

Cykly

Cyklus/smyčka je způsob, kterým lze program donutit provést danou část kódu několikrát. Namísto kopírování a vkládání částí kódu několikrát za sebe, lze jednoduše zapsat daný kód do cyklu, který se bude opakovat dokud je splněna určitá podmínka. Výhoda je zřejmá - pokud je potřeba měnit kód uvnitř smyčky, je potřeba ho změnit pouze jednou. Cykly se také mohou opakovat teoreticky do nekonečna.

Klíčová slova

- Cykly
- for cyklus
- while cyklus
- nekonečný cyklus
- do while cyklus

For cyklus

```

1 for (int i = 0; i < 10; i
    ++)
2 {
3 printf("hello!\n");
4 }
5 for (int j = 0; j < 10; j
    ++)
6 {
7 printf("%d\n", j);
8 }
```

Prvním typem cyklu v C je cyklus **for**. Definice **for** cyklu vyžaduje tři části oddělené středníkem uvnitř kulatých závorek, jak je vidět na řádce 1 vlevo. První část je inicializace - vytvoření proměnné a přiřazení hodnoty **0**. Druhou částí je nastavení podmínky cyklu - dokud bude **i** menší než 10, bude se cyklus opakovat. Třetí část je krok s jakým se bude proměnná **i** měnit. V tomto případě se v každém cyklu zvýší hodnota **i** o 1.

Dokud je **i** menší než 10 bude cyklus pokračovat. Výsledkem je výpis **hello!\n** desetkrát do terminálu.

Změnu hodnoty proměnné **j** lze s výhodou využít i uvnitř cyklu. Příklad je zobrazení hodnoty při každém běhu smyčky (řádky 5 až 8). Na začátku je **j** rovno 0 a 0 se zobrazí v terminálu. Postupně se inkrementací **j** v každém cyklu vytisknou čísla 0 až 9.

While cyklus

C obsahuje syntaxi pro cyklus **while**. Tento cyklus kontroluje vstupní podmínku v kulatých závorkách a pokud je splněna, provede se kód vložený mezi složené závorky. Po ukončení smyčky se podmínka zkontroluje znovu a pokud je stále splněna, provede se smyčka znovu. Cyklus se tedy opakuje dokud podmínky platí (příklad je na řádcích 1 až 6 vpravo).

```

1 int k = 0;
2 while (k < 10)
3 {
4 printf("%d\n", k);
5 k++;
6 }
7 while (true)
8 {
9 printf("hello!\n");
10 }
```

Zvláštním případem může být **while**, kde je podmínka cyklu vždy splněna (**true**) - taková smyčka nikdy neskončí a bude se tedy provádět do nekonečna. Příklad je na řádcích 7 až 10 - podmínka bude vždy splněna. Cyklus **while** je užitečný zejména v případech, kdy nevíme, jak dlouho má smyčka probíhat.

Do-While cyklus

```

1 int j;
2 do
3 {
4 printf("Positive Number: \n");
5 scanf("%d", &j);
6 }
7 while (j <= 0);
```

Cyklus **do-while** je podobný cyklu **while** - probíhá dokud je splněna podmínka. Rozdíl oproti cyklu **while** je ale v tom, že cyklus **do-while** vždy proběhne alespoň jednou, nehledě na splnění podmínky cyklu. Ověření samotné podmínky totiž probíhá až po průchodu cyklem. Takový postup může být někdy výhodný - například pokud je vyžadován uživatelský vstup. Program se jednou zeptá na vstup a pak rozhodne, zda bude pokračovat v cyklech a přijímat další vstupy.

Příklad vlevo vyzve uživatele k zadání celého čísla a bude se opakovat dokud uživatel nezadá celé kladné číslo.