

实验课程名称： 软件工程基础实验

实验项目名称	结对编程			实验成绩	
实 验 者	王汉成	专业班级	软件 1804	组 别	
同 组 者	王龙祥			实验日期	

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) 体验敏捷开发中的两人合作。
- 2) 进一步提高个人编程技巧与实践。
- 3) 通过 Java 程序语言实现“生命游戏”的开发。

二、实验内容及要求

- 1) 选择一个程序实例，练习结对编程(pair programming)实践；
- 2) 要求学生两人一组，自由组合。每组使用一台计算机，二人共同编码，完成实验要求。
- 3) 要求在结对编程工作期间，两人的角色至少切换 4 次；
- 4) 编程语言不限，版本不限。建议使用 Python 或 JAVA 进行编程。

三、实验意义

通过敏捷开发过程中的两人合作的方式，体验与单独一人编程的不同，感受合作开发带来的好处，效率的提升和合作的力量。在结对编程的过程中进一步提高个人变成的技巧和能力，与同伴相互交流、取长补短，通过对一具体项目的实践，提升自身能力，锻炼合作交流的能力，为以后个人的发展奠定基础。

四、问题描述

生命游戏是英国数学家约翰·何顿·康威在 1970 年发明的细胞自动机，它包括一个二维矩形世界，这个世界中的每个方格居住着一个活着的或死亡的细胞。一个细胞在下一个时刻生死取决于相邻八个方格中活着的或死了的细胞的数量。

游戏在一个类似于围棋棋盘一样的，可以无限延伸的二维方格网中进行。例如，设想每个方格中都可放置一个生命细胞，生命细胞只有两种状态：“生”或“死”。图中，用黑色的方格表示该细胞为“死”，其它颜色表示该细胞为“生”。游戏开始时，每个细胞可以随机地（或给定地）被设定为“生”或“死”之一的某个状态，然后，再根据如下生存定律计算下一代每个细胞的状态：

- （1）每个细胞的状态由该细胞及周围 8 个细胞上一次的状态所决定；
- （2）如果一个细胞周围有 3 个细胞为生，则该细胞为生，即该细胞若原先为死则转为生，若原先为生则保持不变；
- （3）如果一个细胞周围有 2 个细胞为生，则该细胞的生死状态保持不变；
- （4）在其它情况下，该细胞为死，即该细胞若原先为生则转为死，若原先为死则保持不变。

五、主要仪器设备及耗材

设备：PC

开发环境：Idea、Eclipse

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）

一、算法设计思路

生命游戏实际上是一个二维的矩阵网格，每个网格代表一个生命体，即细胞。一个细胞的生死由周围相邻的八个细胞的状态决定，规则如下：

如果一个细胞周围有 2 个细胞，则该细胞的生死状态不变；

如果一个细胞周围有 3 个细胞，则该细胞为生；

其他状况，不论周围细胞生死多少，该细胞都为死。

基于以上情况，设计两个类，其中一个为 **Cell** 细胞状态类，另一个类为 **GUI** 页面功能类。

Cell 类中，主要封装了细胞相关的属性和方法，包括长、宽、代数和细胞的状态，用 **int** 类型二维数组 **grid[i][j]** 表示一个细胞，其中的 **i**、**j** 表示细胞的坐标，当 **grid** 的值为 0 时，细胞为死；**grid** 的值为 1 时，细胞为生。

首先将所有的细胞初始化都为死，即设定 **grid** 的值为 0，初始化细胞时，利用随机数，取 0-1 间的任意一个数，判断是否大于 0.5，随机决定每个细胞的生死，将随机判定的状态保存到 **grid** 二维数组中，即可得到随机生成的细胞初代。

细胞的状态变化是取决于周围的细胞状态的，这一过程可以理解为是细胞的繁衍，通过判断细胞周围活着细胞的个数，根据上述规则，判定每一个细胞的状态，将所有的细胞状态，即二维数组进行更新，得到一个有 0 有 1 的新的二维数组，实现一代细胞的更新。

获取邻居细胞的数量方法为，对于第 **i**、**j** 个细胞来说，遍历从 **(i-1, j-1)** 到 **(i+1, j+1)** 这九个细胞状态，当有细胞活着的时候，将数量 **count** 加一，遍历结束后，再将 **count** 减一，即减去自己，即可得到该细胞周围八个细胞中，活着的细胞的个数。

利用 **java.swing**，实现页面的呈现，页面的构成有两部分组成，一部分为按钮组成的网格，表示每一个细胞，另一部分为功能按钮部分，实现相应的功能。

在实现 **GUI** 的过程中，首先将静态的界面布局完成，对于细胞部分使用网格布局，每一个位置对应一个按钮，即一个细胞，用颜色表示细胞的生死，黑色为死，白色为生；对于按钮部分，分别设置开始、暂停、退出、清除等按钮，分别实现相应的功能。

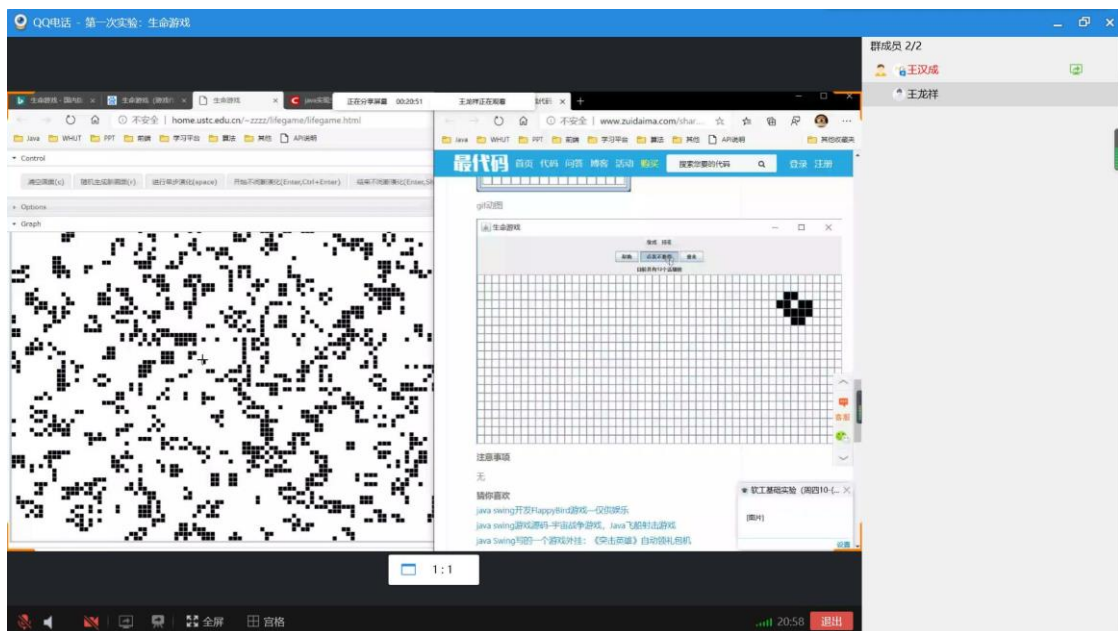
静态布局完成后，分别实现对应按钮的事件监听，通过对点击按钮的判断，利用 **getSource()** 的方法，分别实现不同的功能，主要判断 **isRunning** 遍历是否为 **true**，决定是否进行自动繁衍，判断所有细胞是否死亡，决定繁衍的停止。

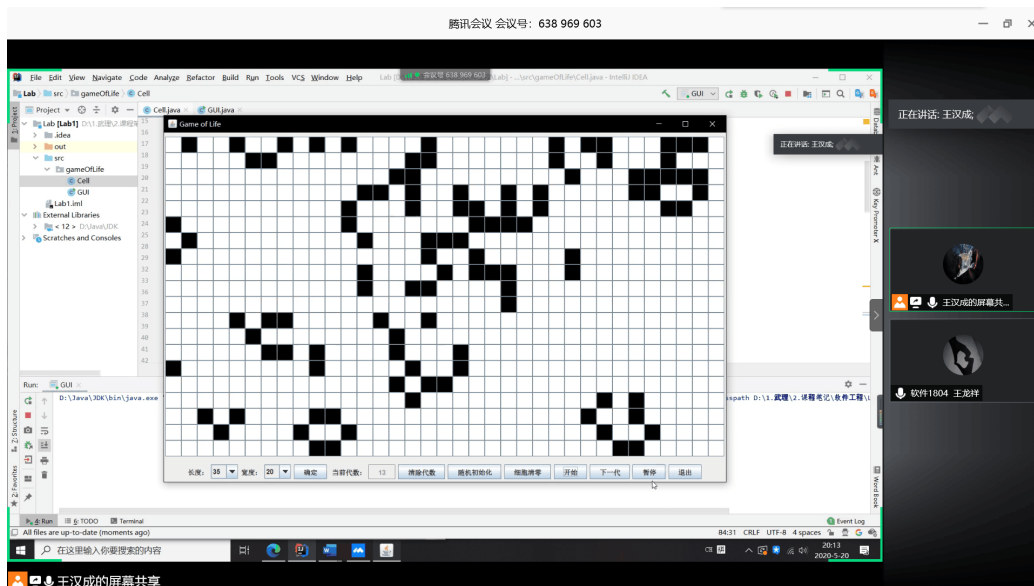
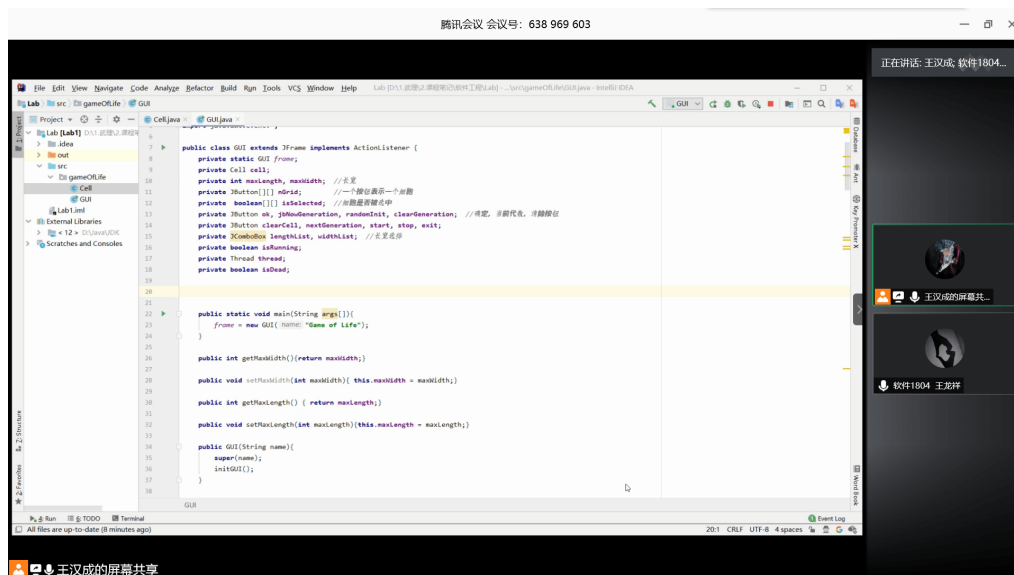
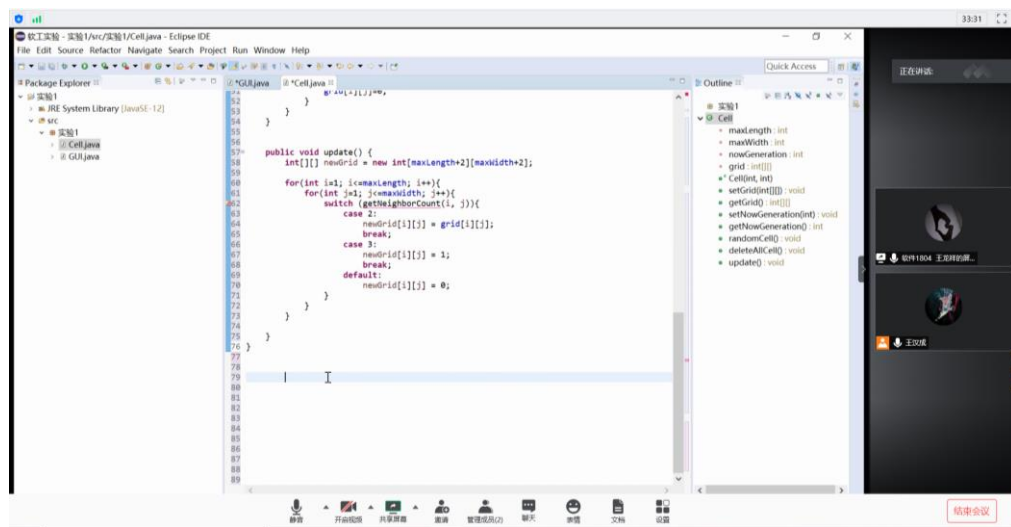
二、结对编程过程

(1) 任务分工表

时间	Driver	Observer	主要工作
2020.05.14 19: 30-21: 30	王汉成	王龙祥	进行 Cell 类的设计和编写，驾驶员负责编写，观察员查询相关算法
2020.05.16 15: 00-17: 00	王龙祥	王汉成	对 Cell 类进行检查和修改，驾驶员负责操作，观察员负责对出现的错误提出修改方法
2020.05.17 14: 30-16: 30	王汉成	王龙祥	进行 GUI 的设计，驾驶员负责编写和设计，观察员负责查询相关函数和完成对接。
2020.05.17 16: 30-18: 30	王龙祥	王汉成	对程序进行整体测试和修改，驾驶员进行操作，观察员提出修改意见

(2) 工作照片



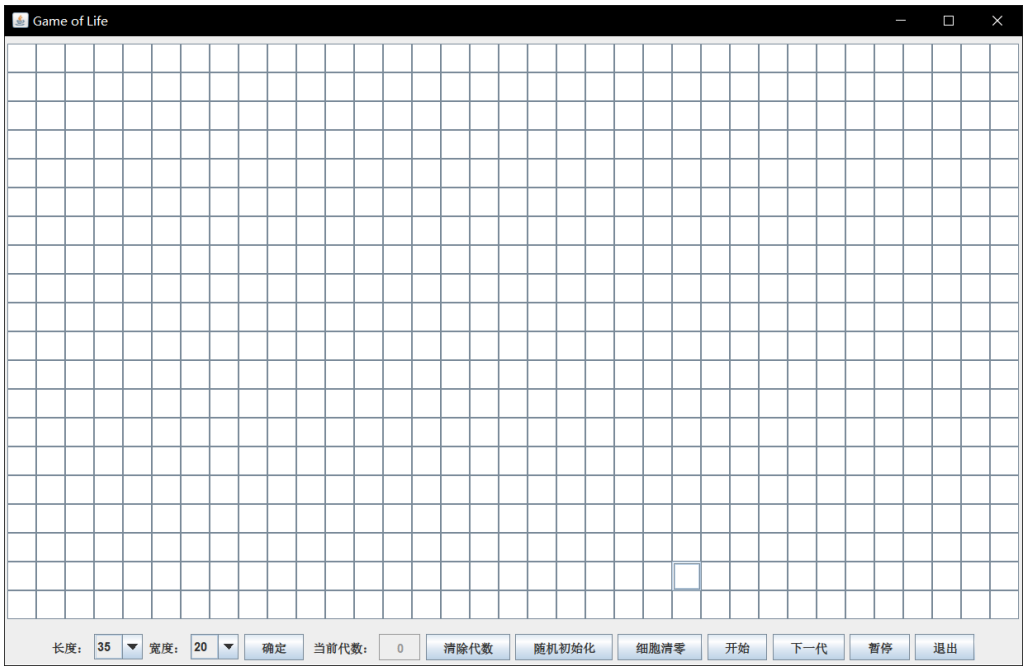


第三部分 结果与讨论（可加页）

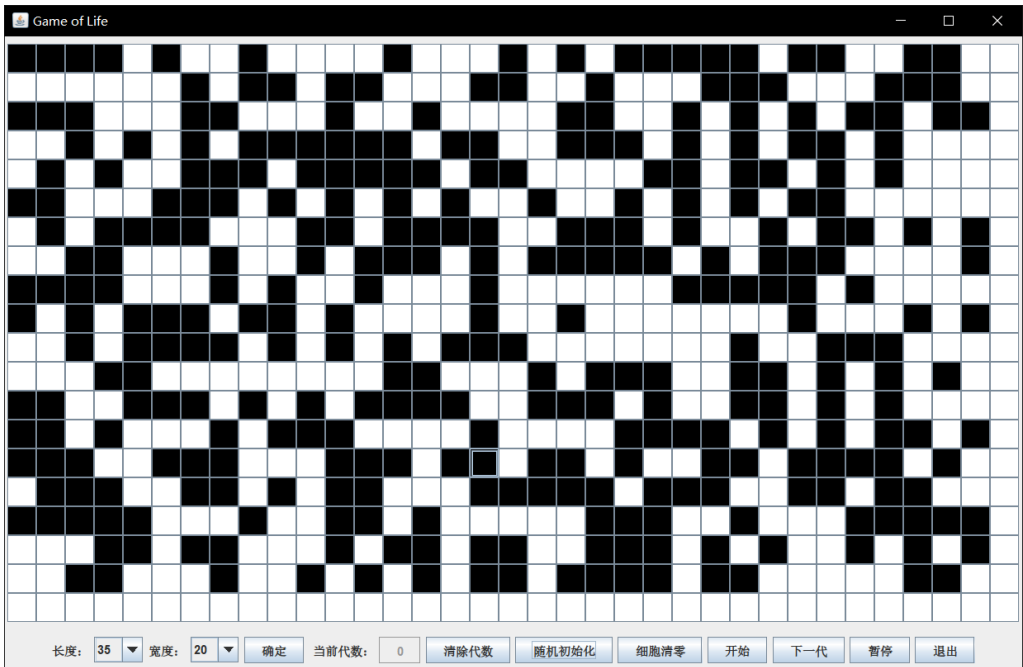
一、实验结果分析（包括数据处理、实验现象分析、影响因素讨论、综合分析和结论等）

实验结果：

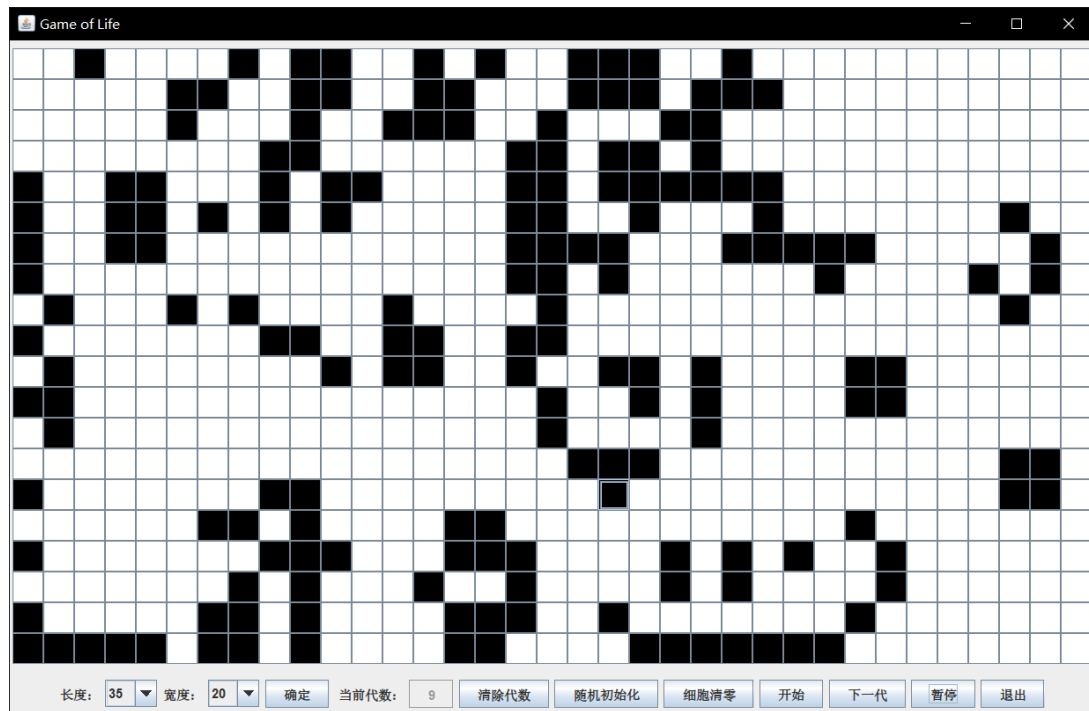
（1）静态界面



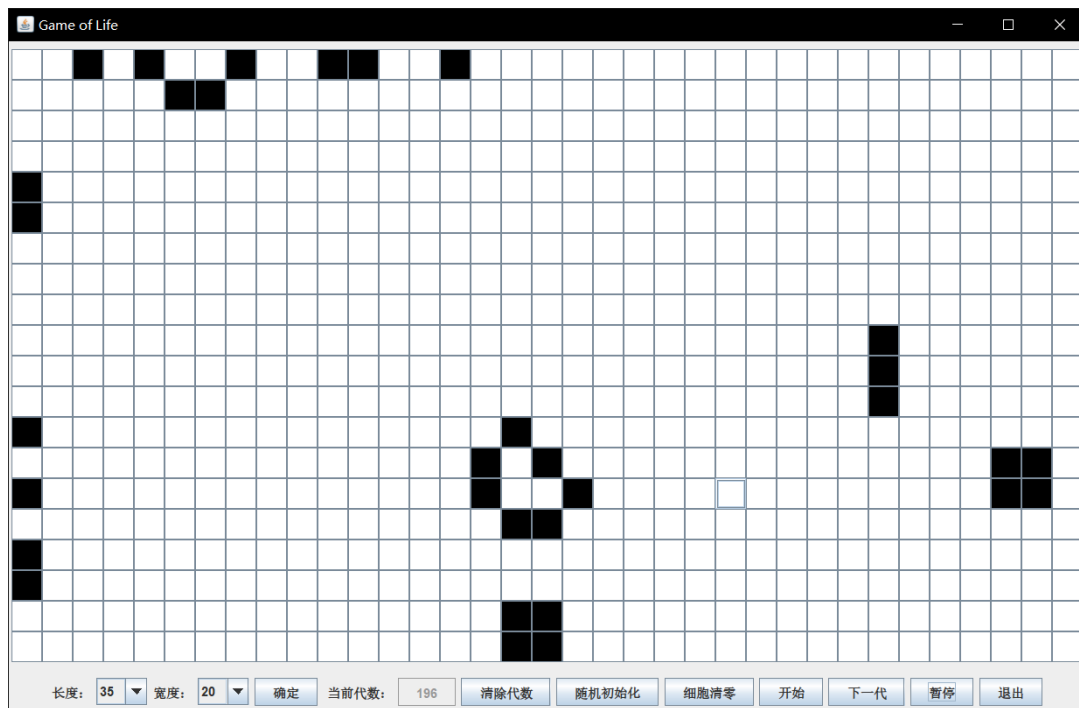
（2）随机生成细胞初代



(3) 开始繁衍



(4) 最终结果



可见图中的状态已经趋于稳定，特殊的形状产生后周围细胞状态将不再改变。

二、实验小结及体会

本次实验让我体验到了全新的编程方式，即敏捷开发中的结对编程，这一全新的编程让我最开始有些不适应，但是最终也有所收获，产生了许多感想。

结对编程相对于个人变成的差别还是很大的，首先对于项目的理解上，本次项目实现了生命游戏的编程，在结对编程的过程中，首先我和组员对项目的思路进行讨论，很快地就确定了主要的技术路线以及一些细节上的点，这在我个人编程中是很难做到的，一个人编程的时候思路 and 想法不会那么发散和完全，对于一些细节上的问题更是无法做到面面俱到，常常是踩过坑之后才发现问题的，结对编程很好地避免和解决了这一问题，帮助我们能更顺利地开展实验。

在结对编程中也出现了一些问题，对于具体的操作的时候，我们常常会有不一样的想法，在讨论的时候也就将进度拖慢了，但是我认为这并不是结对编程的弊端，正相反，通过对于具体实现的讨论、查阅资料最终的意见统一，从中让我们学会了更多的东西，提高了我们自身的编程水平和交流能力，在逐渐地对这种编程模式的熟悉之后，我们的效率也有所提升。

但是在结对编程中仍然需要注意几个问题，首先是编程风格的不同，我们的习惯不一样，很多缩进、命名就产生了一些分歧，最后我们统一了编程的风格，命名上也达到了意见的统一，使我们的项目看起来不会过于冲突。另外一点无法避免的是我们使用的编译环境不一样，Idea 编译器和 Eclipse 编译器在打开同一个项目的时候，对于注释会出现乱码，这也是我们编程过程中出现的一个小插曲。

总体来说，结对编程让我们熟悉了另外一种编程方式，增强了我们沟通交流的能力，在一定程度上也提高了我们个人的编程水平，希望在之后的实验中能够更加熟悉和习惯这种编程方式，充分发挥结对编程的优点，从而让我们的自身得到发展。

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	

