

Ošetření vstupu v jazyce C

©2023
Petrásek Jan

Obecně

- Matkou tohoto přístupu je Margaret Hamilton
- Její tvrzení: „Lidé chybují a stroje s tím musí počítat.“ (volný překlad)
- Může se stát, že na vstup uživatel místo čísla zadá písmena
- Je potřeba ověřit, že vstupem je číslo, když jím má být.
- Ve vyšších jazycích se tento problém řeší pomocí přetypování vstupu v try catch bloku.
- V C není k dispozici try catch konstrukce

Jedno z řešení v C

- **Zjistit, zda funkce scanf() vrátila 1**

- #include <stdio.h>
#include <stdbool.h>

```
void main(){  
    int celeCislo;  
    do{  
        printf("Zadej cele cislo: ");  
        if(scanf("%d",&celeCislo)==1){  
            break; //ukončení cyklu v případě úspěchu  
        }else{  
            printf("Na vstupu není číslo, opakuj zadání\n");  
            getchar();  
        }  
    }while(true); printf("Zadané číslo je %d.", celeCislo);}
```

Řešení v C

- Řešení z minulého snímku si neporadí s mezerami kolem čísla
- Řešením je vlastní konverzní funkce, která do druhého řetězce vykopíruje jen číselný obsah a znaménko
- Vlastní konverzní funkce umožní využívat také desetinnou čárku místo tečky pro datové float a double.
- Příklad vlastní konverzní funkce je v projektu OsetreniVstupu2
- Posledním bodem ošetření vstupu je ošetřit, zda zadaná hodnota odpovídá mezím, které program očekává

Vlastní konverzní funkce

1. Očištění od bílých znaků před číslem
2. Detekce záporné hodnoty (znaménka mínus)
3. Detekce vlastního čísla
4. Očištění od bílých znaků za číslem
5. Konverze stringu na číslo
 - A. Datový typ int => knihovní funkce atoi (stdlib.h)
 - B. Datový typ float a double => funkce atof (stdlib.h)
 - Nutno detekovat desetinnou tečku
 - C. Konverze pomocí matematické funkce