

Cykly for, foreach, while a do/while

Obsah

Co je to cyklus?	2
For.....	3
Definice.....	3
Příklady	4
Využití	6
Foreach	7
Definice.....	7
Příklady	8
Využití	9
While.....	10
Definice.....	10
Příklady	11
Využití	12
Do/While.....	14
Definice.....	14
Příklady	14
Využití	15

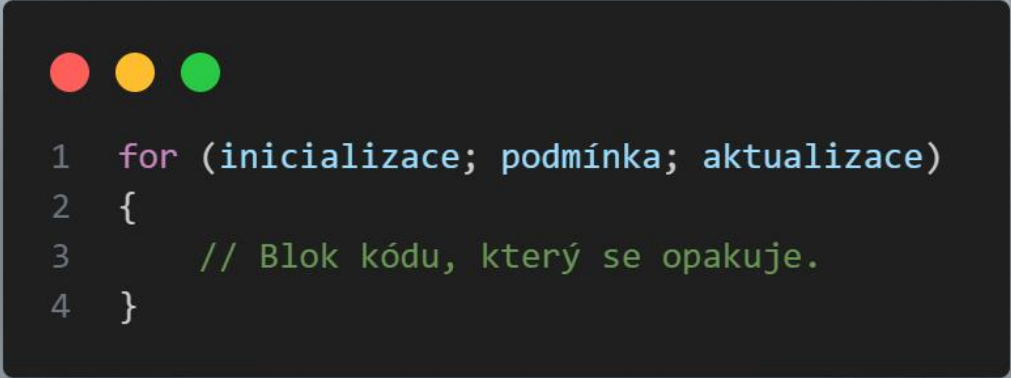
Co je to cyklus?

Cyklus, taktéž smyčka, je nepostradatelnou součástí téměř jakéhokoliv programu. Cykly mohou provádět blok kódu, pokud je dosaženo zadané podmínky. Také jsou praktické, protože šetří čas, snižují chyby a činí kód čitelnějším. Nejvíce jsou užitečné pro opakování úkonů nebo pro zpracování kolekce prvků v poli či programu.

For

Definice

Cyklus for slouží k opakování určitého bloku kódu s

A code editor window with a dark background and three colored window control buttons (red, yellow, green) in the top left corner. It contains a C-style for loop syntax example with line numbers 1 through 4.

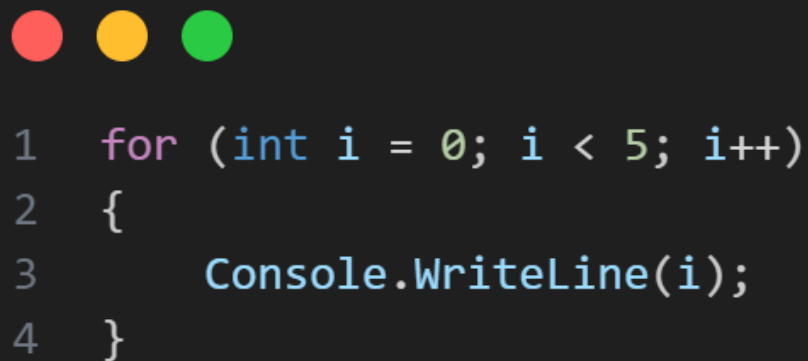
```
1  for (inicializace; podmínka; aktualizace)
2  {
3      // Blok kódu, který se opakuje.
4  }
```

předem známým počtem iterací. Jeho syntaxe je následující:

- Inicializace
 - Část, která se provede jednou na začátku cyklu
 - Slouží k inicializaci proměnných nebo nastavení počátečních hodnot
- Podmínka
 - Logická podmínka, která je ověřována před každým průchodem cyklem
 - Pokud je podmínka pravdivá, cyklus pokračuje; pokud je nepravdivá, cyklus se ukončí
- Aktualizace

- Část, která se provádí po každém průchodu cyklem
- Slouží k aktualizaci proměnných nebo ke změně stavu

Příklady



```
1  for (int i = 0; i < 5; i++)  
2  {  
3      Console.WriteLine(i);  
4  }
```

Výstup: 0, 1, 2, 3 a 4



```
1  for (int i = 5; i > 0; i--)  
2  {  
3      Console.WriteLine(i);  
4  }
```

Výstup: 5, 4, 3, 2 a 1



```
1  double cislo = 0.574;  
2  float plus = 9.88f;  
3  
4  for (double i = cislo; i <= 101; i += plus)  
5  {  
6      Console.WriteLine(i);  
7  }
```

Zaokrouhlený výstup: 0,5; 10,5; 20,3; 30,2; 40,1; 50;
59,9; 69,7; 79,6; 89,5 a 99,4

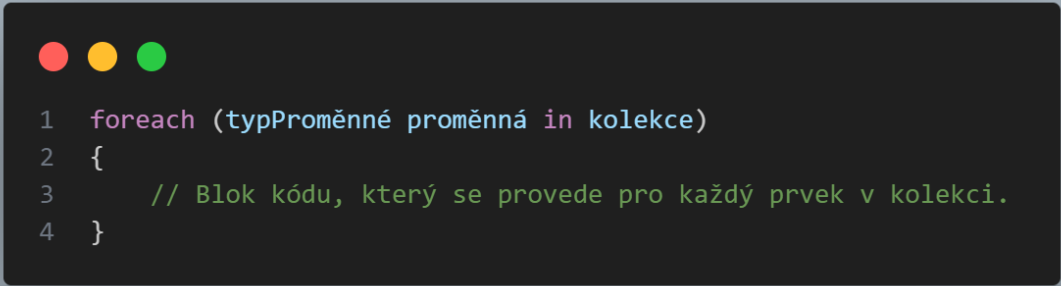
Využití

- Procházení pole, listu nebo kolekce
- Opakování operací
- Výpočty s pevným počtem iterací

Foreach

Definice

Cyklus foreach se používá k procházení prvků v kolekci nebo poli, aniž by bylo nutné pracovat s indexy. Jeho syntaxe je následující:



```
1  foreach (typProměnné proměnná in kolekce)
2  {
3      // Blok kódu, který se provede pro každý prvek v kolekci.
4  }
```

- typProměnné
 - Datový typ proměnné, který bude přijímat hodnoty prvků kolekce.
- Proměnná
 - Proměnná, do které budou postupně uloženy jednotlivé prvky kolekce.
- Kolekce
 - Kolekce (pole, seznam, kolekce) obsahující prvky k procházení.

Příklady



```
1  string slovo = "Kulička";  
2  
3  foreach(char c in slovo)  
4  {  
5      Console.Write(c + " ");  
6  }
```

Výstup:

K u l i č k a



```
1  string[] jmena = { "Adam", "Marek", "Eva", "Dana" };  
2  
3  foreach(string s in jmena)  
4  {  
5      Console.WriteLine(s);  
6  }
```




Adam
Marek
Eva
Dana

Výstup:

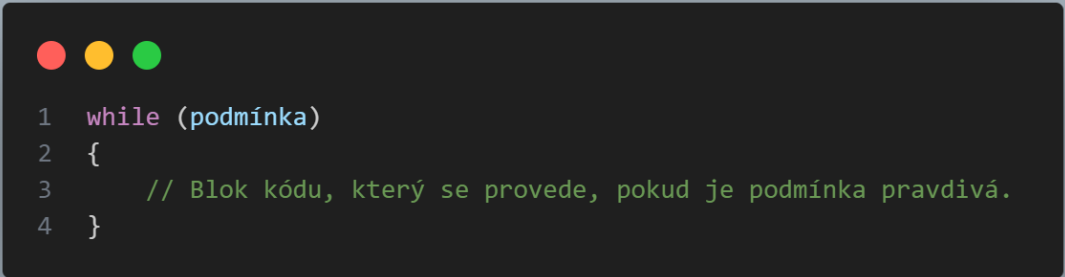
Využití

- Používá se pro procházení prvků v seznamu, kolekci nebo jiných typů kolekcí

While

Definice

Cyklus while v jazyce C# se používá k opakování bloku kódu, dokud je splněná určitá podmínka. Jeho syntaxe je následující:

A screenshot of a code editor window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. The code is written in C# and shows a while loop. The text is as follows:

```
1 while (podmínka)
2 {
3     // Blok kódu, který se provede, pokud je podmínka pravdivá.
4 }
```

- Podmínka
 - Logická podmínka, která je vyhodnocována před každým provedením bloku kódu uvnitř cyklu. Pokud je podmínka pravdivá, blok kódu se provede; pokud je nepravdivá, cyklus se ukončí.

Příklady



```
1  int cislo = 0;
2
3  while (cislo < 5)
4  {
5      Console.WriteLine(cislo);
6      cislo++;
7  }
```

0
1
2
3
4

Výstup:



```
1  bool podminka = true;
2  double hodnota = 2.5;
3
4  while (podminka)
5  {
6      hodnota += hodnota;
7      if (hodnota > 25)
8      {
9          podminka = false;
10     }
11
12     Console.WriteLine(hodnota);
13 }
```

5

10

20

40

Výstup:

Využití

- Iterace s neznámým počtem opakování:


- Používá se, když nemáme předem známý počet iterací nebo když chceme opakovat blok kódu, dokud je splněna určitá podmínka.
- Čekání na událost:
 - while cyklus se může použít k čekání na splnění určité podmínky nebo události, například na vstup od uživatele.
- Práce s uživatelským vstupem:
 - Často se používá k opakování otázek nebo úkolů, dokud uživatel nezadá správný vstup.

Do/While

Definice

Principově stejný jako cyklus While. K opakování dochází v bloku Do.

Příklady



```
1  int cislo = 5;
2
3  do
4  {
5      Console.WriteLine(cislo);
6      cislo--;
7  }
8  while (cislo > 0);
```

Výstup:

```
5
4
3
2
1
```

Využití

- Stejně jako u cyklu While