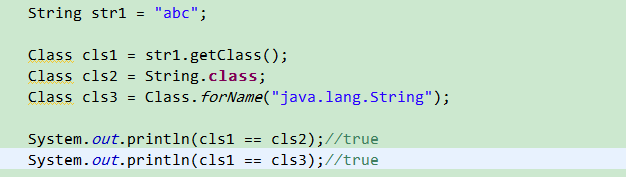
1. 反射的基石：Class类
   1. Java程序中的各个Java类属于同一类事物，描述这类事物的Java类名就是Class。
   2. Java类用于描述一类事物的共性，该类事物有什么属性、没有什么属性，至于这个属性的值是什么，则是由这个类的实例对象来确定的，不同的实例对象有不同的属性值。同理，Java程序中的各个Java类也可以用Class这个类来描述。其中描述有类的名字、类的访问属性、类所属的包名、字段名称的列表、方法名称的列表等等。

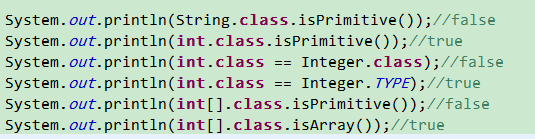
例：人—>Person 等价于 Java类🡪Class

* 1. 对比：Person类代表人，它的实例对象就是张三、李四这样一个个具体的人，Class类代表Java类，它的实例对象就是各个Java类在内存中的字节码，比如Person类的字节码、Date类的字节码。

1. 如何得到某个类的字节码
   1. 类名.class，例如System.class;
   2. 对象.getClass()，例如new Date().getClass();
   3. Class.forName(“类名”)，例如Class.forName(“java.util.Date”)

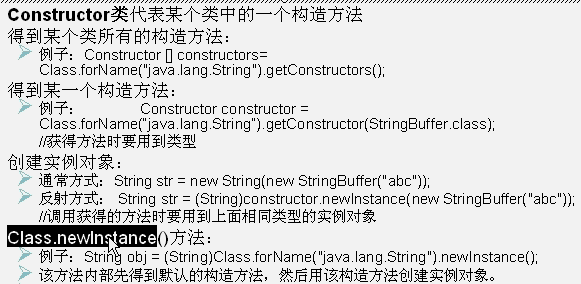


1. Java中9个预定义的Class实例对象(8种基本数据类型+void类型)

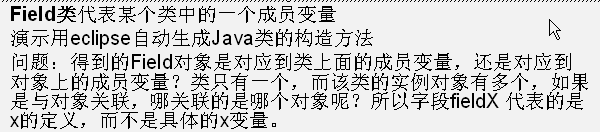


总之，只要是在源程序中出现的类型，都有各自的Class实例对象，例如int[]，void等

1. 反射：**反射就是把Java类中的各种成分映射成相应的Java类**。例如，一个Java类中用一个Class类的对象来表示，一个类中的组成部分：成员变量、方法、构造方法、包等等信息也用一个个的Java类来表示，就像一个汽车是一个类，汽车中的发动机、变速箱等等也是一个个的类。表示Java类的Class类显然要提供一系列的方法来获得其中的变量、方法、构造方法、修饰符、包等信息，这些信息就是用相应类的实例对象来表示，他们是Field、Method、Constructor、Package等等。
2. 一个类中的每个成员都可以用相应的反射API类的一个实例对象来表示，通过调用Class类的方法可以得到这些实例对象。
3. Constructor类

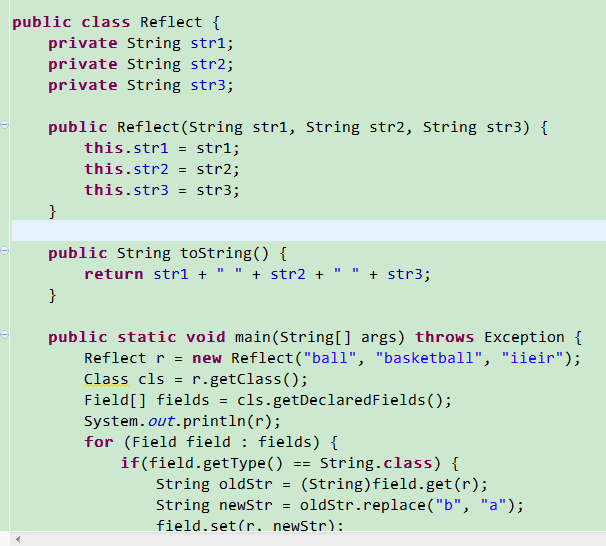


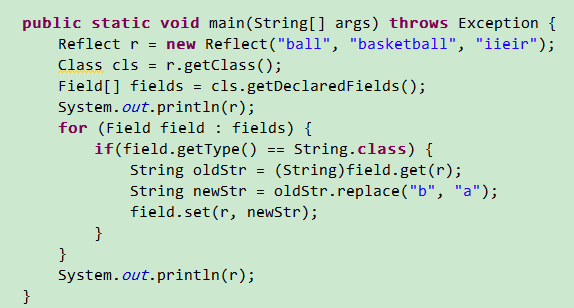
1. Field类



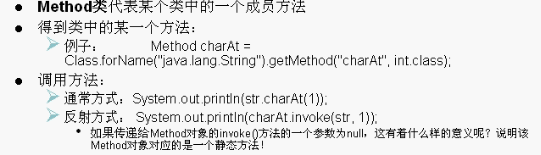


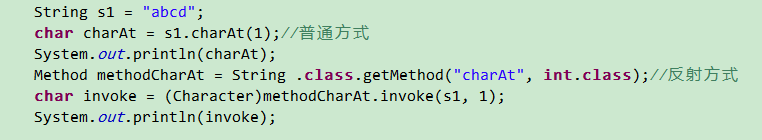
实例：将任意一个对象中的所有String类型的成员变量对应的字符串内容中的‘b’改成‘a’：



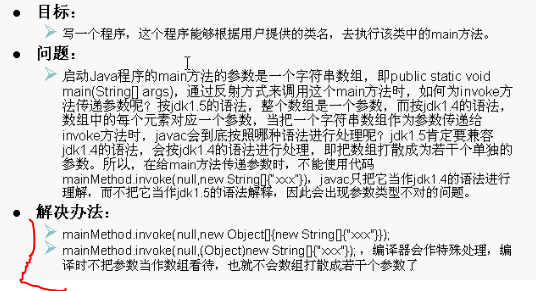


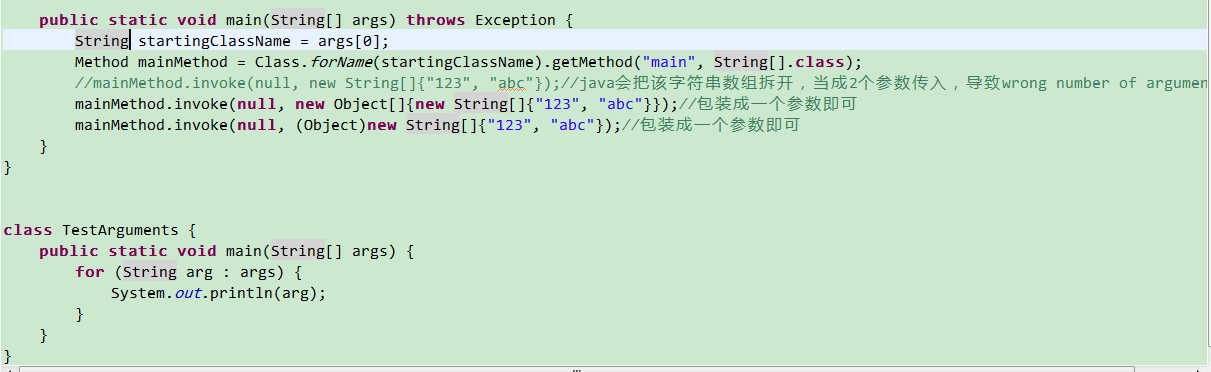
1. Method类



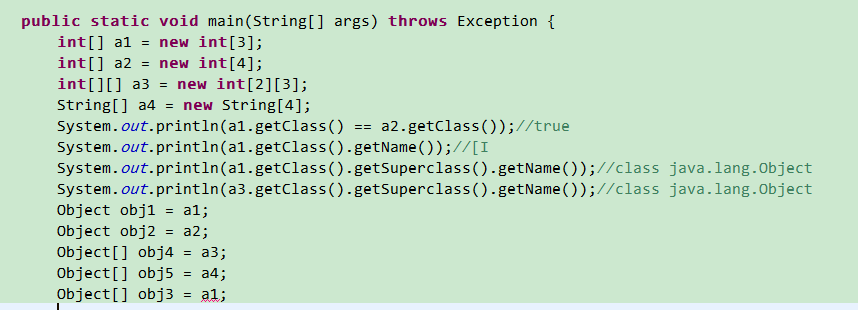


用反射方式执行某个类中的main方法：根据用户提供的类名，去执行该类中的main方法。

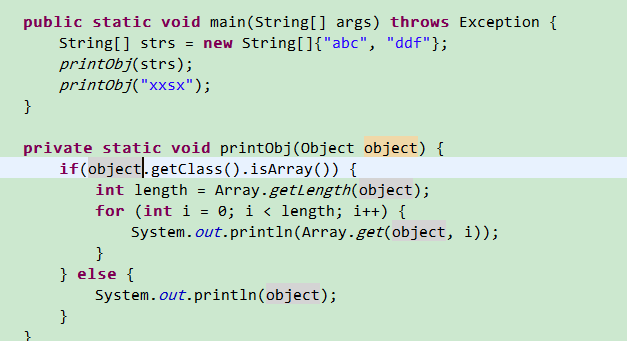




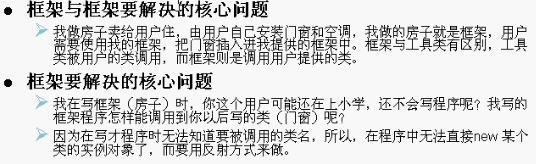
1. 数组的反射
   1. 具有相同维数和元素类型的数组属于同一个类型，即具有相同的Class实例对象
   2. 代表数组的Class实例对象的getSuperClass()方法返回的父类为Object类对应的Class
   3. 基本数据类型的一维数组可以被当做Object类型使用，不能当做Object[]类型使用；非基本类型的一维数组，既可以当做Object类型使用，又可以当做Object[]类型使用。



* 1. 采用java.lang.reflect.Array工具类用于完成对数组的反射操作



1. 反射的作用：实现框架功能



1. 用类加载器的方法管理资源和配置文件

