定义：

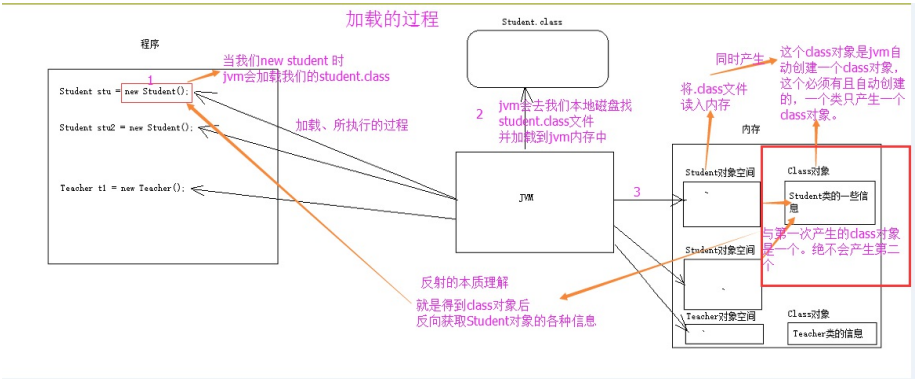
> 在运行状态中，对于任意一个类，都能够知道这个类的所有属性和方法；对于任意一个对象，都能够调用它的任意一个方法和属性；这种动态获取的信息以及动态调用对象的方法的功能称为java语言的反射机制。

- *Class 类十分特殊。它和一般类一样继承自Object，其实体用以表达Java程序运行时的 classes和interfaces，也用来表达enum、array、primitive Java types（boolean, byte, char, short, int, long, float, double）以及关键词void。当一个class被加载，或当加载器（class loader）的defineClass()被JVM调用，JVM 便自动产生一个Class 对象。*
- `Class是Reflection故事起源`：针对任何您想探勘的类，唯有先为它产生一个Class 对象，接下来才能经由后者唤起为数十多个的Reflection APIs。
- Class类中的方法可以获取到的内容如下：
- `构造器`、`方法`、`属性`、`包含的Annotation`、`内部类`、`外部类`、`所实现的接口`、`修饰符`、`所在包`、`类名`、`简称`等；
- 一些判断方法：`注解类？`、`被某个Annotation修饰？`、`匿名类？`、`数组？`、`枚举类？`、`原始类型？`、`接口？`、`是这个类的实例？boolean isInstance(Object obj)`
- 获取Class的方法：

获取 Class 对象的方法	示例
getClass() 注：每个 Class 都有此函数	String str = "abc"; Class c1 = str.getClass();
Class.getSuperClass()	Button b = new Button(); Class c1 = b.getClass(); Class c2 = c1.getSuperClass();
静态方法 Class.forName()	Class c1 = Class.forName("java.lang.String"); Class c2 = Class.forName("java.awt.Button");
.class 属性	Class c1 = String.class; Class c2 = java.awt.Button.class; Class c3 = Main.InnerClass.class; Class c4 = int.class; Class c5 = int[].class;
基本类型包装类的 TYPE 语法	Class c1 = Boolean.TYPE; Class c2 = Byte.TYPE; Class c3 = Character.TYPE; Class c4 = Short.TYPE; Class c5 = Integer.TYPE; Class c6 = Long.TYPE; Class c7 = Float.TYPE; Class c8 = Double.TYPE; Class c9 = Void.TYPE;

博客观后记录

- *class*文件加载过程



- 获取Class对象的三种方式
  - Object# `getClass`();
  - 任何类型都有一个“静态”的 `class` 属性；
  - （常用）通过Class类的静态方法 `forName(String className)`；
- Class常用API

```
/*
 * 通过Class对象可以获取某个类中的：构造方法、成员变量、成员方法；并访问成员；
 *
 * 1.获取构造方法：
 *      1).批量的方法：
 *          public Constructor[] getConstructors(): 所有"公有的"构造方法
 *          public Constructor[] getDeclaredConstructors(): 获取所有的构造方法(包括私有、受保护、默认、公有)
 *
 *      2).获取单个的方法，并调用：
 *          public Constructor getConstructor(Class... parameterTypes):获取单个的"公有的"构造方法：
 *          public Constructor getDeclaredConstructor(Class... parameterTypes):获取"某个构造方法"可以是私有的，或
受保护、默认、公有；
 *
 *          调用构造方法：
 *          Constructor-->newInstance(Object... initargs)
 */
```

- 设置对象属性

```
Class<Student> studentClass =
        (Class<Student>) Class.forName("com.example.demo.multi.springBoot.entity.Student");
Constructor<Student> constructor = studentClass.getDeclaredConstructor(int.class);
System.out.println(Arrays.deepToString(constructor.getGenericParameterTypes()));
Student student = constructor.newInstance( ...initargs: 10);
Field setName = studentClass.getField( name: "name");
setName.set(student, "tom");
System.out.println(student);
```

Student{age=10, name='tom'}

- 反射执行对象的方法

```
/*
 * 获取成员方法并调用:
 *
 * 1.批量的:
 *      public Method[] getMethods():获取所有"公有方法"；（包含了父类的方法也包含Object类）
 *      public Method[] getDeclaredMethods():获取所有的成员方法，包括私有的(不包括继承的)
 * 2.获取单个的:
 *      public Method getMethod(String name,Class<?>... parameterTypes):
 *                  参数:
 *                      name : 方法名;
 *                      Class ... : 形参的Class类型对象
 *      public Method getDeclaredMethod(String name,Class<?>... parameterTypes)
 *
 *   调用方法:
 *      Method --> public Object invoke(Object obj,Object... args):
 *                  参数说明:
 *                  obj : 要调用方法的对象;
 *                  args:调用方式时所传递的实参;
 *
 ):
 */
```

-