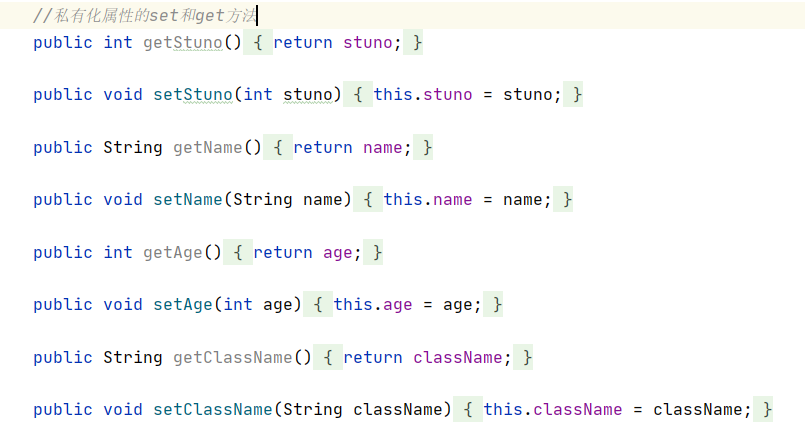
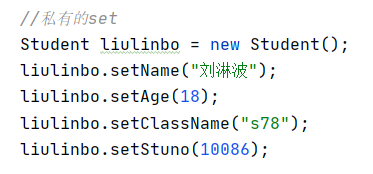
笔记-总结





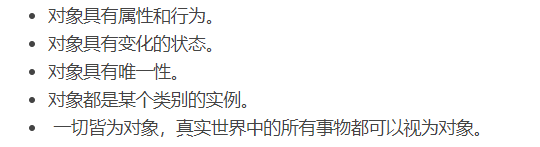
重载

*重载 方法名相同参数不相同的方法  
在同一个类中  
重载是多态的实现之一，重载特性，可以让我们一个方法有多种实现*

重写

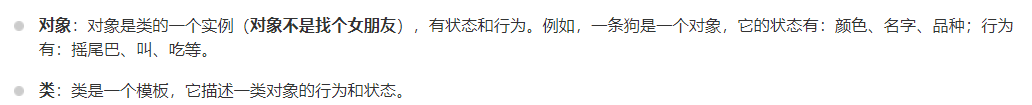
*实际上就是在子类新建一个同名方法覆盖父类的方法、*

类与对象









第二课\_笔记

抽象

在Java中抽象类表示的是一种继承关系，一个类只能继承一个抽象类，而一个类却可以实现多个接口。

1. 抽象类不能被实例化(初学者很容易犯的错)，如果被实例化，就会报错，编译无法通过。只有抽象类的非抽象子类可以创建对象。

2. 抽象类中不一定包含抽象方法，但是有抽象方法的类必定是抽象类。

3. 抽象类中的抽象方法只是声明，不包含方法体，就是不给出方法的具体实现也就是方法的具体功能。

4. 构造方法，类方法（用 static 修饰的方法）不能声明为抽象方法。

5. 抽象类的子类必须给出抽象类中的抽象方法的具体实现，除非该子类也是抽象类。

6.接口里的方法默认都是抽象的

7.接口和抽象类不能直接创建对象

八大数据类型

Int 整形 整数形 32位

Long 长整形 64位

Float 单精度浮点

Double 双精度浮点

Byte 字节 1byte = 8bit

Char 字符

Short 短整形 16位

Boolean 布尔型

引用数据类型 八大基础数据类型的封装类型

String a； 字符串类型

所有类，接口，都是引用数据类型

第三节课--第四节课

**Spring：**轻量级:

1.消耗的资源少(运行内存)

2.框架的侵入程度(自身的代码不需要继承或者实现框架中的接口)

3.独立的实现,不需要依赖其他框架

4.开发的复杂程度(简单)

5.轻量级使用与中小型企业(重量级使用与大型企业,高可用性,高伸缩性)

耦合：a代码必须依赖b代码

解耦：a代码不只是依赖b代码，可以c，d

三大特性： 控制反转( 将创建对象的权限交给spring 容器,由容器统一创建对象,我们使用时,只需要从容器中获取对象即可)

所有容器的父类（ApplicationContext）

相对路径（ClassPathXmlApplicationContext）

基于注解(annotationConfigApplicationContext)

依赖注入(组件之间的依赖关系由容器在应用系统运行期来决定， 也就是由容器动态地将某种依赖关系的目标对 象实例注入到应用系统中的各个关联的组件之中)

面向切面编程()

类的生命周期也是bean的生命周期

java类在容器中都是以一个一个bean存在

过程:*<!-- 定义-> 初始化->使用->销毁-->*

初始化：有两种创建对象的方式：

1.容器加载出配置文件就会把容器创建出来

2.加载配置文件的以后，可以延迟创建，在需要使用的时候在创建

容器初始化 默认 容器加载配置文件时就会把文件创建出来

对象的作用域（scope）

Singlet(单例模式,只有一个实例,容器中每拿一次都是同一个对象，整个容器中只有一个实例对象)

Prototype(原型模式,每拿一次都生成一个不同的对象)

不作作用域的配置,默认都是单例模型、

销毁:关掉所有容器，也就相当与销毁了所有对象

(ClassPathXmlApplicationContext(context)).close()

懒加载：lazyinit

懒加载的方式：初始化容器时，不会创建对象，会在getBean时创建，不配置懒加载时，初始化容器会把所有的对象都创建，有些不需要使用的对象会创建出来浪费内存

依赖注入：给类里面的属性对象设值，

基本数据类型：简单属性的注入

引用数据的注入：对象的注入

集合数据的注入：List，Map等

三种注入方式：

1. 通过构造方法注入
2. Getter，setter 方法注入
3. Ref 引用注入

自动装配的三种方式：

1. 通过构造方法装配(实际上是按构造函数的参数[类型自动装配](http://www.yiibai.com/spring/spring-autowiring-by-type/" \t "https://www.cnblogs.com/luo-mao/p/_blank)。 这意味着，如果一个bean的数据类型与其他bean 的构造器参数的数据类型是相同的，那么将自动装配。)
2. byName装配(**byName会搜索整个配置文件中的bean,如果有相同名称的bean则自动装配，否则显示异常。**)
3. byType装配(**byType会搜索整个配置文件中的bean,如果有相同类型的bean则自动装配，否则显示异常。**)

Ref ：引用方式注入

Psvm

基于注解驱动的配置

Loc容器

在配置文件中要打开一个类

base-package 属性指定扫描范围

@Component 后面加（id） 定义id

@Value(value = "101") 简单属性注

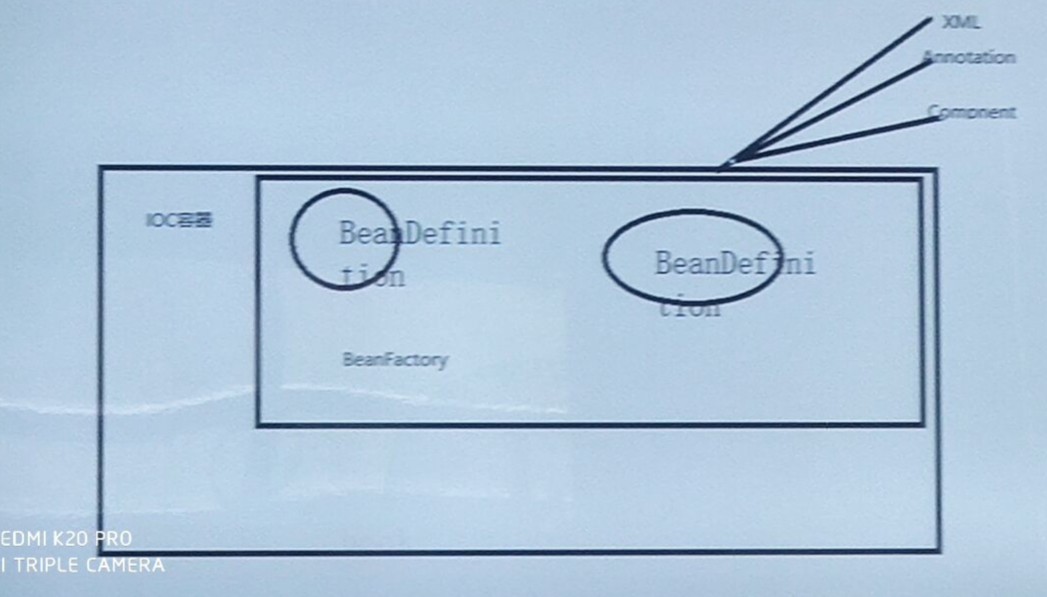
@Configuration *//声明我这个类是一个配置类，在启动的时候sprinng 会将这个类看成是一个\*\*\*.xml的配置*

@Bean *//声明这个方法是一个bean的配置 等同与<bean class="com.Student" id="student">* @Scope("prototype") *//切换为原型模式*

@Value("#{'${A:1,2,3,4}'.split(',')}") //List方式注入

工厂模式BeanFactory 多态的一种实现

Bean在 spring中是以一个一个BeanDefinition类存在的5



生命周期

Spring读取配置文件的机制

配置优于编码的思想