



Java 程序设计

第16章 反射

田英鑫 tyx@hit.edu.cn

中科院计算所培训中心



第16章 反射

本章导读

- 16.1 什么是反射
- 16.2 获取 Class 实例
- 16.3 获取类信息
- 16.4 获取类成员
- 16.5 动态调用方法
- 16.6 使用构造器创建对象





16.1 什么是反射

■ .class文件

- Java在需要用到某个类时才会将其对应的.class文件加载到JVM
- 在JVM中产生java.lang.Class类型的实例代表该class文件，该实例由JVM自动创建
- 任何的.class文件加载后被JVM创建的实例都是Class类型的实例

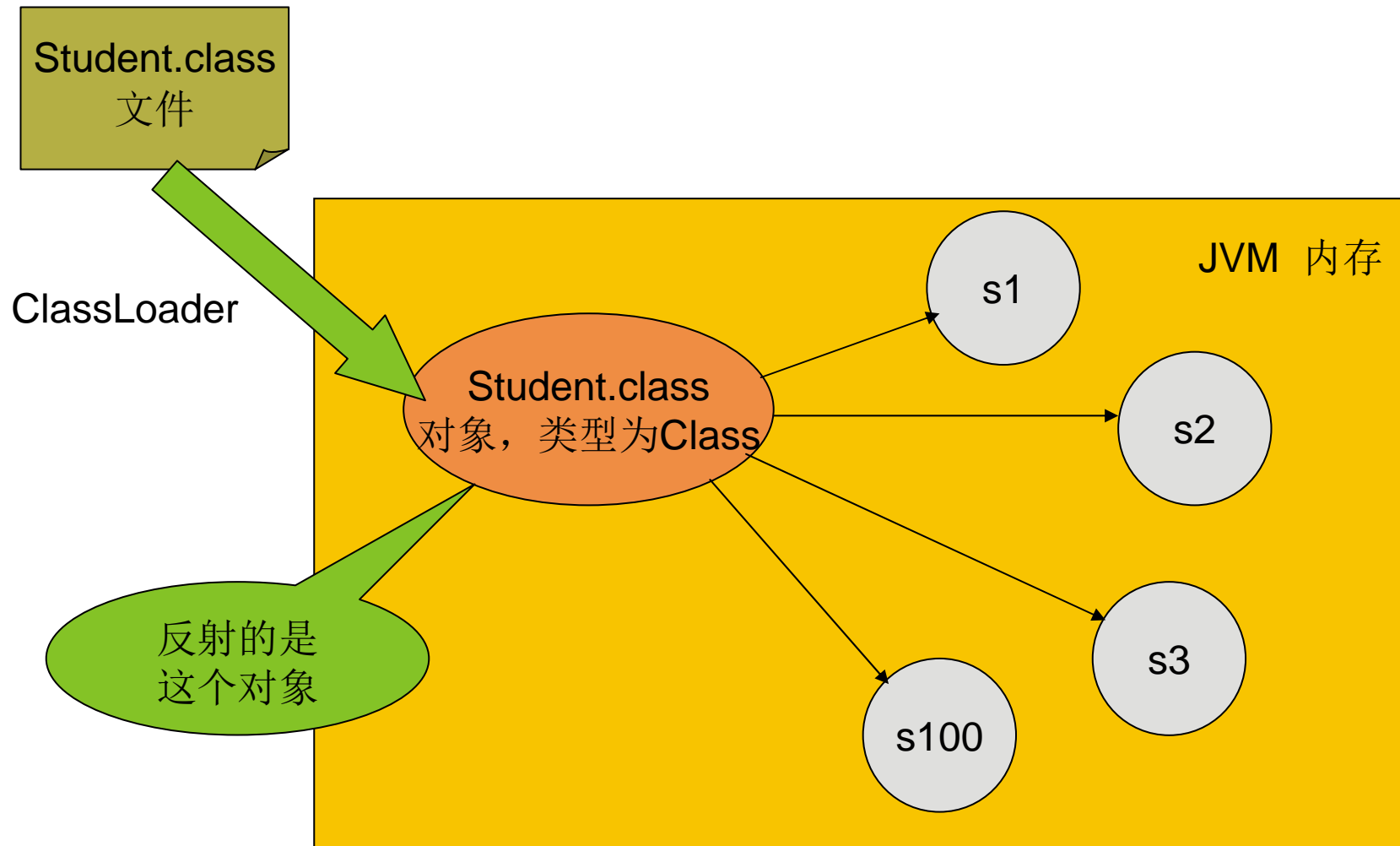
■ 什么是反射

- 运行时动态创建并使用对象的方式称为反射

反射相关类主要放在 `java.lang.reflect` 包中



16.1 什么是反射





16.2 获取 Class 实例

■ 两种方法

- `public final Class getClass()`
- `类型.class`

```
class Student extends Person {}  
class Teacher extends Person {}  
Person p1 = new Student();  
Person p2 = new Student();  
Person p3 = new Teacher();  
p1.getClass() == p2.getClass() //true  
p1.getClass() == p3.getClass() //false  
p1.getClass() == Student.class //true  
p1.getClass().getClass() == p3.getClass().getClass();
```



16.3 获取类信息

■ 通过 Class 实例获取类信息

```
Class<?> clz = Student.class;  
System.out.println("类名:" + clz.getName());  
System.out.println("是否为接口:" + clz.isInterface());  
System.out.println("是否为基本类型:" +  
    clz.isPrimitive());  
System.out.println("是否为数组:" + clz.isArray());  
System.out.println("父类:" +  
    clz.getSuperclass().getName());
```



16.4 获取类成员

■ 成员类

■ Field、Method、Constructor

```
Class<?> clz = Class.forName("java.lang.String");
Method[] methods = clz.getDeclaredMethods();
for(Method method : methods)
{
    System.out.println(method);
}
```



16.5 动态调用方法

■ Method

■ invoke

```
class Student {  
    public String hello(String name) {  
        return "Hello: " + name;  
    }  
}  
  
Class<?> clz = Class.forName("Student");  
Object clzInstance = clz.newInstance();  
Method method = clz.getMethod("hello", String.class);  
Object result = method.invoke(clzInstance, "小明");  
System.out.println(result);
```




16.6 使用构造器创建对象

■ Constructor

■ newInstance

```
class Student {  
    private int id;  
    private String name;  
    public Student(int id,String name) {  
        this.id = id;  this.name = name;  
    }  
}  
  
Class<?> clz = Class.forName("Student");  
Constructor con =  
    clz.getConstructor(int.class,String.class);  
Object result = con.newInstance(1,"小明");
```



Any Question?