

代码块：{ }

代码块的分类：

- 1.普通代码块：在方法内的代码块
- 2.构造代码块：在类中的代码块
- 3.静态代码块：由static修饰的代码块

普通代码块：

调用方法时，普通代码从上向下依次执行

构造代码块：

创建对象时，构造代码块从上向下依次执行

静态代码块：

当类被第一次加载时执行

静态代码块从上向下执行完，再执行静态构造代码块

类的分类：

- 1.普通类
- 2.抽象类
- 3.final修饰的类

一个文件里可以写多个类：

一个文件里只能有一个public修饰的类

文件名要和public修饰的类同名

编译后，有几个类，形成几个.class文件

- 1.内部类
- 2.外部类
- 3.匿名内部类

内部类：

- 1.为外部类服务的，一般很少创建内部类

2.要实例化内部类，需要先实例化外部类

3.定义在类的外部类

格式：

外部类.内部类 内部类名 = 外部类对象.new 内部类();

外部类：

最外层类即文件名称相同的类

匿名内部类：

定义在方法内

格式：

```
new 类名/接口 ( ) {  
    重写的方法  
}
```

反射：

通过反射技术可以获得一个类的属性和方法

获取对象实质是获取字节码文件

三种方法：

方法一：

Class<类> 名 = 类.class;

方法二：

先生成对象

Class<? extends 类> 名 = 对象.getClass();

方法三：（常用方法）

Class<?> 名 = Class.forName("全类名");

通过反射机制获取构造方法并实例化：

Class 名 = Class.forName("全类名");

名.getConstructor(); 获取public构造方法

名.getConstructors(); 获取所有public的构造方法

名.getDeclaredConstructor(); 获取某个构造方法
名.getDeclaredConstructors(); 获取所有构造方法
当构造方法为私有时，需要强制设置为可以访问
名.setAccessible(true);

通过反射机制获取方法并调用：

Class 名 = Class.forName("全类名");

Method method = 名.getMethod("方法名", 类型.class) 获取一个方法；

Method method = 名.getMethods("方法名", 类型.class)；获取多个方法；

Method method = 名.getDeclaredMethod("方法名");

获取私有的方法

Method method = 名.getDeclaredMethods("方法名"); 获取多个私有的方法

在方法为私有的时候注意要强制设置为可以访问
method.setAccessible(true);

调用方法

method.invoke(类名, 参数);

通过反射机制获取属性并使用属性：

类生成对象

Class 名 = Class.forName("全类名");

Field field = 名.getField("属性名"); 获取一个属性

Field field = 名.getFields("属性名"); 获取多个属性

Field field = 名.getDeclaredField("属性名"); 获取一个

私有属性

Field field = 名.getDeclaredField("属性名"); 获取多个

私有属性

field.set(对象, "属性值")

在方法为私有的时候注意要强制设置为可以访问

```
field.setAccessible(true);
```