

面向对象工程思想

October 27, 2020

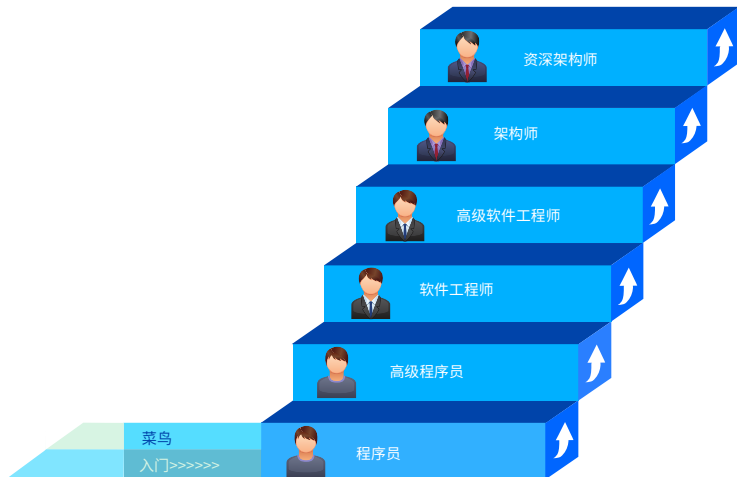
概述

1 抽象类

2 接口

3 多态

Java 工程师的进阶之路

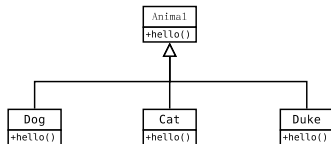


继承回头看

请脑补动物
怎么打招呼?

```
public class Animal {  
    public void hello() {  
        System.out.println("动物在打招呼");  
    }  
}
```

```
public class Dog extends Animal {  
}
```



直接继承真的完美吗？

- 父类实现某些方法可能没有意义，只有子类的具体实现才是有意义的，比如 `Animal.hello()`。

直接继承真的完美吗？

- 父类实现某些方法可能没有意义，只有子类的具体实现才是有意义的，比如 `Animal.hello()`。
- 不允许根据类创建对象，比如禁止 `new Animal()`：强制继承。

定义抽象类

抽象类的定义有两种情形：

代码清单 1: 抽象类可以没有抽象方法

```
1 public abstract class Animal {  
2  
3     public void hello() {  
4         System.out.println("动物在  
           打招呼");  
5     }  
6 }
```

代码清单 2: 具有抽象方法的类必须定义为抽象类

```
1 public abstract class Animal {  
2  
3     public abstract void hello();  
4 }
```

继承抽象类

关键词： extends

```
1 public class Dog extends Animal {  
2  
3     @Override  
4     public void hello() {  
5         System.out.println("汪汪");  
6     }  
7  
8 }
```


纯的抽象类

- 接口 = 纯的抽象类

定义接口

- interface
- 接口的特点

实现接口

- implements
- 实现多个接口，打印机的例子

对象的类型

面向接口的编程

对象的类型、向上转型

对象的存储模型

多态：子类引用父类对象的自然结果