

实验课程名称： 软件工程基础实验

实验项目名称	代码评审与程序性能优化			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	2020.5.28

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) 了解代码审查的含义；
- 2) 了解如何对程序进行性能优化；
- 3) 掌握配置工具的安装与使用；

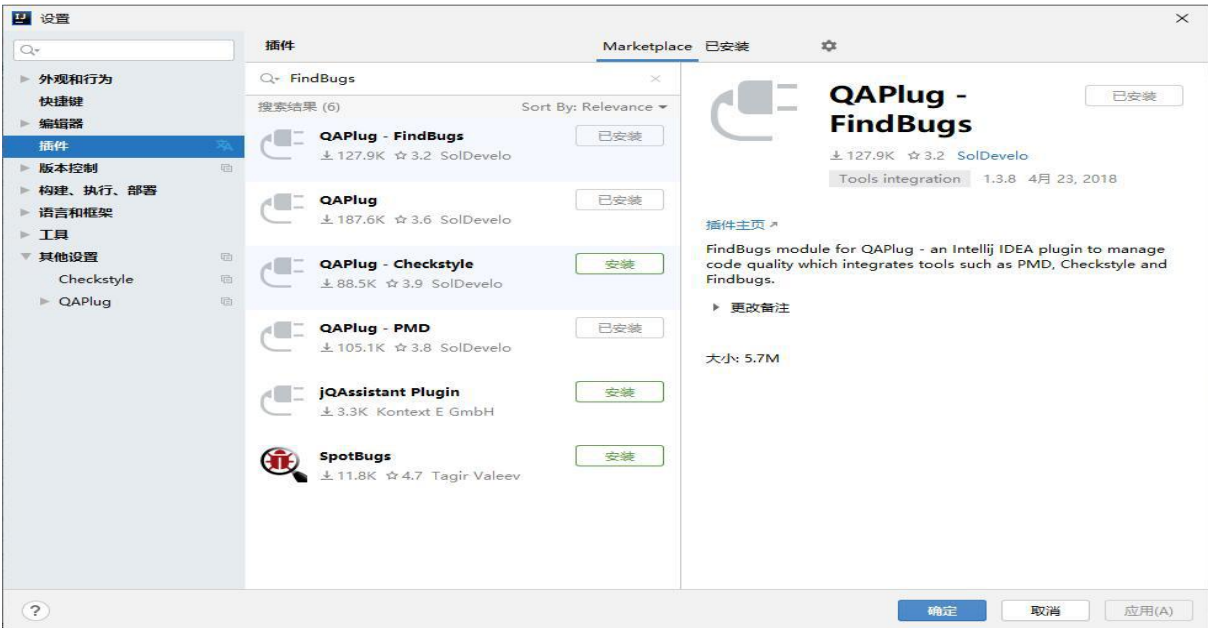
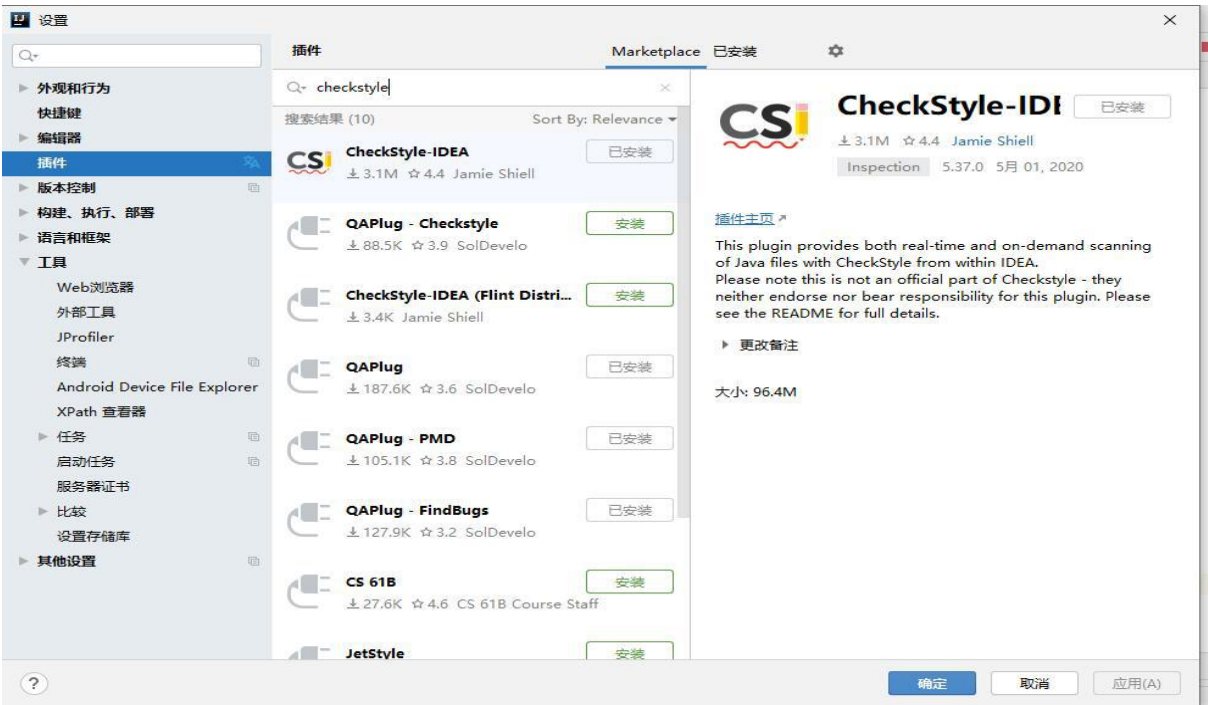
二、问题描述

- 1) 针对前面实验中所完成的代码，进行代码评审(走查)和性能分析，从时间性能角度对代码进行优化；
- 2) 练习代码评审的两个方面：静态分析、动态分析(profiling)；
- 3) 使用以下四个工具完成实验：
 - Checkstyle
 - FindBugs
 - PMD
 - JProfiler

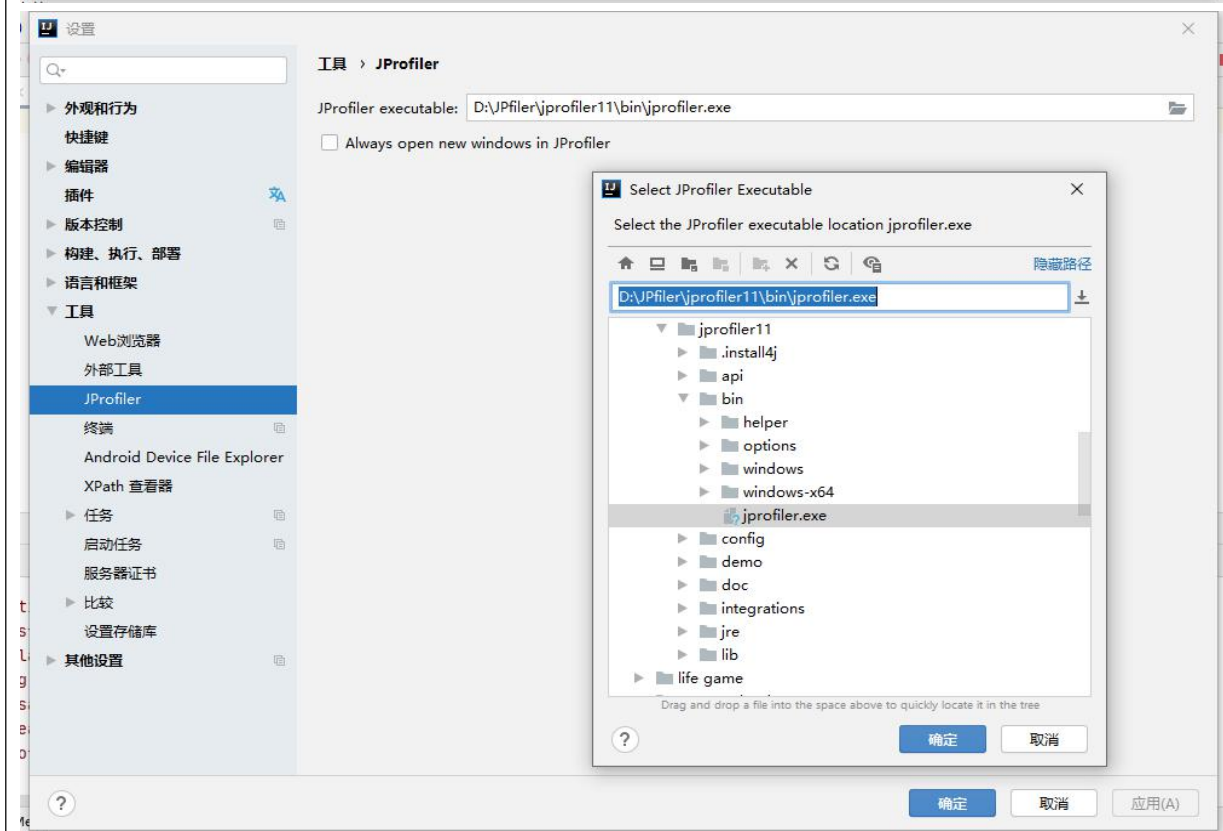
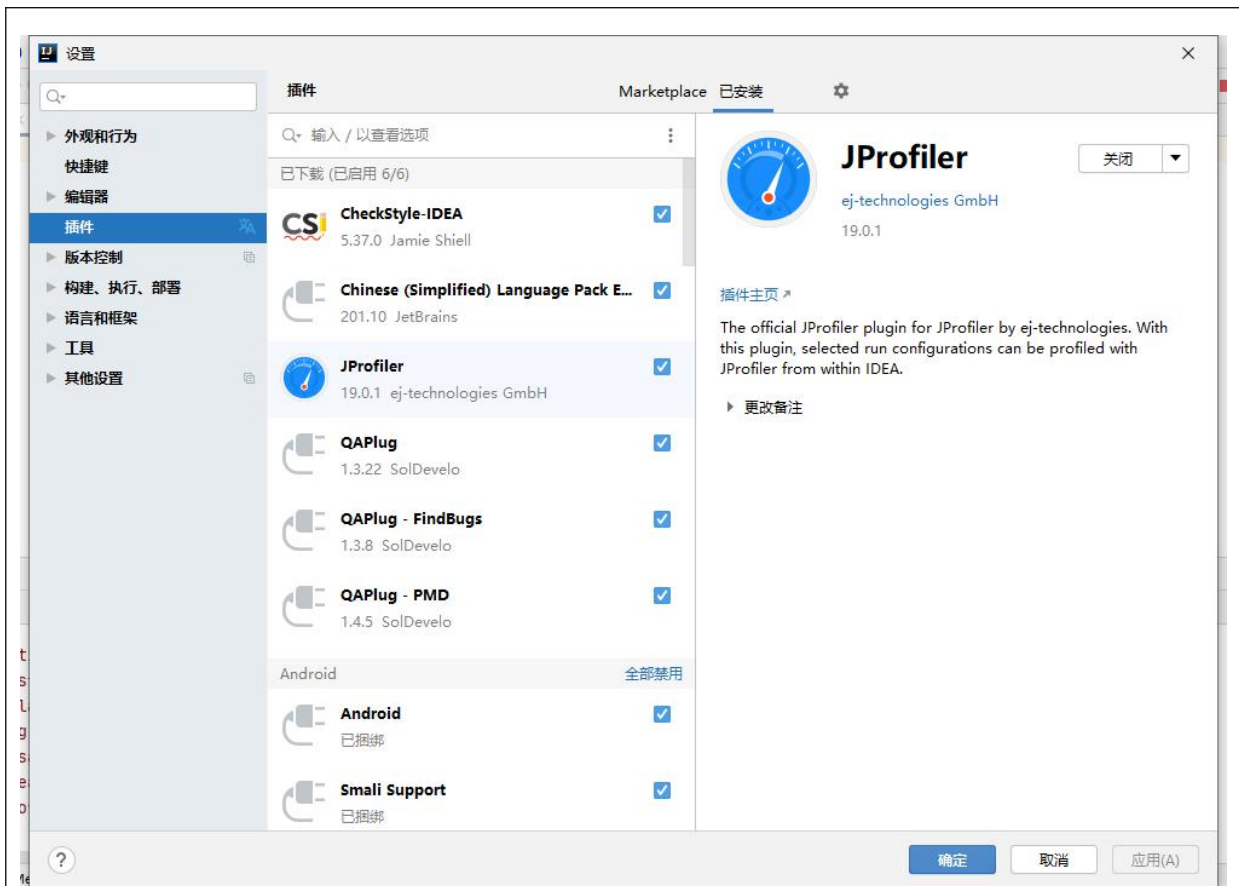
第二部分：实验过程记录（可加页）（包括实验原始数据记录，实验现象记录，实验过程发现的问题等）

一、在 IDEA 上安装必要的静态分析插件以及动态分析插件，来对代码进行分析。

静态分析插件安装过程如下图，安装完之后重启 IDEA 即可运用插件。

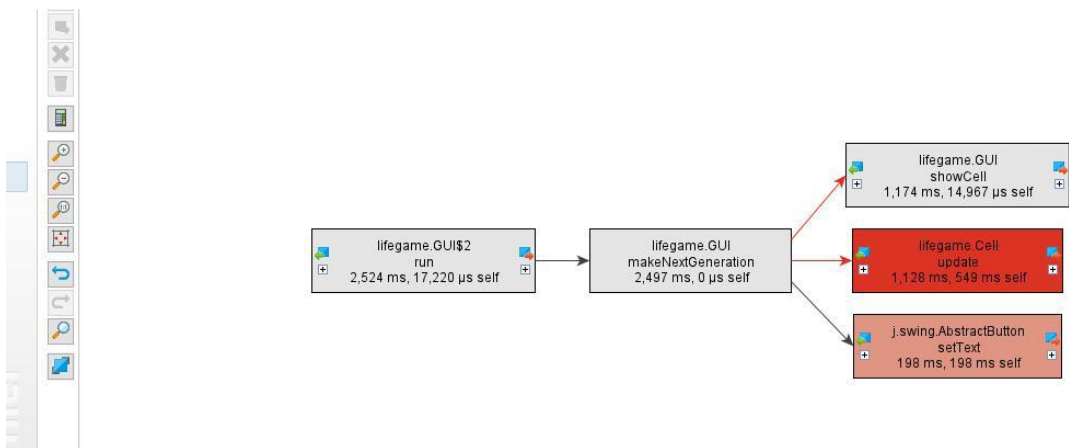
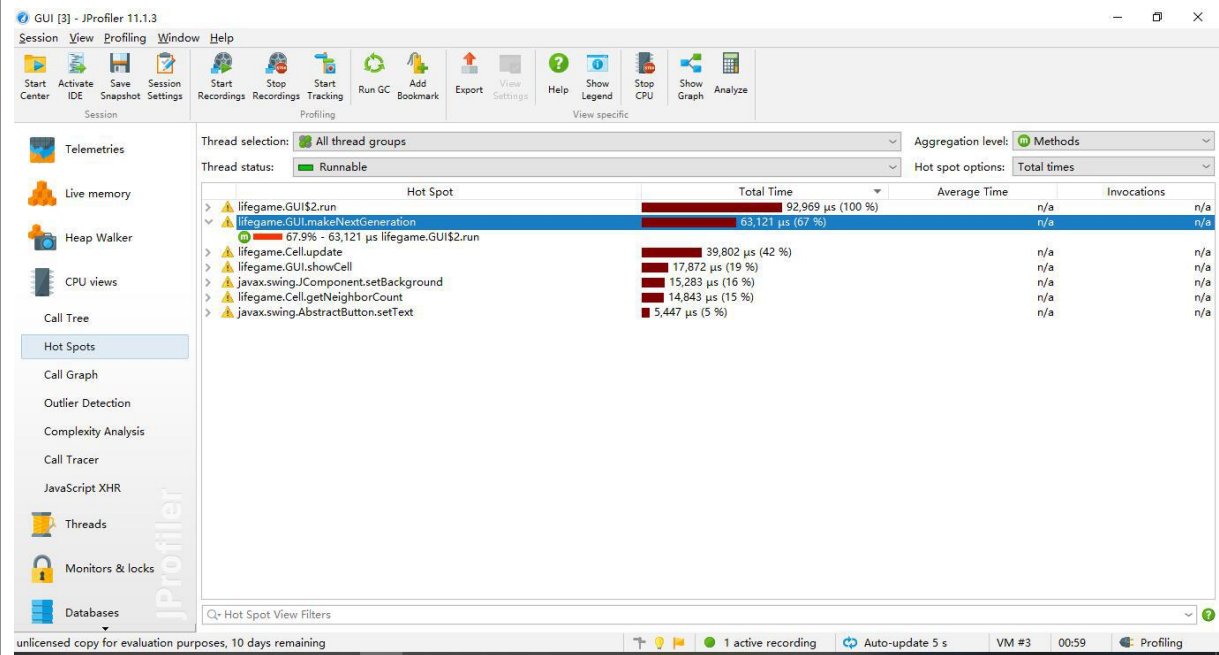


二、动态分析插件的安装以及配置过程如下：



在 IDEA 上下载 JProfiler 后，再在网站上找到相应的应用，下载好后进行相应的配置连结，即可运行分析代码。

二、对代码进行动态分析，结果如下图：



从图中可以看出，update 所占用的时间最多，所以优化应该从 update 入手，观察代码后进行了如下

```
public void update() {
    int[][] newGrid = new int[maxLength + 2][maxWidth + 2];
    for (int i = 1; i <= maxLength; i++)
    {
        for (int j = 1; j <= maxWidth; j++)
        {
            switch (getNeighborCount(i, j))
            {
                case 2:
                    newGrid[i][j] = grid[i][j]; // 周围有2个细胞，细胞状态保持不变
                    break;
                case 3:
                    newGrid[i][j] = 1; // 周围有3个细胞，细胞存活
                    break;
                default:
                    newGrid[i][j] = 0; // 其他情况，细胞死亡
            }
        }
        grid[i][j] = newGrid[i][j];
    }
    for (int i = 1; i <= maxLength; i++)
        for (int j = 1; j <= maxWidth; j++)
            grid[i][j] = newGrid[i][j]; // 把新一代细胞的情况赋值给原数组
    noGeneration++;
}
```

的修改，可以看出，修改后减少了一个二重循环，以此都达到优化大的目的。

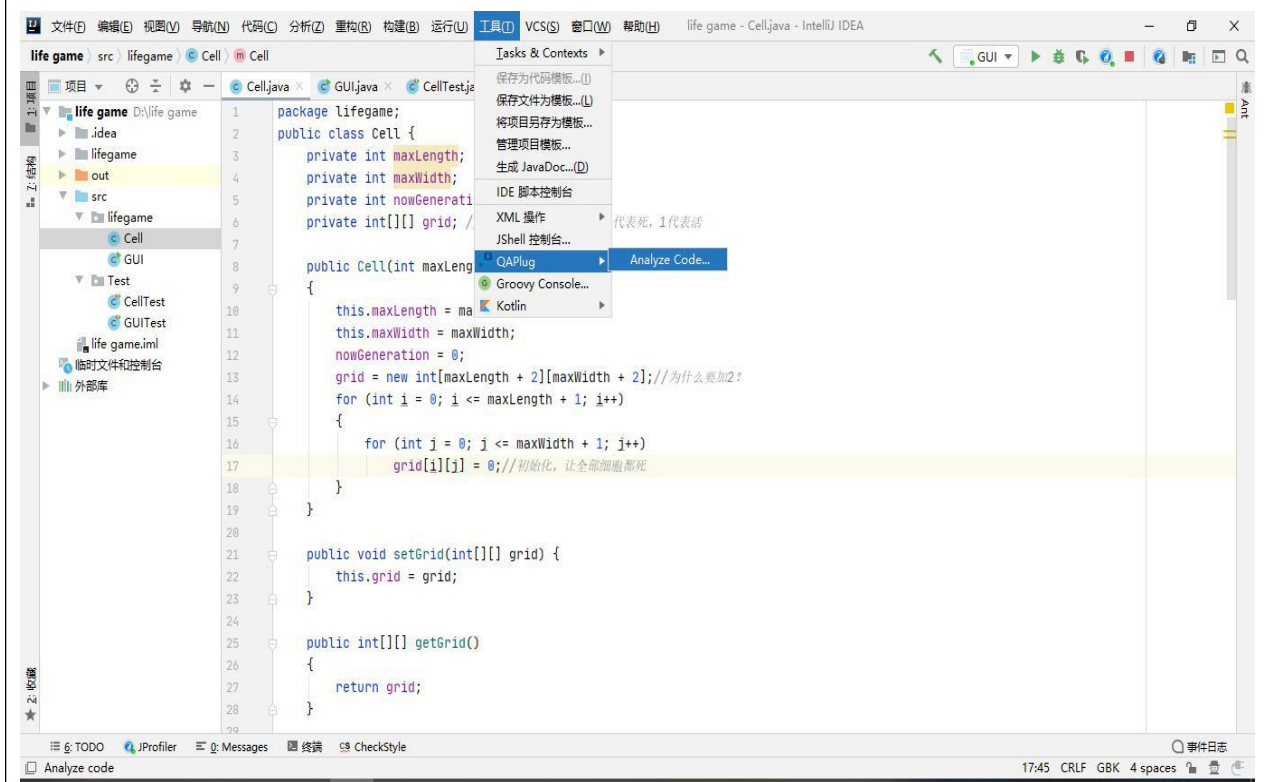
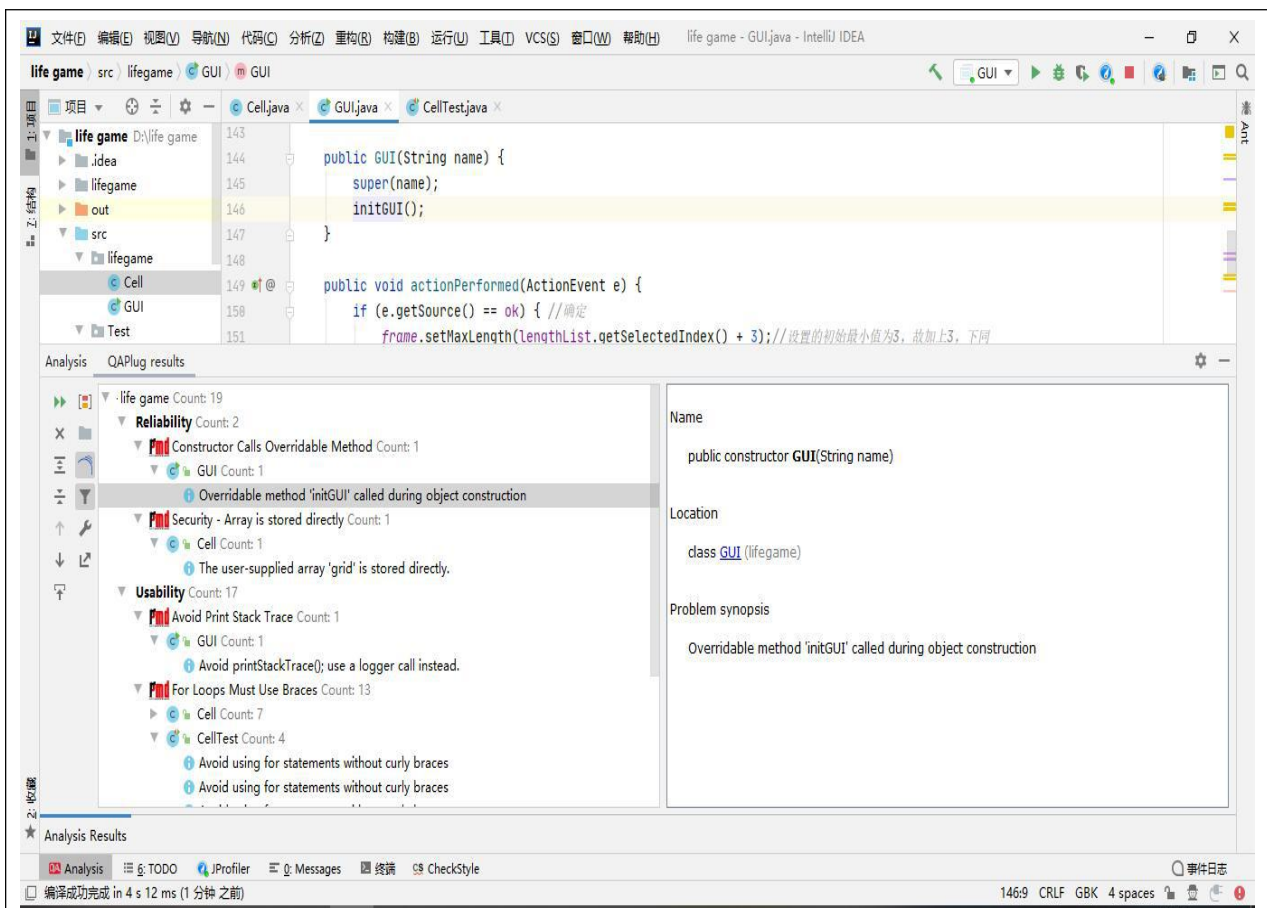
第三部分 结果与讨论（可加页）

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）

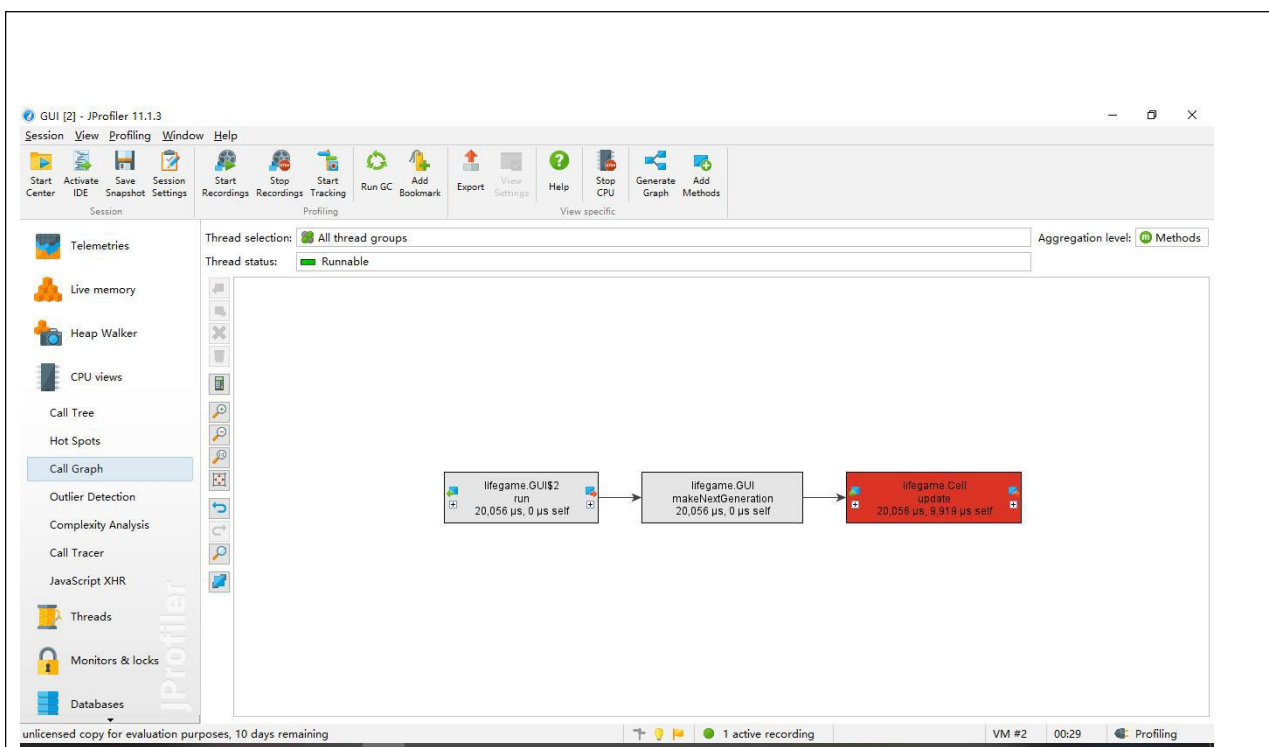
1、首先是静态分析插件对于代码的分析，结果如下图：

The screenshot displays the IntelliJ IDEA interface. The top pane shows the source code of `Cell.java` with several Checkstyle warnings highlighted in red. A tooltip for the first warning indicates 'Checkstyle: 缺少 Javadoc.' (Checkstyle: missing Javadoc). The bottom pane shows the 'Settings' dialog for the 'QAPlug' plugin. The 'Suppressions' section is empty. The 'QAPlug auto updates' section shows 'Check for updates' is checked and set to 'Weekly'. The 'Last check for updates' is '星期一, 六月 1, 2020' and the 'Next check for updates' is '星期二, 六月 2, 2020'. A 'Check & update' button is present. Below this, a table lists installed and available versions for various plugins.

Plugin name	Installed	Available	Plugin status
QAPlug	1.3.22	1.3.22	is up to date
QAPlug - FindBugs	1.3.8	1.3.8	is up to date
QAPlug - PMD	1.4.5	1.4.5	is up to date
QAPlug - Checkstyle		1.4.0	not installed



2、其次是改动后动态分析插件的分析结果：



从图中可以看出，修改后的时间大大减少。

时间		“驾驶员”	“领航员”	具体分工
2020.05.28	20：30-22：30	罗治	贺秋程	罗治负责静态插件的安装，贺秋程查找相关用法
2020.05.30	14：30-16：30	贺秋程	罗治	贺秋程负责对静态分析插件的操作，罗治进行动态分析插件的安装
2020.05.31	18：30-20：30	罗治	贺秋程	罗治与贺秋程一起进行动态分析插件的运行及分析。
2020.06.01	9：30-11：30	贺秋程	罗治	对代码进行测试，并对错误进行修改，并进行修改前后对比

二、实验小结及体会

- 1) 了解了代码审查的含义；
- 2) 了解了如何对程序进行性能优化；
- 3) 掌握了配置工具的安装与使用；
- 4) 对于代码的规范以及 bug 有了更深一步的了解；
- 5) 对于代码的错误以及优化有了更多的手段。

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	

