1. 反射获取类结构信息

在反射机制的处理过程中不仅仅只是一个实例化对象的操作，更多的情况下还有类的组成结构操作，任何一个类的基本组成结构: 父类（父接口），包，类名，方法，属性；

1. 获取类的基本信息

一个类的基本信息主要包括类所在包的名称，父类的定义，父接口的定义。

范例:定义一个类

|  |  |  |
| --- | --- | --- |
| package com.company.反射机制.interfaceDemo;  public interface IChannlMessage {  public boolean connect();  } | package com.company.反射机制.interfaceDemo;  public interface IMessage {  void send();  } | package com.company.反射机制.abs;  public abstract class AbstractBase {  } |
| **package com.company.反射机制.entity;  import com.company.反射机制.abs.AbstractBase; import com.company.反射机制.interfaceDemo.IChannlMessage; import com.company.反射机制.interfaceDemo.IMessage;  public class Person extends AbstractBase implements IChannlMessage ,IMessage{  public Person() {  System.*out*.println("-----Person构造方法------");  }   @Override  public String toString() {  return "我是一个人！";  }   @Override  public boolean connect() {  return true;  }   @Override  public void send() {  System.*out*.println("发送消息");  } }** | | |

|  |
| --- |
| package com.company.反射机制;  import com.company.反射机制.entity.Person; */\*\*  \* 反射机制获取类结构信息  \** ***@author*** *JackWen  \*/* public class JavaDemo3 {  public static void main(String[] args) {  //获取指定类的Class对象  Class<?> aClass = Person.class;  aClass.  } } |

此时如果想要获取类的一些信息可以通过Class类中的如下方法:

·获取包名称：public Package getPackage();

·获取继承父类：public Class<? super T> getSuperClass();

·获取实现父接口：public Class<?> [] get getnterface();

范例:

|  |
| --- |
| package com.company.反射机制;  import com.company.反射机制.entity.Person; */\*\*  \* 反射机制获取类结构信息  \** ***@author*** *JackWen  \*/* public class JavaDemo3 {  public static void main(String[] args) {  //获取指定类的Class对象  Class<?> aClass = Person.class;  Package aPackage = aClass.getPackage();  Class<?> superclass = aClass.getSuperclass();  Class<?>[] interfaces = aClass.getInterfaces();  System.*out*.println(aPackage.getName());  System.*out*.println(superclass.getName());  for (int i = 0; i <interfaces.length ; i++) {  System.*out*.println(interfaces[i].getName());  }  } } |