

实验课程名称： 软件工程基础实验

| | | | | | |
|--------|------|------|---------|------|-----------|
| 实验项目名称 | 单元测试 | | | 实验成绩 | |
| 实 验 者 | 贺秋程 | 专业班级 | 软件 1804 | 组 别 | |
| 同 组 者 | 罗 治 | | | 实验日期 | 2020.5.21 |

第一部分：实验预习报告（包括实验目的、意义，实验基本原理与方法，主要仪器设备及耗材，实验方案与技术路线等）

一、实验目的

- 1) 掌握单元测试的方法；
- 2) 学习 JUnit 测试原理及框架；
- 3) 掌握在 Eclipse 环境中加载 JUnit 及 JUnit 测试方法和过程。

二、问题描述

单元测试：

Eclipse 集成了 JUnit，可以非常方便地编写 Test Case。Eclipse 自带了一个 JUnit 插件，不用安装就可以在项目中测试相关的类，并且可以调试测试用例和被测类。

对一次实验的生命游戏进行单元测试，来验证功能是否正常。

第二部分：实验过程记录（可加页）（包括实验原始数据记录，实验现象记录，实验过程发现的问题等）

三、算法设计思路

对生命游戏中的 Cell 类进行单元测试。

1) 将 JUnit4 单元测试包引入这个项目。

2) 新建单元测试类 CellTest，声明一个 Cell 对象来进行测试。

3) 设置前置条件为更新换代，并对 `getNowGeneration()`，`randomCell()`，`deleteAllCell()`，`update()`，`getNeighborCount()`这 5 个方法进行单元测试。

4) 对于 `getNowGeneration()` 方法，对 cell 进行更新换代后，测试 cell 的代数是否和预测的一样。

5) 对于 `randomCell()` 方法，随机生成的初代细胞分布，只要不是全部都为死的状态则代表成功，对生成后的数组进行检索，只要出现了获得细胞则赋值一，代表成功

```
int flag0=0;
for (int i=0;i<25;i++)
    for (int j=0;j<25;j++)
    {
        if (testgrid0[i][j]==1)
            flag0=1;
    }
Assert.assertEquals(1,flag0);
```

6) 对于 `deleteAllCell()` 方法来说，与 `randomCell` 方法 有着异曲同工之妙，同样，删除所有细胞后，即将所有细胞置为死后对数组进行检索，只有没有 1，则代表全员死亡。

7) 对于 `update()`方法来说，设置一个初始状态后，进行一代更新，与预期的存活数来进行比较，如果与预期的细胞数是一样的话，则测试成功。

```
@org.junit.Test
public void update() {
    int[][] newGrid0=new int[30][30];
    for (int i=1;i<4;i++)
        for (int j=1;j<4;j++)
        {
            newGrid0[i][j]=1;
        }

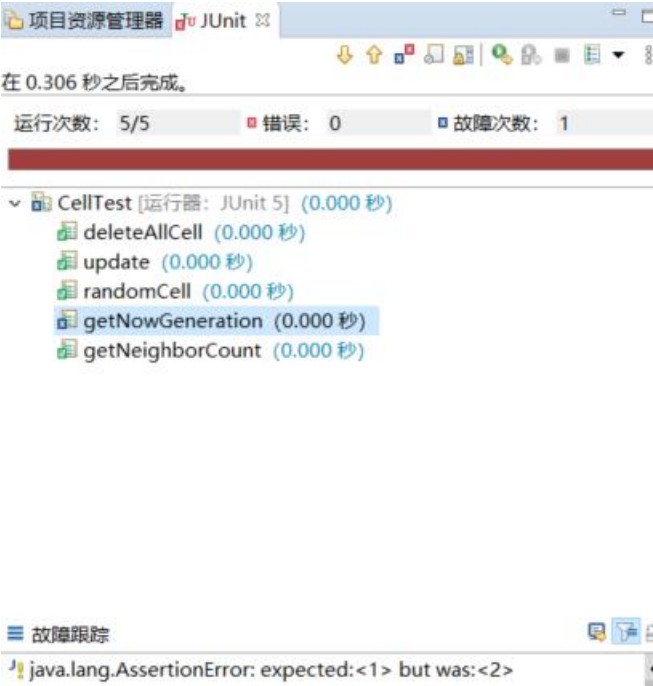
    cell.setGrid(newGrid0);
    int num=0;
    cell.update();
    // Assert.assertEquals(1,newGrid0[1]
    //int[][] newGrid1=cell.getGrid();

    for (int i=0;i<25;i++)
        for (int j=0;j<25;j++)
        {
            if (newGrid0[i][j]==1)
                num++;
        }
    Assert.assertEquals(6,num);
}
```

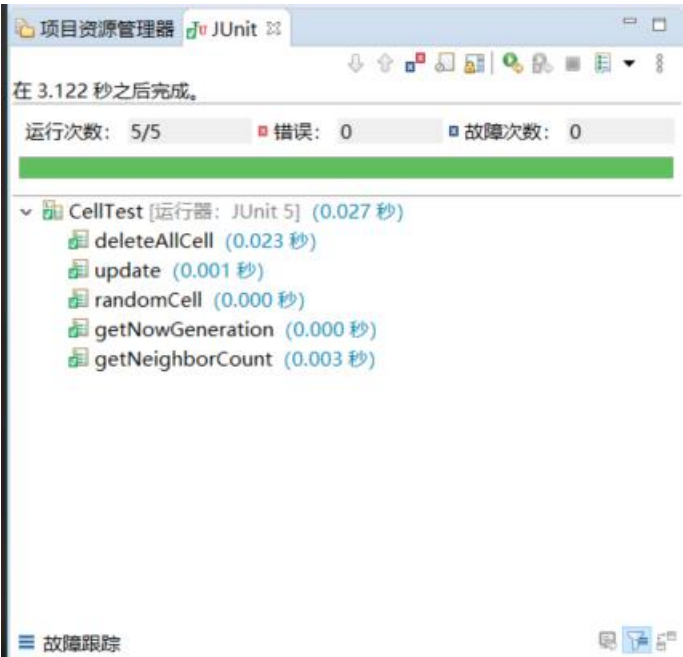
8) 对于 `getNeighborCount()`方法，设置一个状态，然后指定一个细胞返回它的四周的活得细胞数，之后与预期来进行比较后，如果一样则是测试成功。

第三部分 结果与讨论（可加页）

一、实验结果分析（包括数据处理、实验现象分析、影响因素讨论、综合分析和结论等）



代数出现错误，将代数的测试函数进行一下修改，修改后结果如下：



| 时间 | “驾驶员” | “领航员” | 具体分工 |
|--------------------------|-------|-------|---|
| 2020.05.21 20: 30-22: 30 | 罗治 | 贺秋程 | 罗治负责 <code>getNowGeneration()</code> , <code>randomCell()</code> 方法实现, 贺秋程查找相关算法 |
| 2020.05.23 14: 30-16: 30 | 贺秋程 | 罗治 | 贺秋程负责 <code>deleteAllCell()</code> , <code>update()</code> 方法实现, 罗治查找相关函数 |
| 2020.05.24 18: 30-20: 30 | 罗治 | 贺秋程 | 罗治负责 <code>getNeighborCount()</code> 方法实现, 贺秋程检查前面的方法是否有问题 |
| 2020.05.25 9: 30-11: 30 | 贺秋程 | 罗治 | 对代码进行测试, 并对错误进行修改 |

二、实验小结及体会

- 1) 掌握单元测试的方法;
- 2) 学习 JUnit 测试原理及框架;
- 3)掌握在 Eclipse 环境中加载 JUnit 及 JUnit 测试方法和过程。
- 4) 通过进一步的磨合, 对结对编程有了更深的理解, 也体会到了, 多人一起工作效率的明显提高。

成绩评定表:

| 序号 | 评分项目 | 满分 | 实得分 |
|----|----------------|----|-----|
| 1 | 实验报告格式规范 | 2 | |
| 2 | 实验报告过程清晰, 内容详实 | 4 | |
| 3 | 实验报告结果正确性 | 2 | |
| 4 | 实验分析与总结详尽 | 2 | |
| | 总得分 | 10 | |

