

Řídící struktury, cykly (větvení programu, logické operátory, spojování podmínek, druhy cyklů a jejich využití)

Řídící struktury

Řídící struktury jsou základní prvky v algoritmizaci. Jde o posloupnost příkazů, větvení kódu a cyklení kódu. Posloupností příkazů se rozumí seznam příkazů, které jsou vykonávány jeden za druhým – nejzákladnější prvek algoritmu – postup kódem – žádný krok nesmí být vynechán, ani nesmí být zaměněno pořadí kroků.

Větvení programu

Větvení nastane v momentě, kdy se program liší podle splnění nebo nesplnění nějaké podmínky. Větvení se skládá ze tří částí:

1. otázka, na kterou lze odpovědět ano či ne
2. první možnost pokračování programu (odpověď zní ano)
3. druhá možnost pokračování programu (odpověď zní ne)

Vícenásobné větvení je pak případ, kdy je nutné mít více možností pro rozhodování – větev je pak vybrána podle **hodnoty** podmínky (provede se ta větev, která hodnotě podmínky odpovídá).

Logické operátory

Logické operátory jsou součástí zápisu podmínek a (v jazyce Python) jsou celkem 3. Umožňují rozšířit podmínku svým způsobem rozšířit o další. Zjednodušeně řečeno musí být splněno několik podmínek, aby mohl program pokračovat danou větví. Níže jsou uvedeny všechny tři logické operátory současně s příkladem výrazu podmínky a slovním popisem, co musí být splněno, aby byl výraz pravdivý a podmínka tedy byla splněna.

1. **and** (a zároveň) – $x > 0$ and $x < 10$ – x musí být větší než 0 **a zároveň** menší než 10
2. **or** (nebo) – $x < 0$ or $x > 10$ – x musí být menší než 0 **nebo** větší než 10
3. **not** (není) – $\text{not}(x > y)$ – výraz $x > y$ musí být **neppravdivý**, tedy x musí být menší než y

Spojování podmínek

Větve podmínek se po svém vykonání mohou spojovat a program tak bude v určité části kódu postupovat dále stejně, ať už dříve byla nějaká podmínka splněna nebo splněna nebyla.

```
if a > b:
    print("a je vetsi nez b")
else:
    print("a je mensi nez b nebo je a rovno b")

print("zde se vetve predchozi podminky za vsech okolnosti opet setkavaji")
```

Druhy cyklů a jejich využití

Cyklus je založený na podmínce. Pokud je podmínka splněna, vykonávají se příkazy v těle cyklu. Pokud podmínka splněna není, cyklus se ukončí a program pokračuje dále. Rozlišují se tři základní druhy cyklů.

1. **for** – předem musí být znám počet opakování cyklu; používá se např. pro průchod poli či seznamy, kde jsou jednotlivé prvky určeny celočíselným indexem
2. **while** – předem není znám počet opakování, podmínka je na začátku cyklu; používá se například v momentě, kdy chceme vykonávat nějakou činnost do momentu, než se splní určitá podmínka (např. než je zadán vstup od uživatele celé číslo)
3. **do-while** – předem není znám počet opakování; podmínka je na konci cyklu; cyklus je tedy vždy vykonán alespoň jednou; používá se tedy tam, kde je nutný alespoň jeden průchod cyklem

K do-while cyklu: následující kód bude vykonáván tak dlouho, než bude splněna podmínka, že vygenerované náhodné číslo není dělitelné sedmi.

```
<?php
do {
    $random = mt_rand(5, 30);
} while (!($random % 7));
?>
```