

Základy programování v C++ - 4. cvičení Cykly (for), skoky (break, continue, goto)

Zuzana Petříčková

3. října 2019

Už jsme probírali

- čtení textu ze systémové konzole a jeho výpis na konzoli
 - cin, cout,...
- proměnné a datové typy
 - aritmetické a boolovské výrazy
 - implicitní a explicitní přetypování
- Příkazy pro řízení běhu programu
 - podmíněné bloky (podmínky) IF
 - cykly WHILE, DO-WHILE
- Dobrovolný domácí úkol a jeho řešení

Řízení běhu programu

- podmíněné bloky (podmínky)
- cykly (smyčky)
- podprogramy (funkce)

Řízení běhu programu - Podmínky IF

Jeden příkaz:

```
if (testovací_podminka)
    prikaz1;
else
    prikaz2;
```

Více příkazů:

```
if (testovací_podminka)
{
    prikaz1;
    prikaz2;
    ...
}
```

testovací_podminka = výraz, který lze převést na logickou hodnotu

Řízení běhu programu - Cykly

Cykly:

- část kódu, která se provádí vícekrát
- o dalším opakování rozhoduje splnění podmínky
- počet opakování může a nemusí být znám předem

Řízení běhu programu - Cyklus WHILE

Jeden příkaz:

```
while (testovací_podminka)
    prikaz1;
```

Více příkazů:

```
while (testovací_podminka)
{
    prikaz1;
    prikaz2;
    ...
}
```

testovací_podminka = výraz, který lze převést na logickou hodnotu

Řízení běhu programu - Cyklus DO-WHILE

```
do
{
    prikaz1;
    prikaz2;
    ...
}
while (testovaci_podminka)
```

Řízení běhu programu - Cyklus FOR

Jeden příkaz:

```
for ( pocatecni_prikaz; podminka; krok )  
    prikaz1;
```

Více příkazů:

```
for ( pocatecni_prikaz; podminka; krok )  
{  
    prikaz1;  
    prikaz2;  
    ...  
}
```


Řízení běhu programu - Cyklus FOR

FOR-cyklus:

```
for ( pocatecni_prikaz; podminka; krok )  
{  
    prikaz1;  
    prikaz2;  
    ...  
}
```

Ekvivalentní kód:

```
pocatecni_prikaz;  
while ( podminka )  
{  
    prikaz1;  
    prikaz2;  
    ...  
    krok;  
}
```

Řízení běhu programu - Cyklus FOR

Příklad:

- typické použití: provedení pevného počtu opakování:

```
for (i = 1; i <= 10; i++)  
    cout << i << " " << i*i << endl;
```

```
for (i = 10; i > 0; i--)  
    cout << i << " " << i*i << endl;
```

```
for (i = 1; i <= 100; i*=2)  
    cout << i << endl;
```

```
for (i = 1; i <= 10; i++)  
{  
    for (j = 1; j <= 10; j++)  
        cout << i*j << " ";  
    cout << endl;  
}
```

Řízení běhu programu - Cyklus FOR

Složitější příklad:

- **faktoriál:** povídací program, který pro zadané celé číslo n spočte a vypíše hodnotu $n!$.

Faktoriál: řešení

```
pocatecni_prikaz;  
while (podminka)  
{  
    prikaz1;  
    krok;  
}
```

```
for (pocatecni_prikaz; podminka; krok)  
    prikaz1;
```

```
int n;  
... // nacteni n  
long long f = 1;  
int i = 1;  
while (i <= n)  
{  
    f *= i;  
    i++;  
}  
cout << f << endl;
```

```
int n;  
... // nacteni n  
long long f = 1;  
for (int i = 1; i <= n; i++)  
{  
    f *= i;  
}  
cout << f << endl;
```

Řízení běhu programu - skoky

break:

- Ukončí cyklus, program následuje prvním příkazem za cyklem

continue:

- Ukončí aktuální iteraci cyklu a započne novou

goto:

- Přesun řízení na zadané návěští

Příklad

```
int a;  
for (int i=0; i<10; i++)  
{  
    cin >> a;  
    if (a < 0)  
        continue;  
    if (a == 1)  
        break;  
    cout << a;  
}
```

Řízení běhu programu - skoky

Příklad: Program načítá čísla na vstupu, dokud uživatel nezadá celé číslo v daném rozsahu. Program používá nekonečný cyklus.

```
int x, min = 0, max = 100;;
while (true)
{
    cout << "Zadej cislo z intervalu <" <<
        min << ", " << max << ">" << endl;
    cin >> x;

    if (x >= min && x <= max)
        break;
    cout << "Neplatne zadani, zkus to znovu." << endl;
}
cout << "Trefa! " << endl;
```

Řízení běhu programu - skoky

Příklad - pokračování: program si poradí se situací, kdy se načtení čísla nepodařilo - snaží se situaci "opravit".

```
int x, min = 0, max = 100;;
while (true)
{
    cout << "Zadej cislo z intervalu <" <<
        min << "," << max << ">" << endl;
    cin >> x;

    if (!cin) // kontrola vstupu
    {
        cin.clear();
        cin.ignore(256, '\n');
        cout << "Zadany vstup neni cislo , zkus to znovu." << endl;
        continue;
    }
    if (x >= min && x <= max)
        break;
    cout << "Neplatne zadani , zkus to znovu." << endl;
}
cout << "Trefa! " << endl;
```

Řízení běhu programu - skoky

Příklad - pokračování: program si poradí se situací, kdy se načtení čísla nepodařilo - ukončí se.

```
int x, min = 0, max = 100;;
while (true)
{
    cout << "Zadej cislo z intervalu <" <<
        min << "," << max << ">" << endl;
    cin >> x;

    if (!cin) // kontrola vstupu
    {
        cout << "Zadany vstup neni cislo. Konec." << endl;
        return 1;
    }
    if (x >= min && x <= max)
        break;
    cout << "Neplatne zadani, zkus to znovu." << endl;
}
cout << "Trefa! " << endl;
```


Shrnutí:

Jak reagovat na chybu (např. ve vstupu od uživatele)?

- 1 Vypsati informaci o chybě.
- 2 Reagovat na chybu:
 - snažit se chybu opravit.
"Zadej znova..."
"x = min;"
 - Ukončit program
 - Chybu ignorovat → zákeřné

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

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;  
}
```

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 << "tento prikaz se neprovede";

zarazka:
    cout << endl;
```

Řízení běhu programu - cyklus FOR

Příklad na kontrolu vstupu:

- 1 **součet:** Napište povídací program, který sčítá čísla ze vstupu, dokud uživatel nezadá 0. Vypíše výsledek. Pokud uživatel nezadá číslo (udělá překlep ap.), požádá ho program o opravu zadání.

Příklady na procvičení FOR:

- 1 Napište for-cyklus, který vypíše všechna sudá čísla od 10 do 25.
- 2 **mocnina:** Napište povídací program, který pro zadané x a n (double x , int n) spočte a vypíše x^n .
- 3 **dělitelé:** Napište povídací program, který spočítá a vypíše všechny dělitele zadaného čísla n .

Další příklady na procvičení

- 1 **trojúhelník (pyramida):** Napište povídací program, který načte z klávesnice výšku (rovnoramenného) trojúhelníka a následně nakreslí trojúhelník (pyramidu) na obrazovku.
- 2 **kruh:** Napište povídací program, který načte z klávesnice poloměr kruhu a následně nakreslí kruh na obrazovku.
- 3 **prvočíslo:** Napište povídací program, který otestuje, zda je zadané číslo prvočíslem.
- 4 **fibonacci:** Napište povídací program, který otestuje, zda je zadané číslo prvkem Fibonacciho posloupnosti.

Další příklady na procvičení

trojúhelník (pyramida):

```
zadej vysku trojuhelnika: 8
```

```
      *
     * * *
    * * * * *
   * * * * * * *
  * * * * * * * *
 * * * * * * * * *
* * * * * * * * * *
```