Java 核心技术

第四章 面向对象思想、类和对象

第二节 Java类和对象

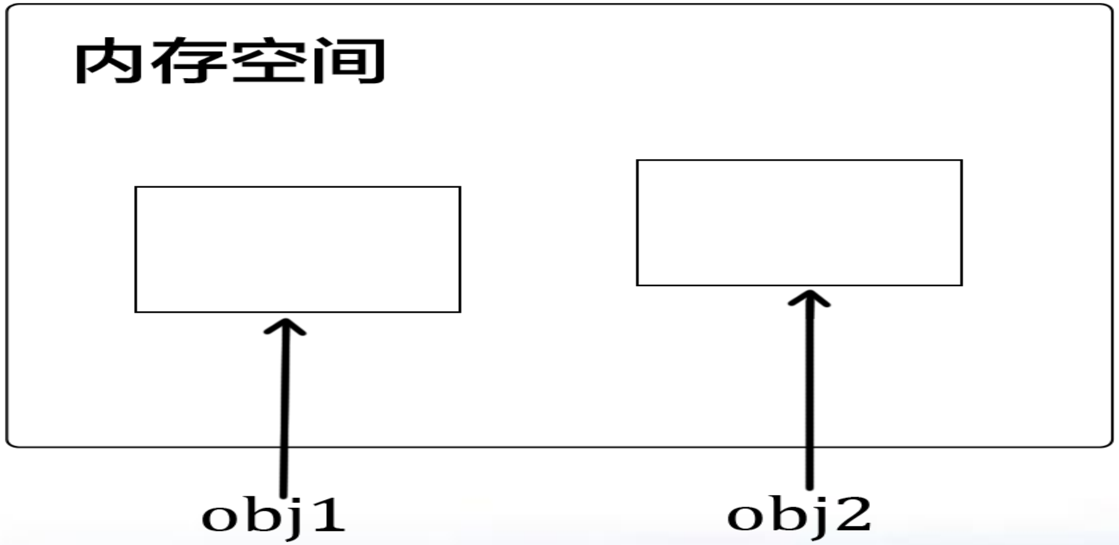
华东师范大学 陈良育

类和对象(1)

- 最简单的类 class A{ } //没有任何属性和行为
- 对象 A obj = new A();
- 类是定义，是规范，是“死”的东西
- 对象是实例，是类的一个实现，是一个具体的东西。
- 打个比方
  - 类等价于一个土豆丝菜谱
  - 对象是根据类制作出的对象，等价于一盘土豆丝

类和对象(2)

- A obj1 = new A(); A obj2 = new A();
- 以上有2个对象，它们的类型都是A，但是这是两个不同的对象，在内存中有不同的存放地址。因此，没有两个对象是完全一样的。



类和对象(3)

- A obj = new A();
- obj 可以看作是内存中一个对象(包括若干个数据)的句柄
- 在C/C++中，obj称为指针，在Java中称为Reference
- 对象赋值是Reference赋值，而基本类型是直接值拷贝
- 参看例子ReferenceTest.java和ArgumentPassingTest.java，来查看基本类型赋值和普通对象赋值的不同

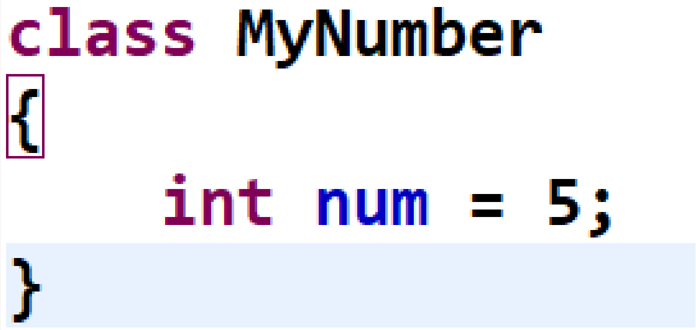
类和对象(4)

- 产生一个对象，A obj = new A();
  - 99%的情况是用new 关键字，还有1%是用克隆和反射生成
- new出对象后，内部属性值默认是？例子Initialization.java
  - short 0   int 0   long 0L
  - boolean false
  - char '\u0000‘
  - byte 0
  - float 0.0f
  - double 0.0d

类和对象(5)

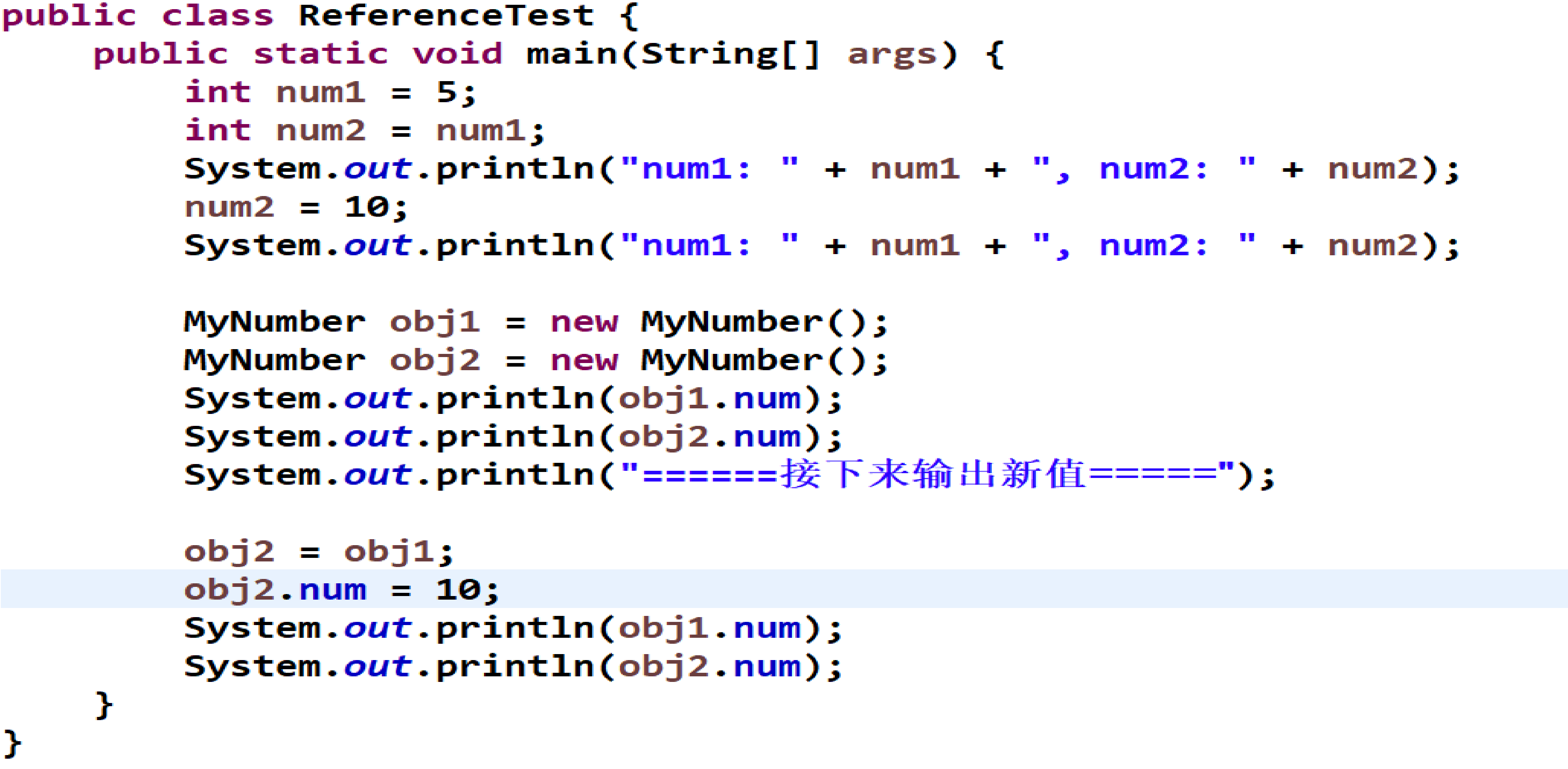
- 总结
  - 类是规范，对象是根据规范产生出的实例
  - 基本型别赋值是拷贝赋值，对象赋值是reference赋值
  - 类成员变量有初值，函数临时变量必须初始化

代码(1) MyNumber.java



```
class MyNumber
{
    int num = 5;
}
```

代码(2) ReferenceTest.java

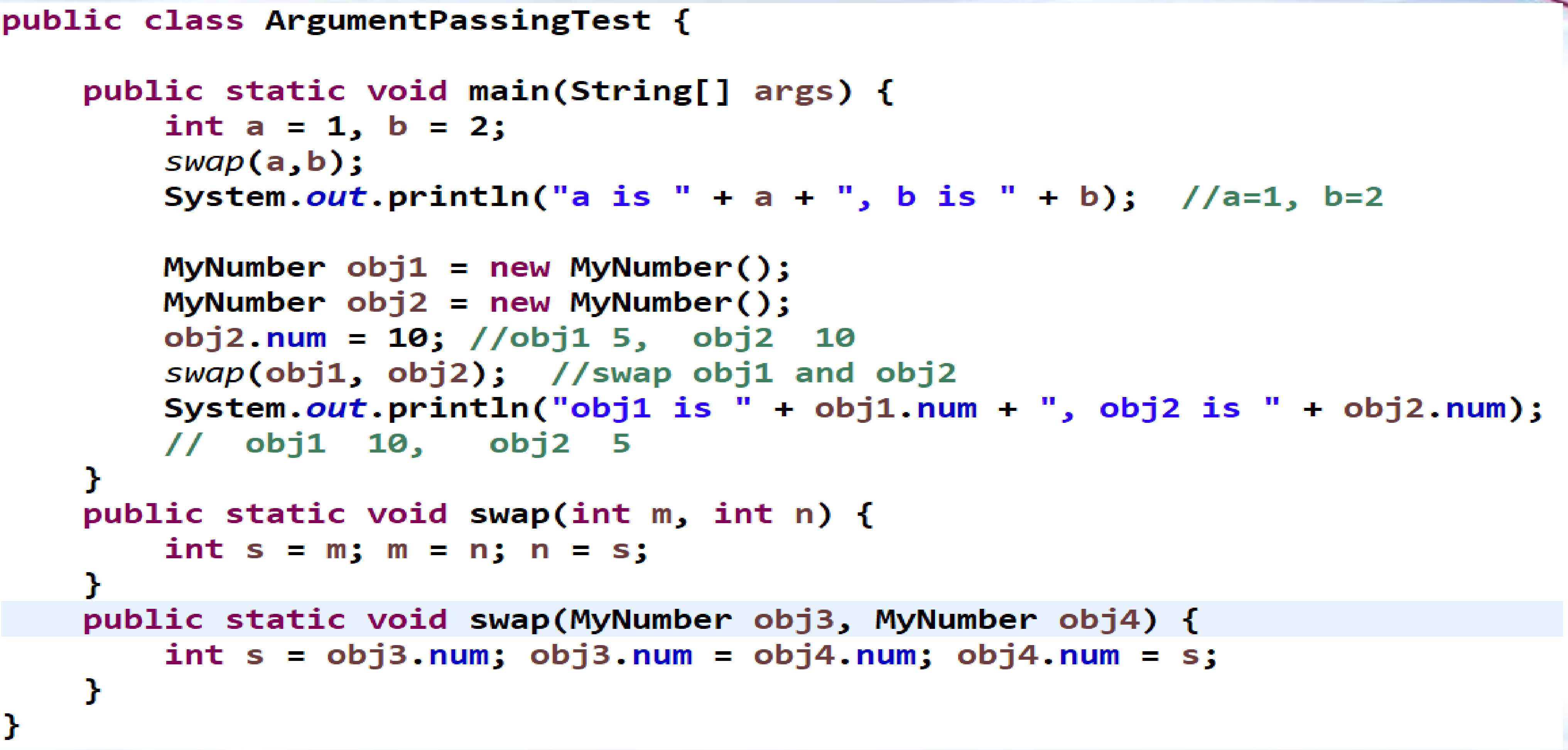


```java
public class ReferenceTest {
    public static void main(String[] args) {
        int num1 = 5;
        int num2 = num1;
        System.out.println("num1: " + num1 + ", num2: " + num2);
        num2 = 10;
        System.out.println("num1: " + num1 + ", num2: " + num2);

        MyNumber obj1 = new MyNumber();
        MyNumber obj2 = new MyNumber();
        System.out.println(obj1.num);
        System.out.println(obj2.num);
        System.out.println("======接下来输出新值=====");

        obj2 = obj1;
        obj2.num = 10;
        System.out.println(obj1.num);
        System.out.println(obj2.num);
    }
}
```

代码(3) ArgumentPassingTest.java

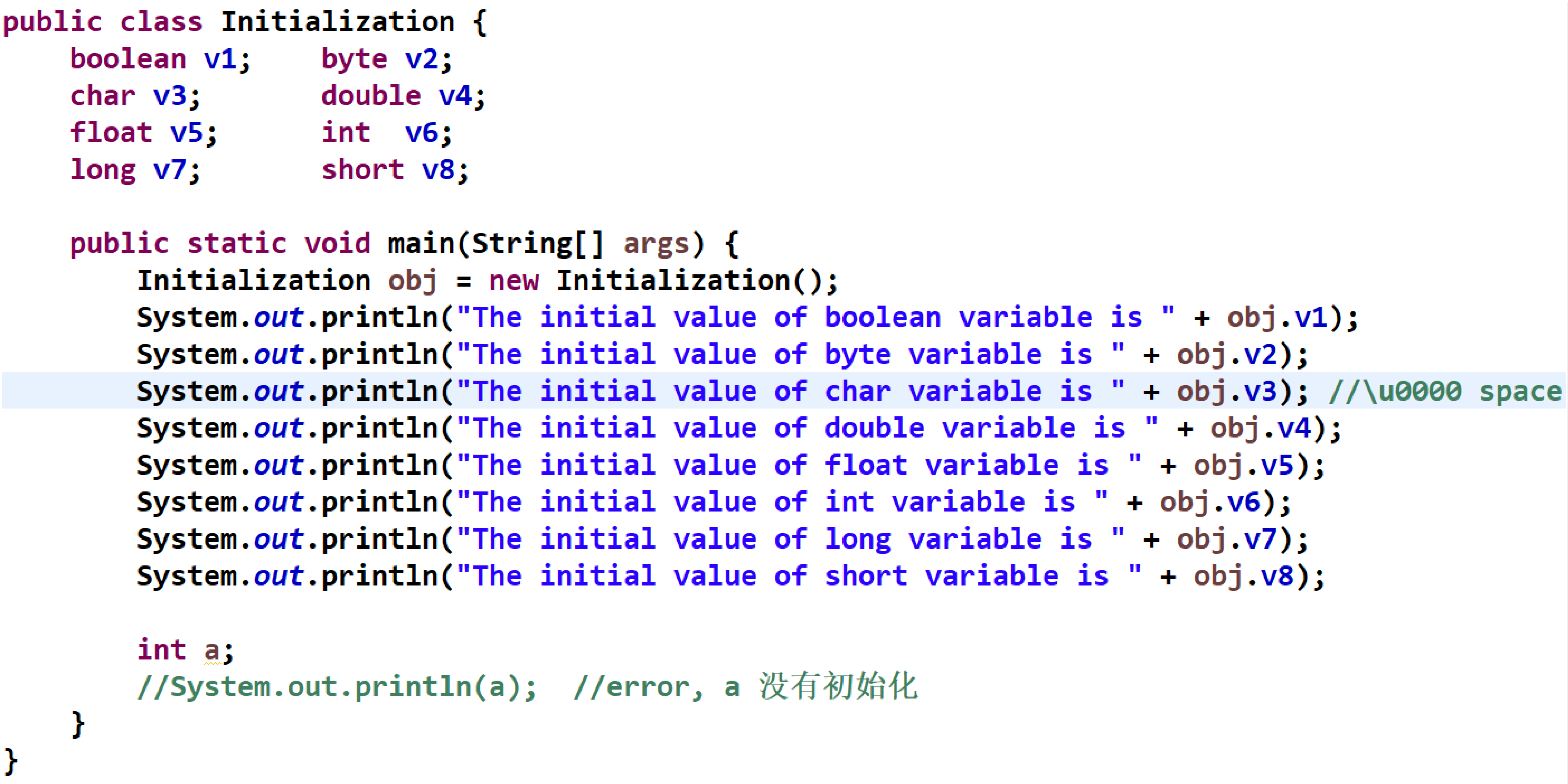


```
public class ArgumentPassingTest {

    public static void main(String[] args) {
        int a = 1, b = 2;
        swap(a,b);
        System.out.println("a is " + a + ", b is " + b);  //a=1, b=2

        MyNumber obj1 = new MyNumber();
        MyNumber obj2 = new MyNumber();
        obj2.num = 10; //obj1 5,  obj2  10
        swap(obj1, obj2);  //swap obj1 and obj2
        System.out.println("obj1 is " + obj1.num + ", obj2 is " + obj2.num);
        //  obj1  10,   obj2  5
    }
    public static void swap(int m, int n) {
        int s = m; m = n; n = s;
    }
    public static void swap(MyNumber obj3, MyNumber obj4) {
        int s = obj3.num; obj3.num = obj4.num; obj4.num = s;
    }
}
```

代码(4) Initialization.java



```java
public class Initialization {
    boolean v1;    byte v2;
    char v3;       double v4;
    float v5;      int  v6;
    long v7;       short v8;

    public static void main(String[] args) {
        Initialization obj = new Initialization();
        System.out.println("The initial value of boolean variable is " + obj.v1);
        System.out.println("The initial value of byte variable is " + obj.v2);
        System.out.println("The initial value of char variable is " + obj.v3); //\u0000 space
        System.out.println("The initial value of double variable is " + obj.v4);
        System.out.println("The initial value of float variable is " + obj.v5);
        System.out.println("The initial value of int variable is " + obj.v6);
        System.out.println("The initial value of long variable is " + obj.v7);
        System.out.println("The initial value of short variable is " + obj.v8);

        int a;
        //System.out.println(a);  //error, a 没有初始化
    }
}
```

谢 谢！