

1. 阶段八(反射注解)

在阶段六的基础上,增加反射注解的相应的练习

1.1. 知识点

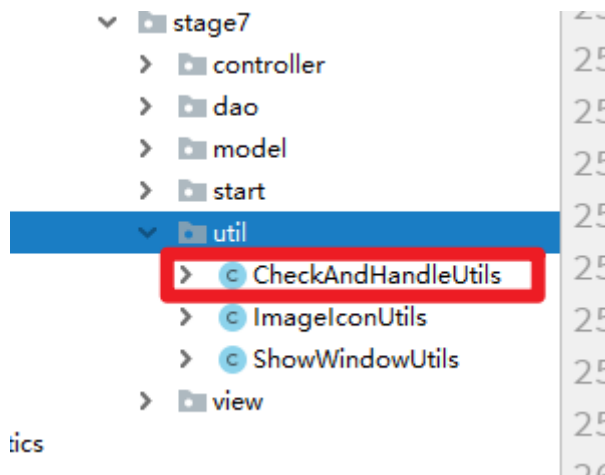
- 反射 注解 相关知识点
 - 自定义注解,注解使用,注解处理
 - 反射获取类中的信息
 - Properties配置文件

1.2. 需求

使用新的文件student3.txt,作为数据源



目前我们输入的数据的校验功能都是在util包下面的CheckAndHandleUtils类中,包括输入的学生的年龄,电话,邮箱等数据



我们找到CheckAndHandleUtils当中的2个方法

- checkName(String name)

- 该方法是对于输入的学生名字的校验,要求是名字的长度在2-5之间,根据输入的内容进行判断,并返回0,1,2这三种状态.注释中有写.
- checkAge(String age)
 - 该方法是对于年龄的校验,要求是年龄在1-150之间,并根据输入的内容进行返回相应的3种状态.

```

186 private static int checkName(String name) {
187     if (name == null || "".equals(name)) {
188         return 1;
189     }
190     if (name.length() > 5 || name.length() < 2) {
191         return 2;
192     }
193     // 正常
194     return 0;
195 }
196
197 /**
198  * 校验用户输入age, 返回一个int状态值:
199  * 0: 一切正常
200  * 1: 输入为空
201  * 2: 非数字
202  * 3: 没有在1-150之间
203  * @since 9:49
204  * @param age 被校验的年龄
205  * @return int
206  * @author wuguidong@cskaoyan.onaliyun.com
207  */
208 private static int checkAge(String age) {
209     if (age == null || "".equals(age)) {
210         return 1;
211     }
212     int ageNum;
213     try {
214         ageNum = Integer.parseInt(age);
215     } catch (NumberFormatException e1) {
216         // 非数字
217         return 2;
218     }
219     if (ageNum < 1 || ageNum > 150) {
220         return 3;
221     }
222     // 正常
223     return 0;
224 }

```

对于这2个方法,我们做出如下改进

- 对于checkName(String name)方法我们使用注解反射配置文件的方式去修改.

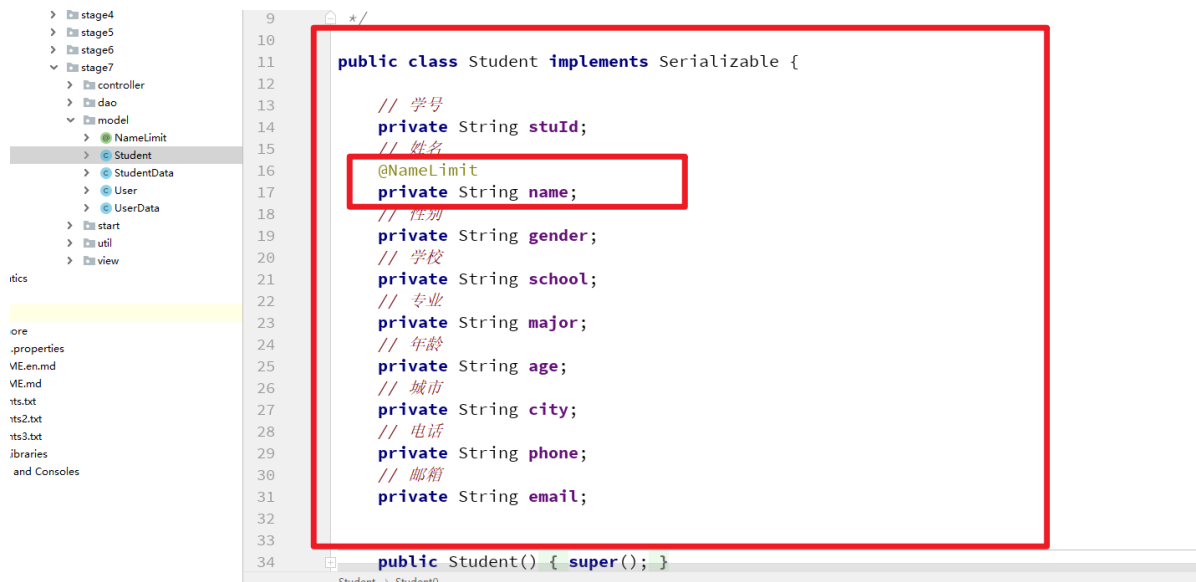
model中新增注解NameLimit, 注意通过元注解设置注解的保留级别,注解中有2个属性,minLength最小名字长度默认值为2, maxLength最大年龄长度为5.

```

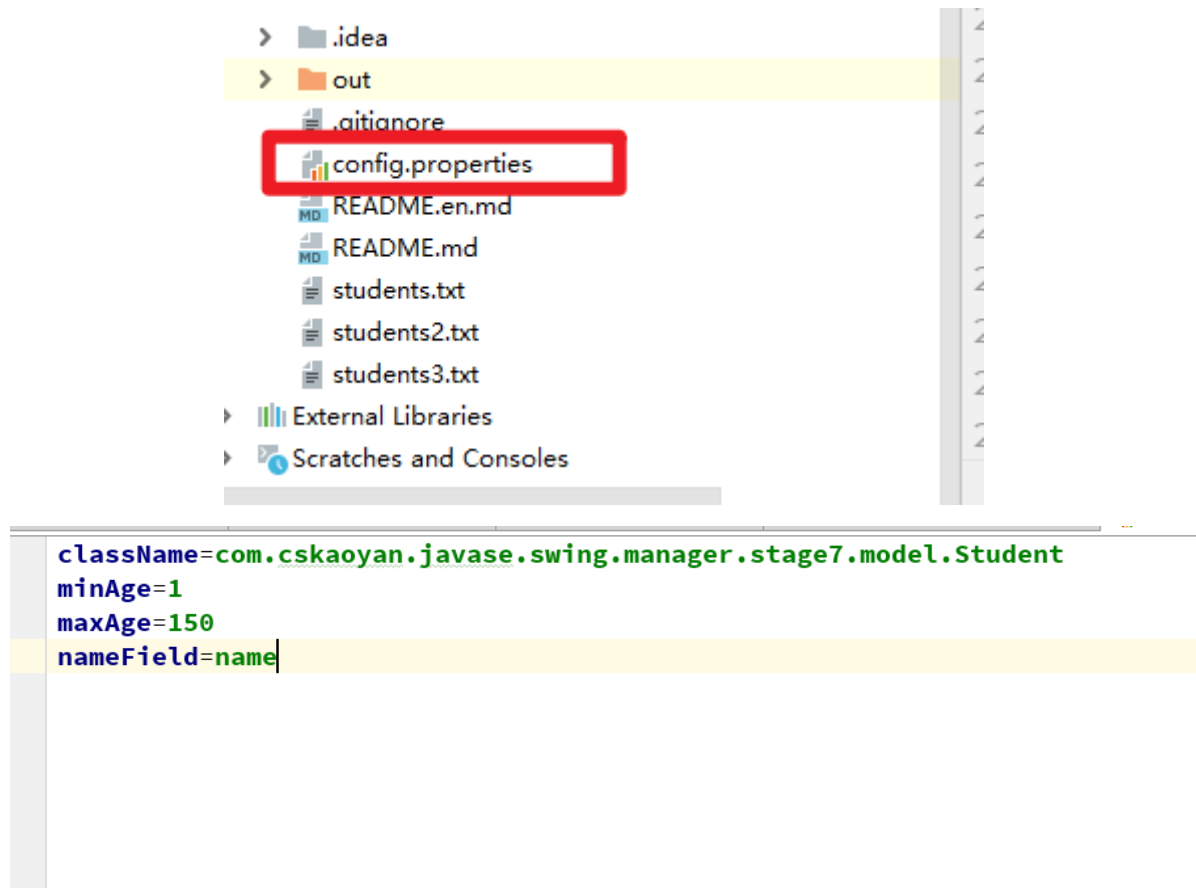
1 package com.cskaoyan.javase.swing.manager.stage7.model;
2
3 import java.lang.annotation.Retention;
4 import java.lang.annotation.RetentionPolicy;
5
6 /**
7  * 该注解 进行名字长度的限制 默认名字长度为2-5之间
8  */
9 @Retention(RetentionPolicy.RUNTIME)
10 public @interface NameLimit {
11     // 最小长度
12     int minLength() default 2;
13
14     // 最大长度
15     int maxLength() default 5;
16 }

```

并用此注解加在Student类中的name的成员变量上



项目中新增properties文件,文件中填写一些配置信息.



className:为全类名

minAge: 年龄的最小限制为1岁

maxAge: 年龄的最大限制为150岁

nameField: 学生类中的名字的成员变量

checkName方法中,名字长度的值2,5是写死的,现在我们使用注解修饰了Student类中的name这个成员变量,2个默认值分别是2,5,我们的目标现在把之前写死的数值更换成为注解中的属性值.

- 对于checkAge(String age)这个方法,是对于年龄的校验,我们通过从配置文件中读取最大年龄maxAge,最小年龄minAge去做比较

从properties文件中读取这2个值,然后跟checkName(String age)当中传入的age进行比较.

1.3. 思路

- checkName(String name)
 - 对于该方法,我们的最终目的是把之前写死的值2,5替换成NameLimit注解中的2个属性值.
 - 我们想要获取这2个属性值,就要获取NameLimit的注解实例
 - 想要获取注解实例,先搞清它加在哪里,是加在Student中的name的成员变量上,所以要先获取Student类的成员变量name
 - 想要获取name成员变量,就需要先获取Student的字节码文件对象
 - 想要获取字节码文件对象,就需要知道该全类名,这个全类名是在properties文件中,所以要读取配置文件,再通过Class.forName(全类名)的方式得到Student中的字节码文件.

我们倒过来去思考就发现逻辑很清晰了,所以整理一下逻辑

1. 读取配置文件中的className的值,拿到全类名
2. 通过全类名,获取Student的字节码文件对象
3. 通过字节码文件对象拿到成员变量name
4. 我们判断一下name成员变量上是否使用了注解
5. 使用了注解,我们就获取NameLimit的注解实例
6. 通过注解实例拿到2个属性值,minLength , maxLength
7. 通过2个属性值跟checkName(String name)的参数name的长度做比较,返回相应的状态码0,1,2

- checkAge(String age)
 - 对于该方法,我们只需要去读取配置文件,通过getProperty(String key)方法拿到minAge,maxAge这2个值
 - 把这2个值跟age去比较,是否在这个年龄区间,并且返回相应的状态码0,1,2,3