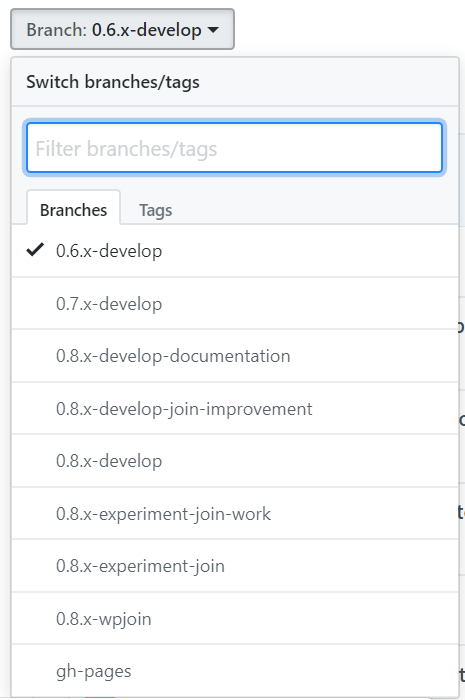
**JCUnit**

1. **发布时间：2013**

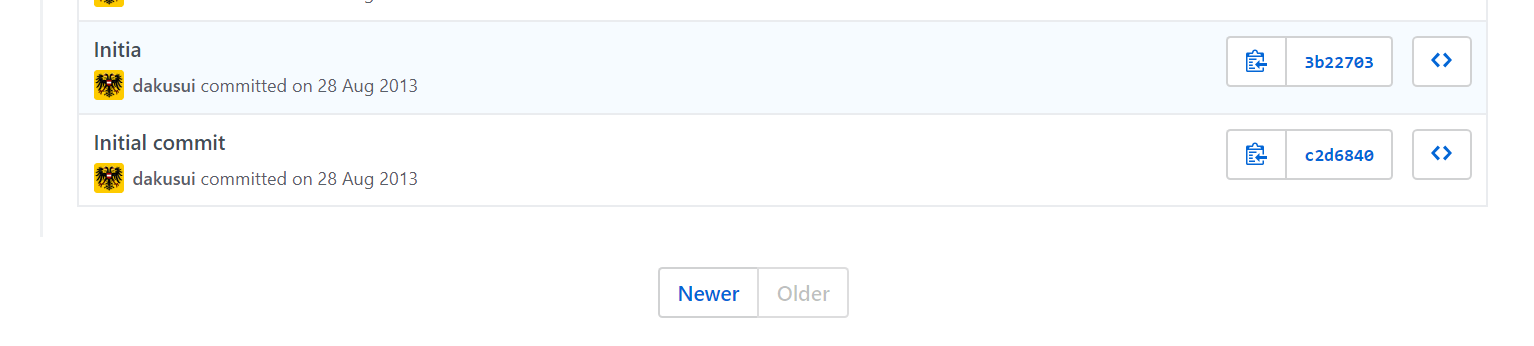
最早的release版本



github上该工具有多个分支



其中0.6.x-develop分支的最早commit时间也是2013年



1. **工具使用形式和跨平台特性**

Library

Cross (发布maven包)

1. **测试生成及其输入输出格式**
   1. **生成算法：多种方法，0.7版本中提到支持AETG方法，0.8版本支持IPOG方法。**

JCUnit employs CIT technique by supporting different covering array engines that can convert a set of factors and constraints into a set of test cases. IpoGcCoveringArrayEngine is JCUnit’s default engine based on the IPOG algorithm

但是在JUNuit8试用中没有发现配置算法的选项。

* 1. **支持维度：**1-n

public static class Config extends ConfigFactory.Base {

@Override

protected Requirement defineRequirement(Requirement.Builder defaultValues) {

return defaultValues.withStrength(1).build();

}

}

定义配置类中withStrength(t)中参数t

* 1. **可变力度：**No
  2. **种子：**Yes addSeed配置种子。通过使用发现，种子必须声明所有参数的取值。

public static class Config extends ConfigFactory.Base {

@Override

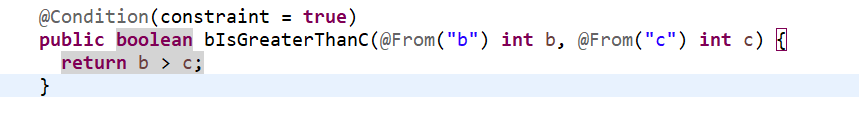
protected Requirement defineRequirement(Requirement.Builder defaultValues) {

return defaultValues.addSeed(Tuple.builder().put("a",0).put("b",2).put("c",2).build()).withStrength(1).build();

}

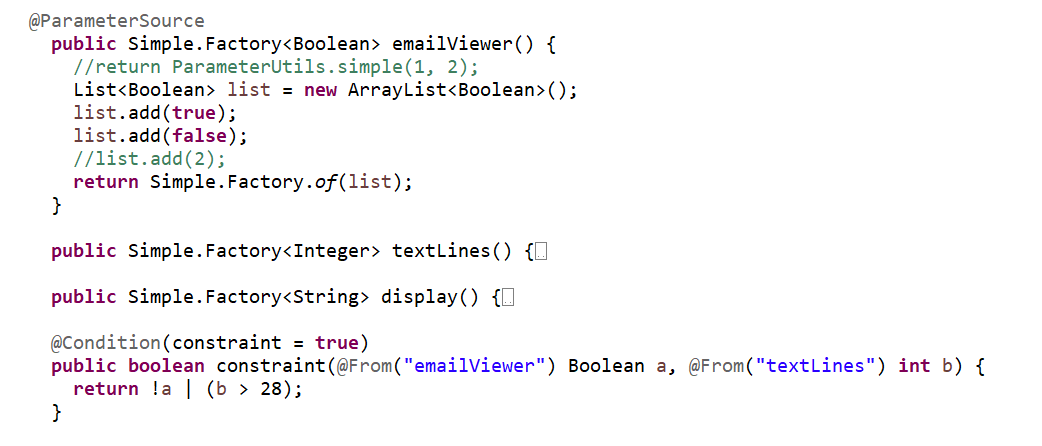
}

* 1. **约束：**Yes，实现一个函数



**约束检查：**No, 加入矛盾约束，点击生成界面无反应

* 1. **建模语言：**源码走声明模型



* 1. **输出格式（包括抽象测试用例、自然语言描述测试计划、代码片段等）：**

Junit执行测试用例，IDE标准输出，可通过以下标注定义额外输出描述

@{Before | After}Class

This annotation lets JCUnit know the annotated static method should be run {before all tests defined by enclosing class are run | after all tests defined by enclosing class are finished}.

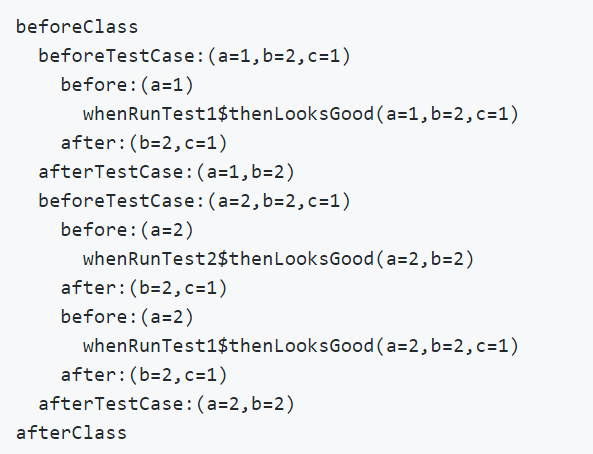
@{Before | After}Class

This annotation lets JCUnit know the annotated static method should be run {before each test case defined by enclosing class are run | after each test case defined by enclosing class are finished}.

@{Before | After}

This annotation lets JCUnit know the annotated method should be run {before each test method defined in enclosing class is run | after each test method defined in enclosing class is finished}.

下图是网站给出的例子，可自定义输出内容。



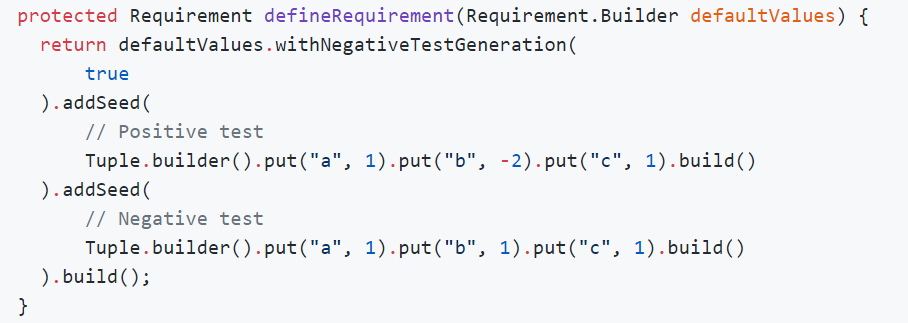
1. **其它测试活动**
   1. **测试用例优化：**No
   2. **测试执行：**Yes

集成Junit，执行测试

* 1. **覆盖率评估：**Yes，0.6版本支持计算有限状态机的状态覆盖、路径覆盖和状态转换覆盖，以及组合覆盖情况。
  2. **故障定位：**No

1. **额外功能** 
   1. **负面测试**

defaultValues.withNegativeTestGeneration(true)



以种子的方式手动添加负面测试用例

1. **和其它测试工具的兼容性：**JUnit
2. **工具演化：**

从2013到2019总计82条release记录

0.8分支从2017到2019总计137条commit记录

0.7分支从2013到2017总计776条commit记录

0.6分支从2013到2016总计659条commit记录

我试用的是JCUnit8版本，该版本和旧版本不兼容，标注语法不一样。且旧版本支持的功能我在新版本中无法使用 (旧版本的例子中标注JCUnit8不识别)