程序说明
version 1.0.0

修订历史

日期	版本	作者	描述
2015.8.21	1.0.0	GA207	实现 Jumper 类基本功能

目录

1 程序功能

功能名	实现函(后三者为 VOID)	说明
无参数构造 Jumper	Jumper()	Jumper 默认 **Hp** 为 **5**
回复生命值	**getHp()**	回复 **Hp**
反应	act()	满足移动条件则移动,否则顺时针转向 **90** 度
转向	turn()	向右转一个随机角度(45 度整数倍)
重生	**reborn()**	已死亡时，复活 **Jumper**
移动	move()	若前方两格不是花或者空位则掉血
受伤操作	**getHurt()**	当 **Hp** 小于 **0** 时，移除目标
移动判断	canMove()	判断是否满足移动条件（前方的第一个没有障碍物）

2 实现过程

导入 gridworld.jar，新建 Jumper.java 与 JumperRunner.java 两个文件。根据题目要求实现 Jumper 类，借用 Bug 类来实现，所以很多与 Bug 类相似。我们把 Bug 类直接复制粘贴至 Jumper 类，然后修改参数，增加了 jump，canJump 函数来实现跳跃功能，然后依旧保留 Bug 类中只移动一步的 move，canMove。需要注意的是，要把 move 函数里面的生成花朵的相关语句删除掉，因为不要求 Jumper 在移动过的地方留下花朵。为了判断是否可以移动，则借助 Bug 类中判断是否能在特定位置放置 Actor 类的语句，如下：

```
public void move() {
    Grid<Actor> gr = getGrid();
    if (gr == null)
        return;
    Location loc = getLocation();
    int dir = getDirection();
    Location next = loc.getAdjacentLocation(dir).getAdjacentLocation(dir);
    if (gr.isValid(next)) {
        Actor neighbor = gr.get(next);
        if ((neighbor != null) && !(neighbor instanceof Flower)) {
            getHurt();
            Location to = next.getAdjacentLocation(dir);
            if (gr.isValid(to))
                neighbor.moveTo(to);
            else
                neighbor.removeSelfFromGrid();
            if (neighbor instanceof Jumper) {
                Jumper jumper2 = (Jumper)neighbor;
                jumper2.getHurt();
            }
        }
        if (hp > 0)
            moveTo(next);
    }
}
```

然后直接运行程序观察 Jumper 的移动情况是否达到预期效果。

为了进一步验证实现是否正确，使用 JUnit 进行单元测试，详情请参见测试报告。

然后使用 Sonar 检查代码质量.正确配置环境后，修改 sonar.projectName，java-module.sonar.projectBaseDir, sonar.projectKey 全为 src(如下图)

```
sonar.projectKey=src
sonar.projectName=src
sonar.projectVersion=1.0
sonar.sourceEncoding=UTF-8
sonar.modules=java-module

java-module.sonar.projectName=Java Module
java-module.sonar.language=java

java-module.sonar.sources=.
java-module.sonar.projectBaseDir=src
```

这样即可把 src 里面的文件进行风格检测，然后根据实训要求以及 sonar 分析结果改进代码。

Ant 运行：

进入文件目录，输入 ant run 运行程序，输入 ant test 进行测试并查看结果。

3 总结体会

由于这次 **Group Job** 分工较为明确，且提供了报告模板，所以小组成员配合较为默契，效率也挺高的。参照 **Bug** 类，我们慢慢扩展出了现在的程序。

这次小组作业着实让我们受益匪浅，已经许久没有这样几个小伙伴好好分配工作，认真工作，然后收获成功的果实了。写程序，写测试，写报告，分工明确，任务的完成效率高了许多。

此外，在写代码时要好好参照原有的程序，在认真思考原有程序的架构后再去扩展写自己的代码。

这次小组作业圆满完成，期待下次也能一样顺利！