

Class对象是Reflection的基础，要操作类中的属性和方法，都必须从获取Class对象开始。编译一个新类就会在JVM中及调用类加载器中的方法自动产生一个对应的Class对象。故不能显式调用Class对象。

获取Class对象的方法

1. 用Class的静态方法传入类的全称即可。

```
public static Class<?> forName(String className)
```

```
eg. Class c = Class.forName("java.lang.String");
```

2. 直接获取一个对象的Class。类名.Class

```
eg. Class<?> x = int.class;
```

```
Class<?> x = employee.class
```

3. 通过基本数据类型的包装类获取基本数据类型的Class。

```
eg. Class<?> x = int.class;
```

```
Class<?> classInt = Integer.TYPE;
```

4. 调用对象实例的getClass()方法。对象名.getClass()

```
eg. StringBuilder str = new StringBuilder("123");
```

```
Class<?> klass = str.getClass();
```

5. 获取当前Class的继承类的Class。ps.

```
eg. List list = new ArrayList();
```

```
Class listClass = list.getClass();
```

```
Class superClass = listClass.getSuperclass();
```

Class方法

instanceof 关键字来判断是否为某个类的实例。

1

```
public native boolean isInstance(Object obj);
```