

第2章-类与对象的基本概念

抽象：过程抽象（如C语言的函数，Java语言的方法），数据抽象（将要处理的数据和这些数据上的操作结合在一起，抽象为类）

封装

继承

Java语言仅支持单继承。

多态

声明一个引用变量时并没有对象生成。

```
1 | Clock aclock; // aclock将用于存储该类对象的引用
```

对象的创建：

```
1 | aclock = new Clock();
```

其作用是（a）在内存中为对象分配内存空间。（2）返回对象的引用。

也可以被赋为空值（默认也是null）

没有实例static修饰的变量称为实例变量（Instance Variables）

类变量，也称为静态变量，声明时需加static修饰符。

```
1 | class Circle{
2 |     static double pi = 3.14;
3 |     double radius;
4 | }
5 |
6 | public class ClassVariableTester{
7 |     public static void main(String args[]){
8 |         Circle x = new Circle();
9 |         System.out.println(x.pi);
10 |        Circle y = new Circle();
11 |        System.out.println(y.pi);
12 |        y.pi = 3;
13 |        System.out.println(x.pi);
14 |    }
15 | }
```

实例方法

不同的类中可以声明相同方法名。使用时，系统会根据接收者对象的类型找到相应类的方法。

```

1  class Test{
2      public static int twice(int x){
3          return 2*x;
4      }
5  }
6
7  public class ClassFunctionTester{
8      public static void main(String args[]){
9          System.out.println(Test.twice(5));
10         // 类方法可以在不建立对象的情况下用类名直接调用
11     }
12 }

```

包是一组类的集合。

面向对象程序设计为了体现细节隐藏，应该尽可能把数据成员设计为private（封装）

如果要允许其他类访问private值，就需要在类中声明相应的公有方法。通常有两类典型方法用于访问属性值，get方法及set方法。

如果形式参数名与实例变量名相同，则需要在实例变量名之前加this关键字，否则系统会将实例变量当成形式参数。

```

1  class circle{
2      private double radius;
3      public void setRadius(double radius){
4          this.radius = radius;
5      }
6  }

```

构造方法是一种和类同名的特殊方法。若类没有定义构造方法，则系统自动提供默认构造方法。

方法名与类名相同。没有返回类型，修饰符void也不能有。不能在程序中显式的调用。在生成一个对象时，系统自动调用该类的构造方法为其初始化。

可以使用this关键字在一个构造方法中调用另外的构造方法。

枚举类型特点：

枚举类型是类，而不是简单的整数类型，枚举值是类的对象。

枚举值是public、static、final的。

JVM为了保证每一个枚举类元素的唯一实例，不允许外部进行new，所以讲构造函数设计成private。（妙啊）

toString()方法的说明：必须声明为public，返回类型为String，方法名必须为toString，没有参数，在方法体中，不要使用输出方法System.out.println()

