  $O(N)$ .

#### a. Prevod výrazu z infixovej do postfixovej notácie

Na vstupe máme zadaný výraz v infixovej notácii. Prevedieme jeden priechod výrazom zľava doprava postupne po znakoch. Na ukladanie operátorov a zátvoriek používame zásobník. Ak je znak:

- číslo: zapíšeme ho priamo na výstup
- ľavá zátvorka: vložíme ju do zásobníku
- pravá zátvorka: zo zásobníku odstraňujeme a zapisujeme na výstup postupne všetky operátory až k najbližšej ľavej zátvorke, túto zátvorku zo zásobníka odstránime  
(v postfixovej notácii teda nie sú zapísané žiadne zátvorky)
- operátor: zo zásobníka odstraňujeme a zapisujeme na výstup všetky operátory vyššej alebo rovnakej priority ako náš operátor po najbližšiu uloženú ľavú zátvorku, potom náš operátor vložíme do zásobníku

Na konci zo zásobníku odstraňujeme a zapisujeme na výstup postupne všetky uložené operátory.

#### b. Vyhodnotenie výrazu v postfixovej notácii

Máme výraz v postfixovej notácii. Prevedieme jeden priechod výrazom zľava doprava postupne po znakoch. Na ukladanie čísel používame zásobník. Ak je znak:

- číslo: vložíme ho do zásobníku
- operátor: zo zásobníku odstránime posledné dve čísla a prevedieme s nimi danú operáciu, výsledok zapíšeme na koniec zásobníka  
(Keďže čísla odstraňujeme z konca zásobníku, sú v opačnom poradí, pri vykonávaní operácií teda najprv odoberáme pravý operand, potom ľavý.)

Na konci ostane v zásobníku jediné číslo – hodnota výrazu.

### 3. Výber algoritmu

Algoritmus prevodu infixovej notácie do postfixovej a jej následné vyhodnotenie pracuje efektívne, preto som sa rozhodla ho zvoliť. Dve časti algoritmu by mohli bežať aj súbežne (priebežné vyhodnocovanie postfixovej notácie počas prevádzania infixovej), ale pre lepšiu prehľadnosť programu som sa rozhodla tieto dve časti oddeliť. Program teda vykonáva najprv jednu a potom druhú.

### 4. Program

V programe je definovaná samostatná funkcia pre každý znak, ktorý môžeme do výrazu zadávať (cifry 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, operátory +, -, \*, /, ^, zátvorky a desatinná čiarka). Tieto funkcie vždy pripisujú daný znak na koniec výrazu, ktorý zadávame. Tieto funkcie sa spúšťajú buď pomocou tlačidiel v grafickej ploche alebo priamo zadávaním z klávesnice. Funkcie sú definované na začiatku programu, zviazanie s tlačidlami a klávesnicou je na konci programu. Väčšinu programu tvorí hlavná funkcia VYHODNOT, ktorá sa spúšťa po stlačení tlačidla „rovná sa“, resp. klávesy „rovná sa“ alebo „enter“. Funkcia vykonáva postupne:

- Prevedie si znaky „ $\times \div$ “ na znaky „\* /“. Program totiž dokáže pracovať iba so znakmi „\* /“, pre užívateľa je však vizuálne príjemnejšie (a aj zaužívanejšie) zadávať znaky „ $\times \div$ “. Toto vykoná jedným priechodom zadaným výrazom zľava doprava postupne po znakoch.
- Rozdelí si zadaný výraz na operátory (5 základných) a operandy (môžu to byť viacciferné čísla a môžu obsahovať desatinnú čiarku). Toto vykoná jedným priechodom zadaným výrazom zľava doprava postupne po znakoch.
- Jednoduchým algoritmom overí, či je výraz správne uzátvorkovaný pomocou zásobníka. Toto vykoná jedným priechodom zadaným výrazom zľava doprava postupne po znakoch.
- Prevedie prevod výrazu z infixovej do postfixovej notácie. (Popísané v sekcii 2.)
- Vyhodnotí výraz prevedený do postfixovej notácie. (Popísané v sekcii 2.)
- Overí, či výraz neobsahuje delenie nulou.
- Overí, či je výsledok desatinný alebo celé číslo a podľa toho vypíše výsledok v adekvátnom type.

### 5. Alternatívne riešenia programu

Jednotlivé „podfunkcie“ funkcie VYHODNOT by bolo možné rozdeliť do samostatných funkcií, prípadne tlačidlu „rovná sa“ definovať triedu a v nej tieto jednotlivé funkcie. Zdalo sa mi to však zbytočné, keďže kalkulačka vykonáva tieto algoritmy aj tak postupne (vysvetlené v sekcii 4.) a všetky potrebujú byť zavolané po stlačení jedného tlačidla.

### 6. Vstupné dáta

Užívateľ zadáva vstupné dáta priamo pomocou grafického prostredia a tlačidiel, ktoré sú v ňom vykreslené alebo priamo z jeho klávesnice. Je to celkom intuitívne, preto sa mu o tom ani nezobrazuje žiadna informácia.

### 7. Výstupné dáta

Užívateľ uvidí výstup (typu int alebo typu float) na „displeji kalkulačky“ priamo v grafickom prostredí. Výstup predstavuje výsledok predtým zadávaného výrazu. Výsledok sa objaví po stlačení tlačidla „rovná sa“ v grafickej ploche alebo po stlačení klávesy „enter“ alebo klávesy „rovná sa“. Tento výsledok sa uloží a ďalej sa zobrazuje ako prvé číslo z ďalšieho zadávaného výrazu. Ak chce užívateľ zadávať úplne nový výraz (nechce vykonávať žiadnu operáciu so získaným výsledkom), tak musí tento výsledok zmazať (pomocou tlačidla „del“ alebo pomocou klávesy „backspace“). Toto vždy maže posledný znak výrazu (teda cifru, operátor alebo zátvorku), resp. výsledku.

## 8. Priebeh písania programu

- a. Základná grafika, návrh tlačidiel, hrubé usporiadanie v grafickej ploche – pre predstavu výzoru užívateľského prostredia. Základné funkcie pre tlačidlá na zadávanie vstupu.
- b. Hlavná časť programu – algoritmy. Funkcia VYHODNOT a jej prvých 5 častí (bližšie popísané v sekcii 4.):
  - prevedenie znakov „ $\times \div$ “, na „ $*$  / “
  - rozdelenie výrazu na operátory a operandy
  - overenie správnosti uzátvorkovania
  - prevod z infixovej notácie do postfixovej
  - vyhodnotenie výrazu v postfixovej notácii
- c. Doladovanie grafiky (tlačidiel).
- d. Doplnenie funkcie VYHODNOT o overenie, či sa vo výraze nevyskytlo delenie nulou.
- e. Doplnenie funkcie VYHODNOT o overenie typu výstupu, aby program nevracal výsledok v tvare celého čísla ako desatinné (teda napr. aby namiesto 36,0 vracal 36).
- f. Zviazanie funkcií, ktoré vykonávajú jednotlivé tlačidlá kalkulačky, s klávesmi klávesnice.

## 9. Nedokončené časti programu

Z dôvodu nedostatku času (či jeho zlej organizácie) program neobsahuje niektoré funkcie, ktoré som pôvodne zamýšľala. Napríklad ďalšie matematické operácie ako logaritmus, odmocnina, modulo, trigonometrické funkcie,...

## 10. Záver

Pri písaní programu som sa naučila veľa nového – ako vlastne fungujú notácie, prečo je dôležitá časová zložitosť a veľa syntaktických záležitostí. Program sa mi síce zdá dosť jednoduchý, no keďže som v programovaní začiatočník, tak som s touto prácou aj tak celkom spokojná.