

Objektovo orientované programovanie

Učiteľ:
Ing. Jozef Wagner PhD.

Učebnica:
<https://oop.wagjo.com/>

OPG

Cvičenie 3

1. Vetvenie
2. Cykly
3. Prerušenie toku

Vetvenie

switch je podmienka pre výber z viacerých možností

- testuje **primitívnu hodnotu**, **String** alebo **enum**
- ak nič nevyhovuje, vyberie **default**
- ak neuvediem **break**, nastáva **fall-through** a vykoná sa aj nasledujúca možnosť

Vetvenie

```
switch (den) {  
    case 1:  
        System.out.println("Pondelok");  
        break;  
    case 2:  
        System.out.println("Utorok");  
        break;  
    case 3:  
        System.out.println("Streda");  
        break;  
    default:  
        System.out.println("Neznámy deň");  
}
```

Vetvenie

Moderná verzia **switch**-u

- je výraz
- nepoužíva fall-through
- jeden prípad môže mať viacero možností
- spoznám ju tak, že má v sebe šípky **->**

Vetvenie

```
String nazov = switch (den) {  
    case 1 -> "Pondelok";  
    case 2 -> "Utorok";  
    case 3 -> "Streda";  
    case 6, 7 -> "Víkend";  
    default -> "Neznámy deň";  
};
```

Cyklus

V Jave máme 4 typy cyklov: `for`, `while`, `do-while` a `for-each`

Riadenie toku cyklov

- `break` ukončí predčasne cyklus
- `continue` preskočí aktuálnu iteráciu a pokračuje ďalšou

Cyklus do-while

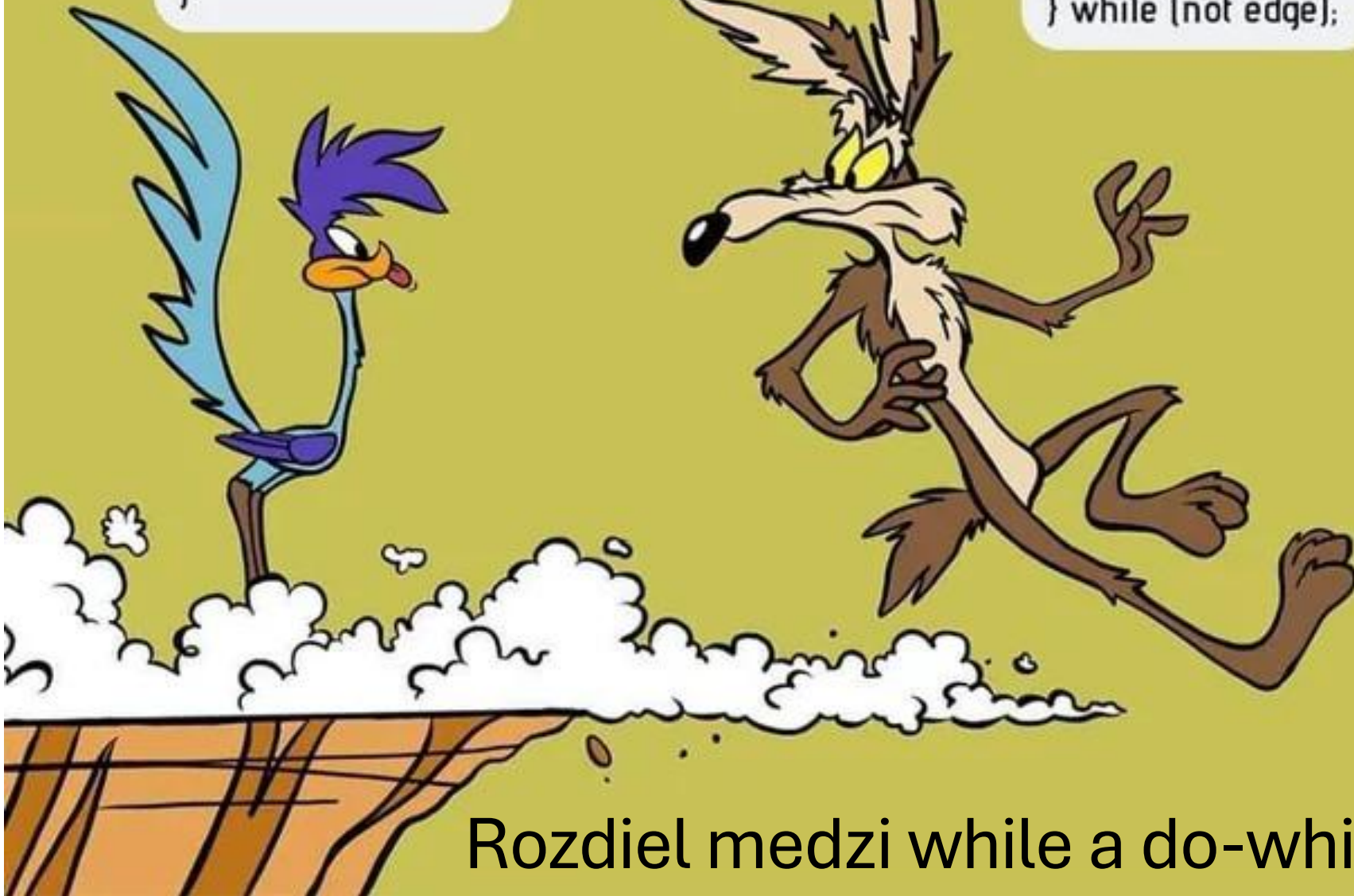
Cyklus **do-while** je podobný cyklu **while**, ale vždy sa vykoná aspoň jedenkrát

```
do {  
    hod = random.nextInt(6) + 1;  
    System.out.println("Padlo číslo: " + hod);  
} while (hod != 6);
```



```
while (not edge) {  
    run();  
}
```

```
do {  
    run();  
} while (not edge);
```



Rozdiel medzi while a do-while

Cyklus for-each

Iteruje naprieč kolekciou

Používa sa na čítanie, nie na modifikáciu

Neposkytuje pozíciu prvku v cykle, nemá index

```
int[] ciska = {1, 2, 3, 4, 5};
```

```
for (int cislo : ciska) {  
    System.out.println("Číslo: " + cislo);  
}
```