

Jazyk Java 1

Seminář 10

Daniel Bazala



Katedra informatiky
Univerzita Palackého v Olomouci

Vnořené a anonymní třídy

Vnořené třídy

- vnořená třída je třída uvnitř jiné třídy
- používá se v konkrétních případech, kdy je třída použita pouze na jediném místě

```
class Outer {  
    String name = "Outer";  
  
    class Inner {  
        String name = Outer.this.name + ".Inner";  
    }  
  
    Inner inner = new Inner();  
}  
  
Outer outer = new Outer();  
System.out.println(outer.inner.name); // Outer.Inner
```

Anonymní třídy

- umožňují vytvořit instanci třídy přímo při její deklaraci
- podobné jako lokální třídy, ale nemají jméno
- lambda výrazy jako syntaktický cukr pro vytváření anonymních tříd

```
Runnable r1 = new Runnable() { // Anonymní třída
    @Override
    public void run() {
        System.out.println("Anonymous class");
    }
};
```

```
Runnable r2 = () -> { // Lambda výraz
    System.out.println("Lambda Expression");
};
```

```
new Thread(r1).start();
new Thread(r2).start();
```

Anonymní třídy

- využití např. jako komparátory

```
List<Integer> numbers = new ArrayList<>(Arrays.asList(1, 2, 3, 4),
numbers.sort(new Comparator<Integer>() {
    public int compare(Integer i, Integer j) {
        if (i < j)
            return 1;
        return -1;
    }
});
System.out.println(numbers);
```

Vlákna

Vlákna v Javě

- doposud program vykonáván sekvenčně pouze v jediném vlákně
- moderní operační systémy podporují souběh více procesů
- vlákno představuje odlehčený proces
- může přistupovat k jiným vláknům v rámci stejného procesu
- přidáním více vláken můžeme docílit paralelního chování

Vlákna v Javě

- nové vlákno můžeme vytvořit prostřednictvím třídy Thread
- třída Thread implementuje funkční rozhraní Runnable

```
Thread t = new Thread(() -> {System.out.println("Hello");});  
t.start();
```

- můžeme také přímo rozšířit třídu Thread

```
class Worker extends Thread {  
    @Override  
    public void run() {  
        System.out.println("Hello");  
    }  
}
```

```
Worker w = new Worker();  
w.start();
```


Vlákna v Javě

- vlákno můžeme pozastavit pomocí metody `sleep()`
- nebo přerušit vyvoláním vyjímky `throw new InterruptedException();`
- vlákno může čekat na jiné vlákno pomocí metody `join()`

Synchronizace

- při sdílení prostředků mezi vlákny může dojít k nekonzistentním stavům
- potřeba synchronizace mezi vlákny
- klíčové slovo `synchronized` - do synchronizované metody může vejít pouze jedno vlákno v jeden čas, další vlákno do ní získá přístup až ji předchodí vlákno opustí
- klíčové slovo `volatile` - pro vytváření zdrojů nad kterými lze provést atomické operace
- metoda `wait()` - čekání na událost `notify()/notifyAll()`

- Více:
<https://docs.oracle.com/javase/tutorial/essential/concurrency/index.html>

Úkol

Úkol seminář 10

- <http://marcus.webly3d.net/ukol10>