## 1 测试说明

### 测试方式

组间测试在与被测试组协调之后，采取了观看被测试组的测试代码，来查看被测试的结果是否满足测试需求规格说明书中的内容。

### 测试说明

#### IOC模块单元测试

该部分对于IOC模块的进行了测试，下面是说明：

1. 本文档记录IOC模块单元测试的测试内容和测试结果。在B组的软件测试说明书\_v1.2.docx版本中所对应的测试用例编号为Blade\_IOC\_TC1。
2. 本文档记录配置模块的单元测试的测试内容和测试结果。在B组的软件测试说明书\_v1.2docx版本中所对应的测试用例编号为Blade\_配置\_TC1。

#### 1.2.2 单元和配置测试

该部分对于部分的单元和部分配置进行了测试，下面是测试方法的说明：

1. createUniqueFileName()： 给出文件名称，根据所给的文件名创建唯一的文件
2. createEndSuffixFileName()： 给出文件名，同时给出想要在文件名与后缀之间插入的后缀名
3. isNotBlank()： 判断所给的字符是否是不为空的，返回布尔型
4. isBlank()： 判断所给的字符是否为空，返回布尔型
5. trim()： 去除字符前后的空格
6. addScanPackage()： 添加扫描包，同时可以使用getPackages()来获取所添加的包名
7. setDev()： 设置开发者模式，true为开启，false为关闭，使用isDev()可以查看设置值
8. setBasePackage()： 设置基本配置包测试，同时可以使用getPackages()来获取所设置的包名

## 2 测试内容

### IOC模块测试

1. 针对IOC模块的测试，具体测试内容包括一下三点。

a.扫描注解后初始化的对象与预期相同。

b.空路径提示错误。

c.IOC容器能够向指定的对象中注入依赖的值。

2. 针对配置模块的测试，测试内容为对程序所使用到的资源进行配置，测试是否能够正确获取到资源。

### 2.2 单元和配置测试

该部分测试了B组Blade项目中的单元测试，测试的对象和测试项如下表格：

|  |  |
| --- | --- |
| 测试对象 | 测试项 |
| createUniqueFileName() | 同文件测试 |
| createEndSuffixFileName() | 有后缀的不同文件测试 |
| createEndSuffixFileName() | 空文件测试 |
| createEndSuffixFileName() | 文件名无后缀测试 |
| isNotBlank() | 字符是否不为空测试 |
| isBlank() | 字符是否为空测试 |
| trim() | 去除空格测试 |
| addScanPackage() | 添加扫描包测试 |
| setDev() | 设置开发者模式测试 |
| setBasePackage() | 设置基本配置包测试 |

## 测试结果

### IOC模块测试

#### 2.1.1 IOC模块

1. 扫描注解后初始化对象与预期相同。

测试前后对象行为一致，扫描注解后的初始化对象与预期相同。

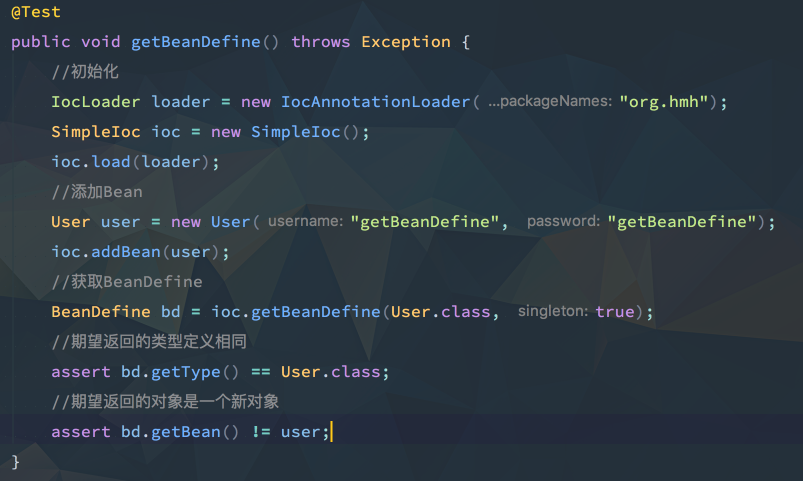


图1（测试生成BeanDefine）

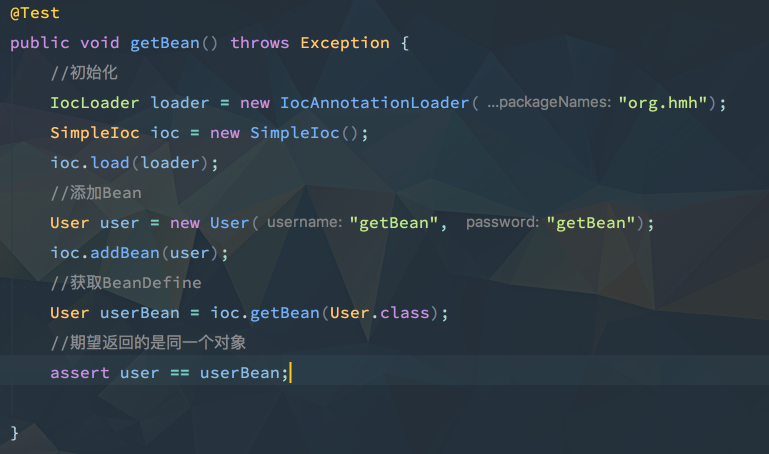


图2（测试放入前后是同一个对象）



图3（清理前后对比）

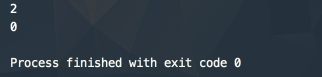


图4（清理前后对比输出结果）

1. 空路径提示错误。

当输入空路径时会提示错误，抛出异常。



图5

1. IOC容器能够向指定的对象中注入依赖的值。

IOC容器成功向指定容器中注入了依赖值。



图6

#### 3.2 IOC配置模块

1. 通过使用预先设定的配置项，程序能够正确加载和使用配置项配置的资源。



图7



图8

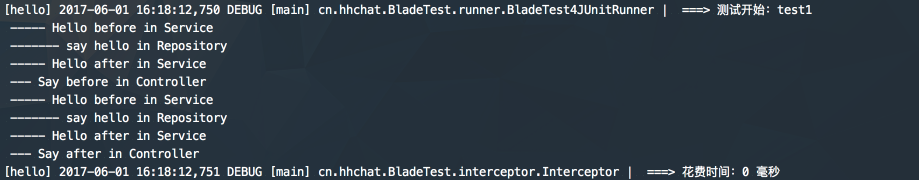


图9

### 单元和配置测试

#### 2.2.1 单元测试

1. 同文件测试

输入相同的文件名，查看是否检查文件名是否相同，输出一个唯一的文件名，结果该方法不对相同文件名进行检查，同时输出两个唯一的文件名，代码以及结果截图如下：

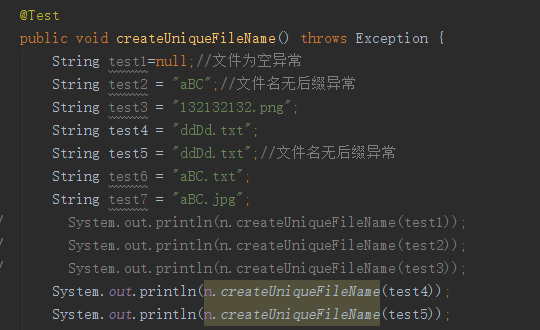




图1 同文件测试

1. 不同文件后缀名测试

输入不同后缀的文件名，查看正常情况下是否会输出不唯一的文件名，结果给方法输出了唯一的文件名，代码以及结果截图如下：



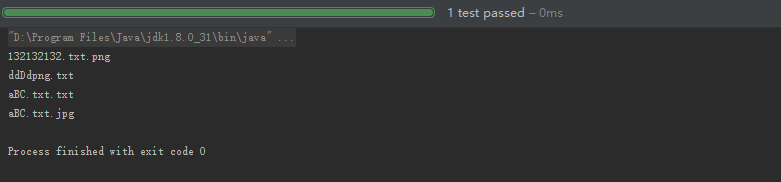
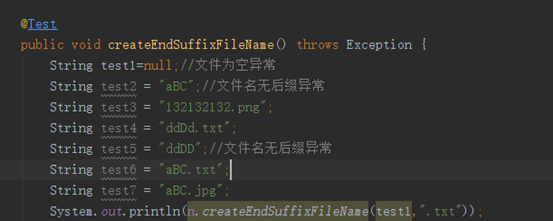


图2 不同文件后缀名测试

1. 空文件测试

输入空文件名，测试该方法是否对空文件进行了检测，结果不检查输入是否为空文件名，代码和结果截图如下：



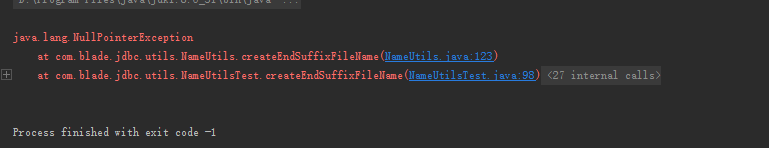
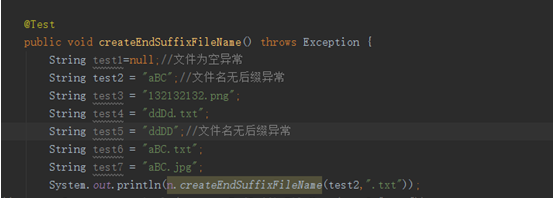


图3 空文件测试

1. 文件名无后缀测试

输入无后缀名的文件，测试该方法是否能对于无后缀文件进行唯一文件名的创建，结果输出字符串索引越界异常，代码和结果截图如下：



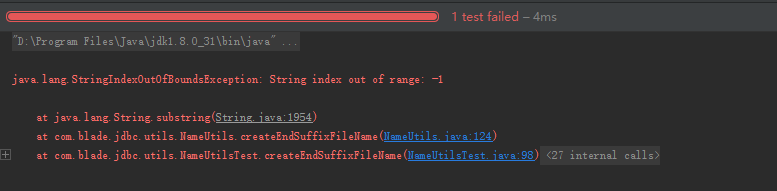
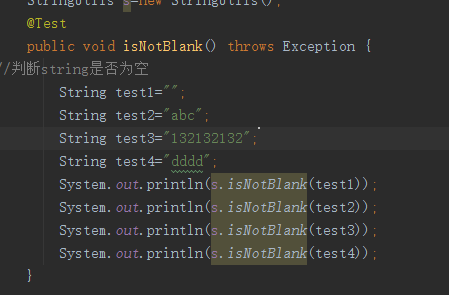


图4 文件名无后缀测试

1. 字符不为空测试

输入空字符和非空字符，对该方法进行测试，结果与预期一样，代码和结果截图如下：



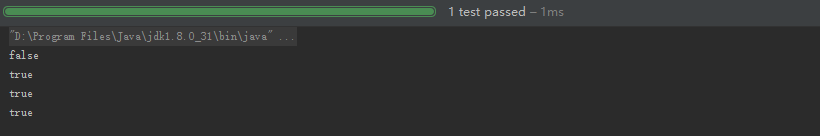
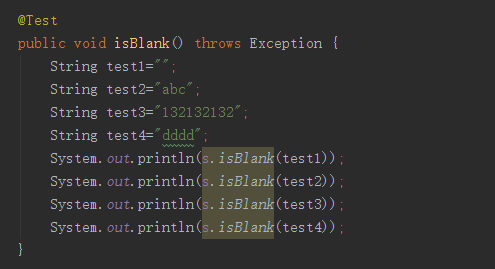


图5 字符不为空测试

1. 字符为空测试

与字符不为空测试类似，输入空字符和非空字符进行测试，结果与预期一致，如下截图：



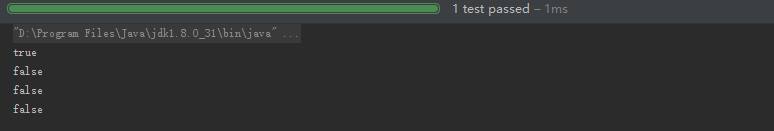
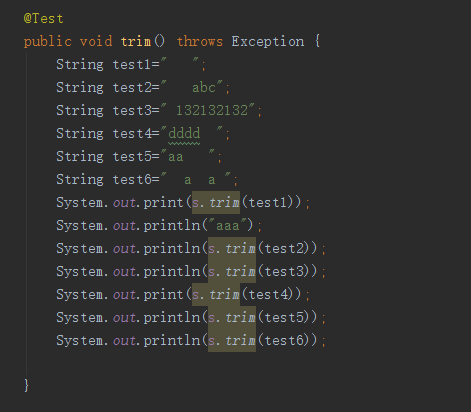


图6 字符为空测试

1. 去除空格测试

根据该方法的说明，田间一些测试字符，查看是否符合预期，结果与预期一致，如下截图：



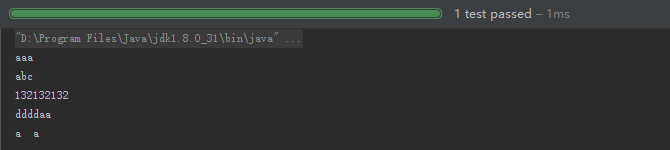


图7 去除空格测试

#### 2.2.2 配置测试

1. 添加扫描包测试

通过使用getPackages()来查看添加的扫描包是否正确添加，如下截图：

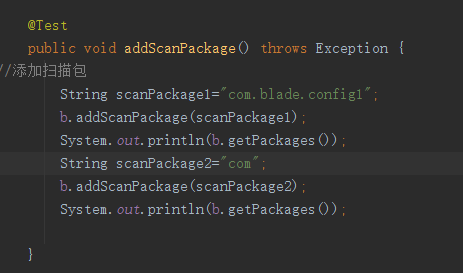




图8 添加扫描包测试

1. 设置开发者模式测试

通过给setDev()方法添加true（false）值来打开（关闭）开发者模式，同时使用isDev()方法来检测当前模式是否为开发者，如下截图：

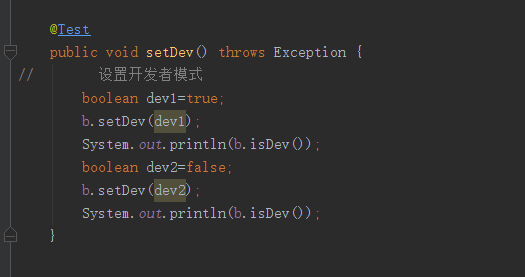
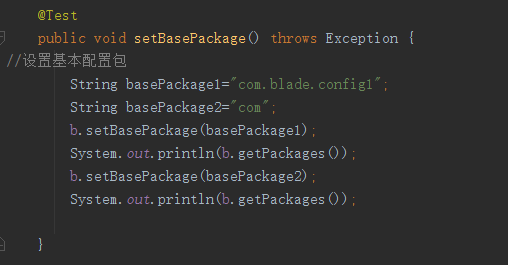




图9 开发者模式测试

1. 设置基本配置包测试

与添加扫描包测试类似，通过使用getPackages()方法可以查看是否正确设置了基本配置包，如下截图：



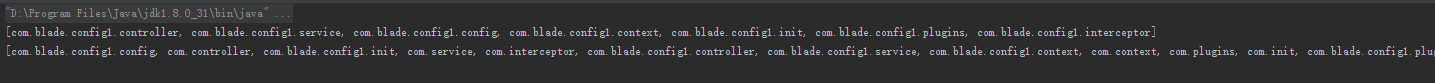


图10 基本配置包测试

## 4 结果分析

首先，由于不太熟悉Blade框架，与B组进行了协商，通过观察B组的测试代码来进行该项目的测试。

其次，在测试过程中，可能由于不能提出全面的测试用例，从而导致测试不太全面，有些问题没有测试出来。

根据此次的测试，也有一些可以看出来的问题存在：

1. 对于createUniqueFileName()的测试中，没有后缀名的文件在现实中是存在的，在查看源代码后发现，该方法根据查找最后一个“.”来进行分割，从而生成一个唯一的文件名
2. 对于createUniqueFileName()的测试中，没有对空文件名进行try…catch的异常检测