## 实验03\_1 面向对象编程基础简单的类和对象

|  |
| --- |
| 实验目的：   1. 掌握类的定义，熟悉属性、构造函数、方法的作用，掌握用类作为类型声明变量和方法返回值； 2. 理解类和对象的区别，掌握构造函数的使用，熟悉通过对象名引用实例的方法和属性； 3. 理解方法中的形参和实参传递在传递基本类型和引用类型时的差异 |
| 实验数据记录及分析（或程序及运行结果）   1. 写一个名为Rectangle的类表示矩形。其属性包括宽width、高height和颜色color，width和height都是double型的，而color则是String类型的。要求该类具有：    1. 使用构造函数完成各属性的初始赋值    2. 使用getter和setter的形式完成属性的访问及修改    3. 提供计算面积的getArea()方法   **package** rectangle;  **public** **class** Rectangle {  **double** width;  **double** height;  String color;    **public** **double** getwidth() {  **return** width;  }    **public** **void** setwidth(**double** width) {  **this**.width = width;  }    **public** **double** getheight() {  **return** height;  }    **public** **void** setheight(**double** height) {  **this**.height = height;  }    **public** String getcolor() {  **return** color;  }    **public** **void** setcolor(String color) {  **this**.color = color;  }    **public** **double** getArea() {  **return** **this**.width\***this**.height;  }    **public** **static** **void** main(String[] args) {  Rectangle picture = **new** Rectangle();  picture.setwidth(3);  picture.setheight(4);  picture.setcolor("red");  System.***out***.println("矩形的宽为:"+picture.getwidth());  System.***out***.println("矩形的高为:"+picture.getheight());  System.***out***.println("矩形的颜色为:"+picture.getcolor());  System.***out***.println("矩形的面积为:"+picture.getArea());  }  }     1. 一副牌Deck有52张扑克Card组成(不含王牌)，每张牌有自己的牌型suit（用char类型）和点数rank（用String类型），补充完整下面的类的定义。   **public** **class** Card {  **public** Card(**char** suit, String rank) {  **this**.suit = suit;  **this**.rank = rank;  }    **public** String toString() {  **return** suit+rank;  }  **public** **static** **void** main(String[] args){  Card c=**new** Card('红',"10");  System.*out*.println(c);  }  }  3、输入下面的类，结合上题中的Card类，完成下面的要求  public class JLab{  static final char suits[] ={'H','S','D','C'};  static final String ranks[] ={"A","2","3","4","5","6","7","8","9","10","J","Q","K"};  static Card[] cards=new Card[52];  /\*\*  \* 初始化扑克牌数组  \*/  public static void init(){  for(int i=0;i<suits.length;i++){  for(int j=0;j<ranks.length;j++){  cards[i\*ranks.length+j]=new Card(suits[i],ranks[j]);  }  }  }  public static void swap1(Card c1,Card c2){  Card c=c1;  c1=c2;  c2=c;  }  public static void swap1(int i,int j){  Card c=cards[i];  cards[i]=cards[j];  cards[j]=c;  }  public static void swap2(Card c1,Card c2){  char s=c1.suit;  c1.suit=c2.suit;  c2.suit=s;  String r=c1.rank;  c1.rank=c2.rank;  c2.rank=r;  }  public static void main(final String[] args) {  init();  //任取两张牌  Card c1=cards[10];  Card c2=cards[12];  System.out.println("第11张牌是："+c1+"\t第13张牌是："+c2);  }  }  1) 在main方法中，添加下面几行语句，写出并分析结果  swap1(c1,c2);  System.out.println("执行swap1(c1,c2)后");  System.out.println("c1引用的牌是："+c1+"\tc2引用的牌是："+c2);  System.out.println("第11张牌是："+cards[10]+"\t第13张牌是："+cards[12]);  第11张牌是：HJ 第13张牌是：HK  执行swap1(c1,c2)后  c1引用的牌是：HJ c2引用的牌是：HK  第11张牌是：HJ 第13张牌是：HK  2) 在main方法中，去掉刚才添加的语句，添加下面几行语句，写出并分析结果  swap1(10,12);  System.out.println("执行swap1(10,12)后");  System.out.println("c1引用的牌是："+c1+"\tc2引用的牌是："+c2);  System.out.println("第11张牌是："+cards[10]+"\t第13张牌是："+cards[12]);  第11张牌是：HJ 第13张牌是：HK  执行swap1(10,12)后  c1引用的牌是：HJ c2引用的牌是：HK  第11张牌是：HK 第13张牌是：HJ  3) 在main方法中，去掉刚才添加的语句，添加下面几行语句，写出并分析结果  swap2(c1,c2);  System.out.println("执行swap2(c1,c2)后");  System.out.println("c1引用的牌是："+c1+"\tc2引用的牌是："+c2);  System.out.println("第11张牌是："+cards[10]+"\t第13张牌是："+cards[12]);  第11张牌是：HJ 第13张牌是：HK  执行swap2(c1,c2)后  c1引用的牌是：HK c2引用的牌是：HJ  第11张牌是：HK 第13张牌是：HJ  4、程序运行后的输出是什么？  class TestReference{  public static void main(String[] args){  int x=2;  TestReference tr = new TestReference();  System.out.print(x);  tr.change(x);  System.out.print(x);  }  public void change(int num){  num = num + 1;  }  }  **22**  5、首先预测下列程序的运行结果，然后编译、执行以验证你的预测：  **class MyParts {**  **public static int x = 7;**  **public int y = 3;**  **}**  **public class JLab0402{**  **public static void main(String[] args) {**  **MyParts a = new MyParts();**  **MyParts b = new MyParts();**  **a.y = 5;**  **b.y = 6;**  **a.x = 1;**  **b.x = 2;**  **System.out.println("a.y = " + a.y);**  **System.out.println("b.y = " + b.y);**  **System.out.println("a.x = " + a.x);**  **System.out.println("b.x = " + b.x);**  **}**  **}**  根据上述程序的结果理解实例成员与类成员的区别。  a.y = 5b.y = 6a.x = 2b.x = 2  **6、对下面的源程序进行编译，排错并运行。**  class StaticDemo {  static int x;  int y;  public static int getX() {  return x;  }  public static void setX(int newX) {  x = newX;  }  public int getY() {  return y;  }  public void setY(int newY) {  y = newY;  }  }  public class KY3\_4 {  public static void main(String[] args) {  System.out.println("静态变量x="+StaticDemo.getX());  System.out.println("实例变量y="+StaticDemo.getY());  StaticDemo a= new StaticDemo();  StaticDemo b= new StaticDemo();  a.setX(1);  a.setY(2);  b.setX(3);  b.setY(4);  System.out.println("静态变量a.x="+a.getX());  System.out.println("实例变量a.y="+a.getY());  System.out.println("静态变量b.x="+b.getX());  System.out.println("实例变量b.y="+b.getY());  }  }  Exception in thread "main" java.lang.Error: 无法解析的编译问题：  不能对类型 StaticDemo 中的非静态方法 getY（）进行静态引用  删掉System.out.println("实例变量y="+StaticDemo.getY());  静态变量x=0静态变量a.x=3实例变量a.y=2静态变量b.x=3实例变量b.y=4  7、解释下面的程序运行结果输出为什么是null  public class My {  String s;  public void My(){  s = "Constructor";  }  public void go() {  System.out.println(s);  }  public static void main(String args[]) {  My m = new My();  m.go();  }  }  在main方法中没有调用My()方法； |
| 实验小结： |