

# Základy programování v C++ - 7. cvičení

Zuzana Petříčková

23. října 2018

# Přehled

- 1 Bylo minule
  - Skoky
- 2 Znaky
- 3 Příkaz switch

# Už jsme probírali

- proměnné a některé základní datové typy
- rozdělení programu na podprogramy (funkce)
- Příkazy pro řízení běhu programu
  - podmíněné bloky (podmínky)
  - cykly (smyčky)
  - skoky
- Ošetření chyb uživatele

# Řízení běhu programu - skoky

## **break:**

- Ukončí cyklus, program následuje prvním příkazem za cyklem
- Přerušuje se jen nevnitřnější cyklus

## **continue:**

- Ukončí aktuální iteraci cyklu a započne novou
- Přerušuje se jen nevnitřnější cyklus

## **return:**

- Ukončí aktuální funkci, může vrátit hodnotu

## **goto:**

- Přesun řízení na zadané návěští
- Nemohu přeskakovat mezi funkcemi

# Řízení běhu programu - skoky

## Příklad

```
for (int i=0; i<5; i++)
{
    for (int j=0; j<5; j++)
    {
        if (i == 2 && j == 2)
            break;
        cout << "(" << i << ", " << j << ") , ";
    }
    cout << endl;
}
```

## Zkrácené vyhodnocení podmínek

- konjunkce: je-li první podmínka nepravda, je celý výrok nepravda (druhá podmínka se nevyhodnocuje)
- disjunkce: je-li první podmínka pravda, je celý výrok pravda (druhá podmínka se nevyhodnocuje)

## Jak přerušit vnější cyklus z vnitřního?

# Řízení běhu programu - skoky

## Příklad: Jak přerušit vnější cyklus z vnitřního I.

```
bool konec = false;
for (int i=0; i<5; i++)
{
    if (konec)
        break;
    for (int j=0; j<5; j++)
    {
        if (i == 2 && j == 2)
        {
            konec = true;
            break;
        }
        cout << "(" << i << " , " << j << ") , ";
    }
    cout << endl;
}
```

# Řízení běhu programu - skoky

## Příklad: Jak přerušit vnější cyklus z vnitřního II.

```
for (int i=0; i<5; i++)
{
    for (int j=0; j<5; j++)
    {
        if (i == 2 && j == 2)
            goto zarazka;
        cout << "(" << i << ", " << j << ") ";
    }
    cout << endl;
}
cout << "neprovede se";

zarazka:
    cout << endl;
}
```

# Znaky v C++

- typy: **char** (1B), **unsigned char** (1B, 0...255), **signed char** (1B, -128...127),  
**wchar\_t** (2 nebo 4 B dle implementace), **char16\_t**, **char32\_t**
- jedná se o speciální celočíselné datové typy

## Typ char

- hodnoty 0, 1, ...255
- aritmetické operátory +, -, \*, /, %
- relační operátory: <, <=, >=, ==, !=
- rozdíl je při použití s proudy cin, cout : nezobrazují číselnou hodnotu, ale znak na odpovídající pozici v ASCII tabulce

## Typ wchar\_t

- wide char, kódování Unicode



# Znaky v C++

## Znakové konstanty

```
char a = 'a';  
char b = 97; // ASCII kod  
char c = '\x61'; // ASCII kod hexadecimalne
```

## Operátory

```
char znak = 'a';  
...  
znak +=8;  
if (znak > 'b')  
    cout << "Je vetsi";  
if ('A' > 'a')  
    ...  
...
```

# Znaky v C++

## Řídící posloupnosti

```
char a = '\n';    // nový řádek
a = '\t';        // tabulátor
a = '\\';        // apostrof
a = '\"';         // uvozovka
a = '\\\';       // obrácené lomítko
...
```

## Načítání znaku z konzole

```
char a;
cin >> a;        // vynechá bílé znaky
a = getchar();   // nevynechá bílé znaky
putchar(a);      // vypisí znak
...
```

# Znaky v C++

## Příklady

- Napište a zavolejte funkci **void vypis\_o\_znaku(char c)**, která vypíše na konzoli informaci, zda je zadaný znak malé písmeno, velké písmeno, číslice nebo jiný typ znaků. Funkce vypíše také ASCII kód daného znaku.
- Napište a zavolejte funkci **char na\_velke(char c)**, která převede malé písmeno na velké. Pokud znak není malé písmeno, vrátí ho beze změny.
- Napište a zavolejte funkci **char na\_male(char c)**, která převede velké písmeno na male. Pokud znak není velké písmeno, vrátí ho beze změny.
- Napište funkci **void cely\_radek()**, která postupně načte z konzole celý řádek (až po znak '\n') a vypíše ho na konzoli s tím, že všechna malá písmenka změní na velká.
- Napište funkci **sachovnice()**, která vytiskne na konzoli sachovnici (bílé políčko bude dvakrát znak 219, černé políčko bude dvakrát znak 176)

# Znaky v C++

## Příklad 1

```
void vypis_o_znaku(char znak)
{
    if (znak >= 'a' && znak <= 'z')
        cout << "Zadal_jsi_male_pismeno." << endl;
    else if (znak >= 'A' && znak <= 'Z')
        cout << "Zadal_jsi_velke_pismeno." << endl;
    else if (znak >= '0' && znak <= '9')
        cout << "Zadal_jsi_cislici." << endl;
    else
        cout << "Zadal_jsi_jiny_typ_znaku." << endl;
    cout << "ASCII_kod_znaku_je" << (int)znak << endl;
}
```

# Znaky v C++

## Příklad 2 a 3

```
char na_velke(char znak)
{
    if (znak >= 'a' && znak <= 'z')
        znak -= ( 'a' - 'A' );
    return znak;
}
```

```
char na_male(char znak)
{
    if (znak >= 'A' && znak <= 'Z')
        znak += ( 'a' - 'A' );
    return znak;
}
```

# Znaky v C++

## Příklad 4

```
void cely_radek ()  
{  
    cout << "Zadej_radek_textu:" <<endl;  
    char znak;  
    do  
    {  
        znak=getchar();  
        cout << na_velke(znak);  
    } while (znak != '\n')  
}
```

# Znaky v C++

## Příklad 5

```
void sachovnice()  
{  
    char bilePolicko = 219;  
    char cernePolicko = 176;  
    cout << endl;  
    for(int i = 0; i < 8; i++)  
    {  
        for(int j = 0; j < 8; j++)  
        {  
            if(j % 2 == i % 2)  
                cout << bilePolicko << bilePolicko;  
            else  
                cout << cernePolicko << cernePolicko;  
        }  
        cout << endl;  
    }  
    cout << endl;  
}
```

## Funkce ze standardní knihovny pro práci se znaky

### **funkce pro klasifikaci znaků (knihovna ctype)**

isalnum()	písmeno nebo číslice
isalpha()	písmeno
isdigit()	čísllice
isspace()	bílý znak
isupper()	velké písmeno
islower()	malé písmeno
isalnum()	písmeno nebo číslice
isprint()	tisknutelný znak
ispunct()	tisknutelný znak, co není mezera, písmeno, číslice

...

### **převod znaků na velká nebo na malá písmena**

tolower()	na velká písmena
toupper()	na malá písmena



# Řízení běhu programu - příkaz SWITCH

**Příklad 1:** "KALKULAČKA" objemů a obsahů těles (pokračování pcv2/ctverec1.cpp)

Kalkulacka pro vypocet objemu a obsahu

Pro vypocet obsahu ctverce stiskni a.

Pro vypocet obsahu kruhu stiskni b.

Pro vypocet objemu valce stiskni c.

Pro vypocet objemu koule stiskni d.

Pokud chces ukoncit vypocet , stiskni k.

## Řízení běhu programu - příkaz SWITCH

**Příklad 1:** "KALKULAČKA" objemů a obsahů těles (pokračování pcv2/ctverec1.cpp)

Kalkulacka pro vypocet objemu a obsahu

Pro vypocet obsahu ctverce stiskni a.

Pro vypocet obsahu kruhu stiskni b.

Pro vypocet objemu valce stiskni c.

Pro vypocet objemu koule stiskni d.

Pokud chces ukoncit vypocet, stiskni k.

D

Zadej polomer koule: 1.5

Objem koule je 14.1372

Pro vypocet obsahu ctverce stiskni a.

Pro vypocet obsahu kruhu stiskni b.

Pro vypocet objemu valce stiskni c.

Pro vypocet objemu koule stiskni d.

Pokud chces ukoncit vypocet, stiskni k.

k

Koncim .

# Řízení běhu programu - příkaz SWITCH

```
switch (vyraz)      // celociselny datovy typ (ne float)
{
    case konstantni_vyraz_1 :           // navesti
        ...
    case konstantni_vyraz_2 :
        ...
    case konstantni_vyraz_n :
        ...
    default :
        ...
}
```

# Vyhodnocení příkazu SWITCH

- 1 vyhodnotí se výraz
  - 2 pokud se hodnota výrazu rovná některému z návěstí `konstantni_vyraz_1, ... konstantni_vyraz_n` → program pokračuje za daným návěstím
  - 3 jinak pokud je uvedeno návěstí **default**, program pokračuje za ním
  - 4 jinak **switch** zkončí (a nic nevykoná)
- není možné větvit podle intervalů nebo podle racionálních čísel
  - návěstí **default**: je nepovinné

# Řízení běhu programu - příkaz SWITCH

## Příklad

```
void vypis (int a)
{
    switch (a)
    {
        case 1 :
            cout << "jedna_";
        case 2 :
            cout << "dva_";
        case 3 :
            cout << "tri_";
    }
    cout << endl;
}

...
int main()
{
    vypis(1);
    vypis(4);
```

# Řízení běhu programu - příkaz SWITCH

## Příklad

- pro ukončení provádění příkazů je třeba používat příkaz **break**

```
void vypis (int a)
{
    switch (a)
    {
        case 1 :
            cout << "jedna _";
            break;
        case 2 :
            cout << "dva _";
            break;
        case 3 :
            cout << "tri _";
            break;
    }
}
```

# Řízení běhu programu - příkaz SWITCH

## Příklad

```
void vypis (int a)
{
    switch (a)
    {
        case 1 :
            cout << "jedna_";
            break;
        case 2 :
            cout << "dva_";
            break;
        case 3 :
        case 5 :
            cout << "tri_nebo_pet_";
            break;
        default :
            cout << "jine_cislo_";
    }
}
```

## Řízení běhu programu - příkaz SWITCH

**Příklad 1:** "KALKULAČKA" objemů a obsahů těles (pokračování pcv2/ctverec1.cpp)

Kalkulacka pro vypocet objemu a obsahu

Pro vypocet obsahu ctverce stiskni a.

Pro vypocet obsahu kruhu stiskni b.

Pro vypocet objemu valce stiskni c.

Pro vypocet objemu koule stiskni d.

Pokud chces ukoncit vypocet, stiskni k.

D

Zadej polomer koule: 1.5

Objem koule je 14.1372

Pro vypocet obsahu ctverce stiskni a.

Pro vypocet obsahu kruhu stiskni b.

Pro vypocet objemu valce stiskni c.

Pro vypocet objemu koule stiskni d.

Pokud chces ukoncit vypocet, stiskni k.

k

Koncim .



# Řízení běhu programu - příkaz SWITCH

**Příklad 1** testík z matematiky pro malé školáky

**int testik(int pocet)**

- Funkce náhodně generuje příklady na základní operace +, -, \*, / nad dvěma "malými" kladnými celými čísli (např.  $10 + 5 =$ ,  $56/7 =$ )
- Funkce porovná správné a zadané řešení.
- Funkce vrátí výslednou známku (každá chyba známku zhoršuje).
- Pocet je počet vygenerovaných příkladů

Test z matematiky

$13+4 = 17$

Spravne.

$221/13 = 12$

Nespravne. Spravny vysledek je 17.

$10*4 = 40$

Spravne.

Tvoje znamka je: 2

# Řízení běhu programu - příkaz SWITCH

## Generování pseudonáhodných čísel

```
#include <ctime> // time
#include <cstdlib> // srand, rand
...
int main()
{
    srand(time(NULL)); // pocatecni nastaveni generatoru
                       // (volame pouze jednou)
    ...
    int cislo = rand(); // nahodne cele cislo 0 ... RAND_MAX
    cislo = rand() % 10; // nahodne cele cislo 0 ... 9
}
```