

# Java语言基础

【Day08】

# 面向对象的三大特征

## — 封装

# 封装的概念和流程

- 若代码不做限制，则很多属性值是无效的。封装就是保证属性值有效的技术。
- 封装的步骤：
  - 1 属性(非常量属性)必须private，确保外部的调用者无法直接赋值。
  - 2 提供操作属性的方法，一般都是读/写属性方法，get属性和set属性方法
  - 3 构造方法中，也要调用set属性方法进行赋值。

# static关键字

# static关键字的由来和概念

- 通常情况下，属性/方法都是隶属于对象层级，就是每个对象都有自己独特的属性空间，有些属性需要属于整个类，就是所有对象要共享。
- static就是把对象级提升到类级。static的属性/代码块/方法都是隶属于类，在类加载时就准备完成了，而不需要创建对象(new)。
- 类加载只做一次，包括：
  - 1 类名. 的时候会类加载。
  - 2 new 对象时会类加载。
  - 3 程序员可以用程序加载，比如：Class.forName()。
- 静态的成员(属性和方法)可以用对象. 调用，但一般推荐用类名. 调用。

# static关键字的使用方式

- static的属性/方法在类加载时就已经做好准备，因此类名.就可以调用，与对象存在不存在无关。
- 非static的属性/方法隶属于对象，必须先创建对象，才能使用。
- static的方法中，不能使用非静态的属性或者方法，必须先创建对象，然后用对象.调用；
- static方法的作用在于提供一些“工具方法”和“工厂方法”。
- 而非static的方法中，可以直接使用静态的属性或方法
- 只有隶属于类(所有对象共享)的属性才能加static，static不能随便加。

# 单例设计模式

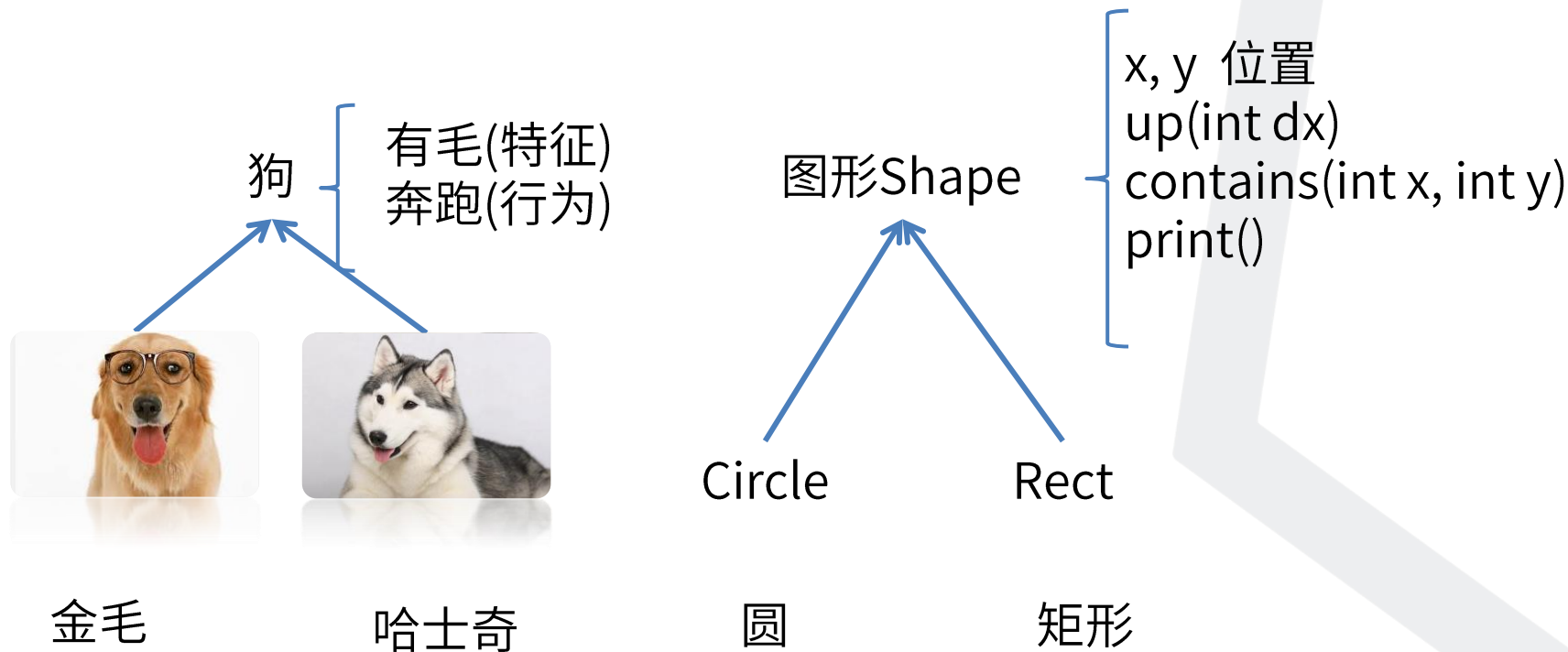
- 设计模式是形成标准化流程的经验总结，是特定问题的固定解决方案。
- 有时候，一个类只需要创建一个对象，不需要第二个，使用单例模式。
- 单例模式的思路：
  - 1 先private构造，阻止外部去创建对象。
  - 2 本类提供一个对象，定义一个private static的本类类型属性。
  - 3 对外提供一个get方法，把唯一的对象传出去。
- 单例有饿汉式和懒汉式两种写法。

# 面向对象的三大特征

## — 继承



# 继承的由来



# 继承的基本概念

- 继承就是子类复用父类的代码，关键字extends表示类和类的继承关系。
- 子类（Sub class）可以继承父类（Super class）的成员变量及成员方法。同时也可以定义自己的成员变量和方法。
- 父类中private的方法，子类继承不过来；但private的属性可以被继承。

```
class Circle extends Shape{  
    int r;  
    public void setR(int r) {  
        this.r = r;  
    }  
}
```

# 项目案例

- 编程实现Person类的封装，特征有：姓名和年龄，提供打印所有特征的行为  
编程实现Student类继承Person类的封装。

# 继承中的构造方法

- 子类的构造方法中必须通过super关键字调用父类的构造方法，这样可以妥善的初始化继承自父类的成员变量。
- 如果子类的构造方法中没有调用父类的构造方法，也没有使用this()调用本类构造，Java编译器会自动的加入对父类无参构造方法的调用（如果该父类没有无参的构造方法，会有编译错误）。
- this()和super() 必须出现在构造的第一行，因此不可能同时出现。

```
Circle() {}  
Circle(int x, int y, int r) {  
    super(x, y);  
    this.r = r;  
}
```

编译器会加入无参的调用（"super()"）。

通过super关键字调用父类的构造方法。

# 继承的注意事项

- 继承不是随便写的，逻辑上存在 is a 关系才能使用继承。子类 is a 父类。
- Java语言不支持多重继承，一个类只能继承一个父类。但一个父类可以有多个子类。

# 方法的重写(Override)

- 有些方法从父类继承过来但是不太适合子类直接使用，子类可以使用 方法重写(覆盖) 去改变父类继承过来的方法。
- 方法重写的原则：
  - 1 相同的方法名称，相同的参数列表，相同的返回值类型或者返回子类。
  - 2 访问权限不能变小，可以变大。
  - 3 不能抛出更大的异常。
- 在子类重写的方法中，可以通过super关键字调用父类的“原始”方法。
- static的方法重写以后还是static的。

# 方法重写的实例

```
class Shape{  
    int x;  
    int y;
```

```
    public boolean contains (int x, int y){  
        ...
```

```
    }  
}
```

```
class Circle extends Shape {  
    int r;
```

```
    public boolean contains (int x, int y) {  
        .....  
    }
```

```
}
```

# 项目案例

- 编程实现Student类对父类Person中打印特征方法的重写。



# 总结与答疑

技术资料





IT兄弟连  
ITXDL.CN

变态严管 让学习成为一种习惯