

# Cvičenie 3

## 2. úloha

- nepovinná, ale jednoduchá
- deadline o 7 dní
- dotazy?

## Opakovanie C

- odpovedník v IS (thanks to J. Bayer)

# 3. úloha

- odovzdanie povinné!

/export/e1/pb161/váš\_login/cvic3

- deadline o 14 dní

- bodovanie

+2 body – odovzdanie do nedele

+1 bod – bonusové rozšírenie (*rand()*)

- zadanie

<http://www.fi.muni.cz/usr/jkucera/pb161/hadani.htm>

- vzorové riešenie

/home/jkucera/pb161/cvic3/hadani

# 3. úloha

- poznámky (prevzaté zo zadania):
  - V programu môžete očakávať, že jako vstup bude vždy poskytnuta typově korektní hodnota (tj. vždy číslice, nikoli např. řetězec). Číslice však může být mimo možný rozsah voleb v nabídce.
  - Vzorové řešení neopisuje vstup. Porušení této konvence bude tolerováno i u odevzdaných příkladů, jelikož jde v tomto případě o čistě interaktivní program (u kterých opisování vstupu ztrácí smysl).

# C++ vs. C: Čo *nie* je v C++?

- funkcie podľa Kernighana a Ritchieho

```
int funkcia(a, b)
int a;
int b;
{ /*...*/ }
```

- deklarácia typu návratovej hodnoty v deklarácii funkcie

```
struct struktura {
    double y;
} funkcia(void) { /*...*/ }
```

- implicitný návratový typ *int*

– v C++ návratový typ povinný, pre konštruktor a deštruktor zakázaný

# C++ vs. C: Čo *nie* je v C++?

- volanie funkcie bez známeho prototypu, s menším počtom parametrov ako v deklarácii

```
int main(void) {  
    printf ("%d", funkcia(1));  
    return 0;  
}  
  
int funkcia (int a, int b) {  
    return -a;  
}
```

– v C++ máme ale implicitné parametry

```
int funkcia (int a, int b = 0) {  
    return -a;  
}  
  
int main(void) {  
    printf ("%d", funkcia(10));  
    return 0;  
}
```

# C++ vs. C: Čo *nie* je v C++?

- skok cez deklaráciu s inicializáciou

```
switch(cislo) {  
    case 1 : printf("1");  
             int test = 1;  
             break;  
    case 5 : printf ("5");  
             break;  
}
```

- v C++ potrebujem obaliť do bloku

```
switch(cislo) {  
    case 1 : {  
                printf("1");  
                int test = 1;  
                break;  
            }  
    case 5 : printf ("5");  
             break;  
}
```

# C++ vs. C: Čo *nie* je v C++?

- *main()* nejde volať rekurzívne
- otvorené pole v štruktúrach (v C++ objektový typ)

```
struct S {  
    double y;  
    char c[];  
};
```

- nutnosť zastaralej deklarácie typu

– v C

```
typedef struct { double y; } ST; ST xx;  
struct ST x;
```

– v C++

```
struct ST { double y; }; ST x;
```



# C++ vs. C: Čo *nie* je v C++?

- pôvodný význam neuvedených parametrov funkcie
  - v C nehovoríme o parametroch nič
  - v C++ je neuvedenie parametrov chápané ako *void*
    - => môže dôjsť k volaniu inej funkcie, ako sme plánovali
    - => treba uvádzať *void*
- v C++ vytvárajú operátory „?:“ hodnotu, ktorá môže stáť pri priradení vľavo
  - (x1 < max ? x1 : x2) = 100;
  - ak x1 < max, 100 sa priradí do x1, inak do x2

Príjemnú prácu na úlohe!

Dotazy?