

Java核心技术

第四章 面向对象思想、类和对象 第一节 面向对象思想 华东师范大学 陈良育

面向对象(1)



- 现实世界是由对象和对象之间相互作用共同组成的。
- 每个对象有自己的特有属性,也有自己专有的方法。外部对象想要调用这些方法,可以向它请求并传入参数,等方法执行结束后,返回结果。
- 对象=属性+方法。
- 对象的规范=属性定义+方法定义。

面向对象(2)



- · Java中,引入对象和类的概念
 - 对象是一个变量(具体的东西);
 - 类就是类型(是规范,是定义),从万千对象中抽取共性;
 - 类规定了对象应该有的属性内容和方法;
 - 对象是类的具体实现,是活生生的;
 - 例如: 土豆丝菜谱是类, 一盘土豆丝是对象。

面向对象(3)



- · 从程序发展的角度来理解,OO(Oriented Object)是对OP(Oriented Procedure)的一种改进。
- · OP的典型代表是C和Pascal。更强调方法动作,所有的变量 是被动参与进来,没有自主决定权。
- · OO的方法属于每个对象的。能否实现是由每个对象说了算的,有主人翁精神。

面向对象(4)



- · OP的例子
- ·OO的例子
- 计算机的发展和社会发展也是相似之处
 - 更强调方法的执行主体
 - 增加类型的重用部分(内容和行为)

面向对象(5)



- 变量定义的变迁: 更加功能强大
- 基本类型(一种变量)=》结构体(多种变量捆绑)=》类 (多种变量+方法)。
- · 类可以继承: 子类可以继承父类所有内容(不能直接访问 private成员)
 - 基本类型无法做到
 - 结构体只能做到全包含,不能控制包含粒度。

面向对象(6)



- · 基本变量例子: int a; float b;
- 结构体变量例子

```
//c语言结构体不能包含函数
struct Simple
{
  int a;
  double b;
}; //成员默认均为public
```

```
struct Complex
{
   char c;
   struct Simple foo;
   //Complex必须包含Simple所有内容
}
```

面向对象(7)

• 类/对象例子

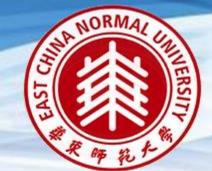
```
public class Father {
    private int money = 100; //私有
    long mobile = 1399999999991;

    public void hello() {
        System.out.println("hello");
    }
}
```



```
public class Son extends Father {
   public void hi() {
       //子类可以扩展自己的成员方法
       System.out.println("hi~~~~");
   public static void main(String[] a) {
       Son s = new Son();
       System.out.println(s.mobile); //Son没有定义mobile, 而是通过父类继承的
       //System.out.println(s.money); //error 父类的money是私有的,子类无法直接访问
       s.hello(); //Son没有定义f1, 而是通过父类继承的
       s.hi(); //Son可以自定义自己的成员方法
```

面向对象(8)



- · OO最早的始于Simula 67,成形于20世纪70年代Smalltalk,
- · 当前最主要的代表是C++和Java。
- 面向对象语言主要特点
 - 识认性:辨识、认定对象。
 - 类别性: 归类对象。
 - 多态性: 同类的对象有共性, 也有差异。
 - -继承性: 子类继承父类所有的东西。

面向对象(9)



· OO总结

- 现实世界是由对象和对象之间相互作用共同组成的
- 对象不仅包括成员变量,还有成员方法
- 对象的内容可以继承

代码(1) OPExample.java



```
public class OPExample
   public static void main(String[] args) {
       int a,b,c;
       a=1;
       b=2;
       c=add(a,b);
      System.out.println("c is " +c);
   public static int add(int m, int n) { //函数定义
       return m+n;
```

代码(2) OOExample.java



```
public class OOExample {
    private int a;
    public void setA(int a) {
        this.a = a;
    public int add(int b) {
        return this.a + b;
    public static void main(String[] a) {
        int b = 5;
        OOExample obj = new OOExample();
        obj.setA(10);
        System.out.println(obj.add(b));
```

代码(3) Father.java



```
public class Father {
   //私有变量 子类无法访问
   private int money = 100;
   long mobile = 139999999991;
   public void hello() {
       System.out.println("hello");
```

代码(4) Son.java



```
public class Son extends Father {
   public void hi() {
      //子类可以扩展自己的成员方法
      System.out.println("hi~~~~");
   public static void main(String[] a) {
      Son s = new Son();
      System.out.println(s.mobile); //Son没有定义mobile, 而是通过父类继承的
      //System.out.println(s.money); //error 父类的money是私有的,子类无法直接访问
      s.hello(); //Son没有定义f1, 而是通过父类继承的
      s.hi(); //Son可以自定义自己的成员方法
```



谢 谢!