Junit：一款免费开源的测试套件。用于验证程序的运行结果是否符合预期。本文档仅仅记录在maven工程下使用Junit的要点。Maven的pom.xml文件新增依赖：

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.12</version>

<scope>test</scope>

</dependency>

## @Before测试前准备

写在方法上，测试前准备，被@Before注释的方法将会在@Test注释的方法之前运行，一般做准备基础数据，实验数据等工作。

## @Test测试

写在方法上，测试代码写在被@Test注释的方法中，测试类中的测试方法可以有多个，junit会全部执行，但不会按照顺序执行，若需要按要求执行，可参考@FixMethodOrder(后面讲)。方法中编写测试代码会用到Assert类

### Assert

|  |  |
| --- | --- |
| ~.assertArrayEquals(expecteds,actuals); | 比较期望值数组与实际值数组是否相等 |
| ~.assertEquals(expected,actual); | 比较期望值与实际值是否相等 |
| ~.assertNotEquals(expected,actual); | 比较期望值与实际值是否不相等 |
| ~.assertTrue(condition); | 比较实际值是否为true，为true则通过测试 |
| ~.assertFalse(condition); | 比较实际值是否为false，为false则通过测试 |
| ~.assertNull(object); | 比较对象是否为null |
| ~.assertNotNull(object); | 比较对象是否不为null |
| ~.assertSame(expected,actual); | 比较对象是否一样 |
| ~.assertNotSame(unexpected,actual); | 比较对象是否不一样 |

@Test方法支持时间验证——程序在指定时间内跑完算通过。例子如下（根据现在PC的速度会报错。）

@Test(timeout = 10)

public void \_2() {

Double dou = 0.0;

for (int i = 0; i < 1000000; i++) {

dou += RandomUtils.nextDouble(0,100);

}

}

若希望@Test方法时产生一个Exception异常表示测试通过，使用@Test的expected属性。

@Test(timeout = 10000,expected = NullPointerException.class)

public void \_2() {

Double i = null;

Double j = 10.2;

Double k = i+j;

}

## @FixMethoder(MethodSorters.JVM)

写在测试类上，

MethodSorters.JVM：按jvm生成的hash值执行@Test

MethodSorters.NAME\_ASCENDING：按方法名字顺序执行@Test

MethodSorters.DEFAULT：默认方法，不可预期执行顺序

## @Orde@Ignore跳过@Test方法

写在方法上，被@Test注释的方法上加上@Ignor，junit执行时将跳过执行此方法。

## @After测试后收尾

写在方法上，测试后收尾，被@After注释的方法将会在@Test注释的方法之后运行，一般做：关闭数据连接，清理日志，缓存等工作