Jazyk Java 1

Seminář 4

Daniel Bazala

Katedra informatiky
Univerzita Palackeho v Olomouci

Úvod do OOP - pokračování

Modifikátory přístupu

- implicitně výchozí viditelnost - přístup ze tříd ve stejném balíčku
- private - přístup pouze uvnitř třídy
- protected - přístup v rámci třídy a prostřednictvím dědičnosti také mimo balíček
- public - veřejný přístup i mimo třídu ze všech balíčků

Princip zapouzdření

- standardní technika v OOP
- atributy objektu označeny jako privátní - přístup pouze uvnitř třídy
- pro přístup k atributům vytvořeny zvláštní veřejné metody - tzv. “gettry“ a “settry“

```
public class Student {
  private int age;

  Student(int age) {
    this.age = age;
  }

  public int getAge() {
    return age;
  }

  public void setAge(int age) {
    this.age = age;
  }
}
```

- zapouzdření zvyšuje bezpečnost, flexibilitu, kontrola validity
- read-only nebo write-only atributy

Dědičnost

- třídy mohou "přejímat" atributy a metody jiné třídy
- používáme v případě, že je třída specializací jiné třídy
- pravidlo is-a

```
public class Animal {
  private int numOfLegs;

  Animal(int numOfLegs) {
    this.numOfLegs = numOfLegs;
  }

  public int getNumOfLegs() {
    return numOfLegs;
  }
}

public class Dog extends Animal {
  Dog() {
    super(4);
  }
}
Dog arnie = new Dog();
System.out.println(arnie.getNumOfLegs()); // 4
```

Polymorfismus

- specifické chování stejné metody u různých podtříd

```
public class Animal {
  public void makeSound() {
  }
}

public class Dog extends Animal {
  @Override
  public void makeSound() {
    System.out.println("haf!");
  }
}

public class Cat extends Animal {
  @Override
  public void makeSound() {
    System.out.println("mňau!");
  }
}
Animal arnie = new Dog();
Animal kattie = new Cat();
arnie.makeSound(); // haf!
kattie.makeSound(); // mňau!
```

Abstraktní třídy

- třída `Animal` z předchozích příkladů nepředstavuje žádné konkrétní zvíře
- implementace metody `makeSound()` nedává smysl, nevíme o jaké zvíře se jedná
- řešením jsou abstraktní třídy a metody

```
public abstract class Animal {
  public abstract void makeSound();
}

public class Dog extends Animal {
  @Override
  public void makeSound() {
    System.out.println("haf!");
  }
}
```

- nelze vytvořit instanci abstraktní třídy
- abstraktní metody nemají tělo
- neabstraktní třída dědící z abstrakní třídy musí implementovat její abstraktní metody

Rozhraní

- rozhraní představuje veřejnou vrstvu objektu, soupis veřejných metod
- v Javě též nástroj vícenásobné dědičnosti

```
public interface FlyingInterface {
  public void fly();
}

public interface SwimmingInterface {
  public void swim();
}

public class Duck extends Animal implements FlyingInterface,
SwimmingInterface {
  public void makeSound() {
    System.out.println("gaga!");
  }
  public void fly() {
    System.out.println("letím!");
  }
  public void swim() {
    System.out.println("plavu!");
  }
}
```

Rozhraní

- rozhraní popisuje metody, které třída musí implementovat
- neříká jak, pouze které
- metody v rozhraní jsou abstraktní, nemají tělo
- jedna třída může implementovat více rozhraní
- může obsahovat konstantní atributy - klíčové slovo `final`
- objekt může být zastoupen svým rozhraním:

```
SwimmingInterface duckie = new Duck();
duckie.swim() // plavu!
```

Operátor instanceof

- operátor zjišťující, zda je objekt instancí určité třídy
- funguje i v rámci stromu dědičnosti

```
public class Animal {
}

public class Dog extends Animal {
}

public class Cat extends Animal {
}

Dog arnie = new Dog();
System.out.println(arnie instanceof Dog); // true
System.out.println(arnie instanceof Animal); // true
System.out.println(arnie instanceof Cat); // false
```

- možnost přetypování
- pozor na špatné používání, vhodnější je využití dědičnosti, polymorfismu

Záznamy

Záznamy

- od Javy verze 14
- neměnné "třídy", všechny atributy jsou `private final`
- automaticky vygenerované "gettry", které se jmenují stejně jako atributy
- klíčové slovo `record`
- Více: https://docs.oracle.com/en/java/javase/14/language/records.html

Úkol

Úkol seminář 4

- http://marcus.webly3d.net/ukol4