

Lesson1 面向对象编程

2023/03/09

part1 基本数据类型

序号	数据类型	位数	默认值	取值范围	举例说明
1	byte(位)	8	0	$-2^7 - 2^7-1$	byte b = 10;
2	short(短整数)	16	0	$-2^{15} - 2^{15}-1$	short s = 10;
3	int(整数)	32	0	$-2^{31} - 2^{31}-1$	int i = 10;
4	long(长整数)	64	0	$-2^{63} - 2^{63}-1$	long l = 10l;
5	float(单精度)	32	0.0	$-2^{31} - 2^{31}-1$	float f = 10.0f;
6	double(双精度)	64	0.0	$-2^{63} - 2^{63}-1$	double d = 10.0d;
7	char(字符)	16	空	$0 - 2^{16}-1$	char c = 'c';
8	boolean(布尔值)	8	false	true、false	boolean b = true;

part2 面向对象编程

1. 面向对象编程

面向过程的程序设计把计算机程序视为一系列的命令集合，即一组函数的顺序执行。为了简化程序设计，面向过程把函数继续切分为子函数，即把大块函数通过切割成小块函数来降低系统的复杂度。

面向对象的程序设计把计算机程序视为一组对象的集合，而每个对象都可以接收其他对象发过来的消息，并处理这些消息，计算机程序的执行就是一系列消息在各个对象之间传递。

2.对象是一个抽象的概念

在Java语言中，在定义一个对象的时候，会把对象的**状态**、**行为**、**标识**等信息写在一个class代码块里，这个代码块被称为一个类。类通过实例化创建一个对象。

3.继承

继承的出现，是为了减少我们复制另一个类的属性或方法。

继承指的是有一个类A，它自身的属性和要做的是和已有的类B大多相似，那么类A可以使用extends关键字去继承类B，在关系上类B被称为父类，或者基类，类A被称为子类。如下所示：

```
public class ClassA extends ClassB{  
  
}
```

举例，有一个形状类Shape，它有color属性和getInfo()行为；

```
public class Shape {
    // 成员变量
    private String color;

    public void setColor(String color){
        this.color = color;
    }

    public String getColor(){
        return this.color;
    }

    // 成员方法
    void getInfo(){
        System.out.println("color: " + this.color);
    }
}
```

后来有了具体的形状出现，如Circle、Square等，他们除了有color属性和getInfo()行为，还会有一个属于自身的行为calArea()，这个称为扩展子类的行为。

```
public class Circle extends Shape{
    // 继承父类的非私有成员变量和方法

    int radius = 2;

    public void calArea() {
        System.out.println("area: " + 3.14 * radius * radius);
    }
}
```

4.多态

面向对象语言使用了一种后期绑定的机制，直到运行到这段代码时才确定哪段代码会被调用，同一方法显现出多种模态，即为多态。

多态的实现方式有三种：**重写父类的方法、实现抽象类的虚方法、实现接口。**

- **重写父类的方法**

通过@Override注解，重写父类已有的方法，使子类的该方法发生变化，从而实现多态。

```
// 子类Circle继承Shape
public class Circle extends Shape{
    @Override
    public void getInfo(){
        System.out.println("circle's color is: " + this.getColor()); // 通过重写实现多态
    }
}
```

- 实现抽象类的虚方法

通过abstract修饰词定义抽象类和抽象方法，再在子类中实现具体的抽象方法，来实现多态。

```
// 抽象类Shape
public abstract class Shape {
    // 抽象方法 move
    abstract void move();
}
```

```
// 通过继承抽象类，并实现抽象方法产生不同的行为
public class Circle extends Shape{
    @Override
    void move() {
        System.out.println("gun");
    }
}
```

- 实现接口

在接口ShapeUtils中定义一个calArea()，通过实现接口来产生不同的行为，从而实现多态。

```
// 接口类
public interface ShapeUtils {
    void calArea();
}
```

```
public class Circle extends Shape implements ShapeUtils{
    int radius = 2;

    @Override
    public void calArea() {
        System.out.println("area: " + 3.14 * radius * radius);
    }
}
```

一个类只能继承一个父类，但是可以实现多个接口

5.引用

JAVA中没有指针，JAVA中有指针，哪个说法对呢？

答：都对，JAVA中没有指针，因为我们不能对指针直接操作，像C++那样用->来访问变量。

JAVA有指针，因为JDK中封装了指针。

```
类名 引用变量 = new 类名();
```

对于这句话的解释是：创建一个对象，将对象的地址赋值给引用变量，此时变量就持有了对象的引用；而此时我们就可以通过引用变量来访问对象中的所有成员。

实例化对象类；把对象的地址赋值给了变量，此时变量指向了对象，那么这不就是C、C++中的指针吗，和C++、C里面的指针定义很像。

6.深拷贝与浅拷贝

浅拷贝只复制指向某个对象的指针，而不复制对象本身，新旧对象还是共享同一块内存。

深拷贝会另外创建一个一模一样的对象，新对象跟原对象不共享内存，修改新对象不会改到原对象，是“值”而不是“引用”。