

「测试」 - 单元测试 & JUnit 4忽略异常执行



isisiwish

关注他

1 人赞同了该文章

按照正常的单元测试要求，一个case应该只有一个断言，不会出现多个断言在一个case中的场景，但是有时为了效率，希望多个断言都在case中执行，但是JUnit 4触发断言失败后，其他断言不会执行。

```
@Test
public void testIEEE754()
{
    double addRs = 0.05 + 0.01;
    double subRs = 1.0 - 0.42;
    double mulRs = 4.015 * 100;
    double divRs = 123.3 / 100;

    log.info("0.05 + 0.01 = {}", addRs);
    log.info("1.0 - 0.42 = {}", subRs);
    log.info("4.015 * 100 = {}", mulRs);
    log.info("123.3 / 100 = {}", divRs);

    Assert.assertEquals(0.06, addRs);
    Assert.assertEquals(1.42, subRs);
    Assert.assertEquals(401.5, mulRs);
    Assert.assertEquals(1.233, divRs);
}
```

一、AssertJ软断言。

```
@Test
public void testIEEE754()
{
    double addRs = 0.05 + 0.01;
    double subRs = 1.0 - 0.42;
    double mulRs = 4.015 * 100;
    double divRs = 123.3 / 100;
    log.info("0.05 + 0.01 = {}", addRs);
    log.info("1.0 - 0.42 = {}", subRs);
    log.info("4.015 * 100 = {}", mulRs);
    log.info("123.3 / 100 = {}", divRs);

    SoftAssertions softAssert = new SoftAssertions();

    softAssert.assertThat(addRs).isEqualTo(0.06);
    softAssert.assertThat(subRs).isEqualTo(1.42);
    softAssert.assertThat(mulRs).isEqualTo(401.5);
    softAssert.assertThat(divRs).isEqualTo(1.233);
    softAssert.assertAll();
}
```

软断言允许执行下一步而不抛出异常，在末尾assertAll()方法中抛出所有收集错误。

```
org.assertj.core.api.SoftAssertionError:
The following 4 assertions failed:
1) expected:<0.06[]> but was:<0.06[0000000000000005]~
at BigDecimalUtilTest.testIEEE754(BigDecimalUtilTes
```

赞同 1

添加评论

分享

喜欢

收藏



```
2) expected:<[1.42]> but was:<[0.5800000000000001]>
at BigDecimalUtilTest.testIEEE754(BigDecimalUtilTest.java:33) expected:<[1.42]> but wa
3) expected:<401.[5]> but was:<401.[4999999999999994]>
at BigDecimalUtilTest.testIEEE754(BigDecimalUtilTest.java:34) expected:<401.[5]> but w
4) expected:<1.23[3]> but was:<1.23[2999999999999999]>
at BigDecimalUtilTest.testIEEE754(BigDecimalUtilTest.java:35) expected:<1.23[3]> but w
```

二、ErrorCollector

```
@Rule
public ErrorCollector collector = new ErrorCollector();

@Test
public void testIEEE754A()
{
    double addRs = 0.05 + 0.01;
    double subRs = 1.0 - 0.42;
    double mulRs = 4.015 * 100;
    double divRs = 123.3 / 100;

    collector.checkThat(addRs, IsEqual.equalTo(0.06));
    collector.checkThat(subRs, IsEqual.equalTo(1.42));
    collector.checkThat(mulRs, IsEqual.equalTo(401.5));
    collector.checkThat(divRs, IsEqual.equalTo(1.233));

    try
    {
        throw new RuntimeException("Exception");
    }
    catch (Exception ex)
    {
        collector.addError(ex);
    }
}
```

通过JUnit内置的ErrorCollector TestRule，也能够达到要求。

编辑于 2019-06-10

[软件测试](#)

推荐阅读



Junit5 + YAML 轻松实现参数化和数据驱动，让 App 自动...

测吧

发表于软件测试开...

一文教会你如何在Spring中进行集成测试，太赞了

前言：不得不说，测试真的是太重要了！但并不是所有的开发者都这样认为，这种感觉在我回到洛阳后尤其强烈。竟然有团队成员不经测试就把代码提交到代码库，并且是会报错的那种，我天呐，遇到...

Java汤...

发表于Java高...



使用jav
归测试

宜信技术

▲ 赞同 1 ▼ ● 添加评论 ➦ 分享 ♥ 喜欢 ★ 收藏



还没有评论

评论由作者筛选后显示

