

哈尔滨工业大学计算机科学与技术学院

2017 年秋季学期《软件工程》

## Lab 4: 代码评审与程序性能优化

姓名	学号	联系方式
钟宇宏	1151200119	nanua.zqz@gmail.com/18846832803
王巍	1150340114	1450648595@qq.com/15546631826

# 目 录

第 1 章 实验要求 .....	1
第 2 章 在 IntelliJ 中配置代码审查与分析工具 .....	2
2.1 Checkstyle .....	2
2.2 PMD .....	2
2.3 FindBugs .....	2
2.4 VisualVM .....	5
第 3 章 本次实验所评审的代码 .....	7
第 4 章 代码 review 记录 .....	8
第 5 章 Checkstyle 所发现的代码问题清单及原因分析 .....	9
第 6 章 PMD 所发现的代码问题清单及原因分析 .....	10
第 7 章 FindBugs 所发现的代码问题清单及原因分析 .....	11
第 8 章 VisualVM 性能分析结果 .....	12
8.1 执行时间的统计结果与原因分析 .....	12
8.1.1 生成与展示有向图 .....	12
8.2 内存占用的统计结果与原因分析 .....	13
8.3 代码改进之后的执行时间统计结果 .....	13
8.4 代码改进之后的内存占用统计结果 .....	13
第 9 章 利用 Git/GitHub 进行协作的过程 .....	14
第 10 章 评述 .....	15
10.1 对代码规范方面的评述 .....	15
10.2 对代码性能方面的评述 .....	15
第 11 章 计划与实际进度 .....	16
第 12 章 小结 .....	17
参考文献 .....	18



## 第 1 章 实验要求

本次实验的要求为对 Lab1 所完成的代码进行代码的评审以及性能分析，并从性能角度对代码进行优化。其中，代码的评审包括静态分析以及动态分析两个方面，并且在实验中逐一使用 Checkstyle, FindBugs, PMD, VisualVM 四个工具对代码进行评审以及性能分析。

在本次实验中，采用 Lab1 的分组方式（即两人一组），并随机分配另一组作为本组的评审和分析对象，并要求实验期间不能与原作者进行沟通。

## 第 2 章 在 IntelliJ 中配置代码审查与分析工具

在本章中,将通过文字以及图片来对在 IntelliJ 中配置 Checkstyle, PMD, FindBugs, VisualVM 四种工具的步骤逐一进行叙述。

### 2.1 Checkstyle

在本节中,将描述在 IntelliJ 中安装以及配置 Checkstyle 的方法。

首先,在 IntelliJ 中打开项目,然后在顶端的 File 选项栏中选择 Setting 选项,并在弹出的窗口中的左侧边栏中选择 Plugins 选项卡。然后,在输入栏中输入“Checkstyle”,如图2-1所示。

然后,点击其中的 Search in repositories,在弹出的窗口中点击 Install,如图2-2所示。等待安装完成后,由此即完成了 IntelliJ 上 Checkstyle 的安装。

在安装完成后,我们在 Setting 中左边栏选择 Other Settings, Checkstyle。在出现的页面中的 Configuration File 选项中选择默认的 Sun Checks 作为风格检查的标准,如图2-3所示,然后点击 OK 进行确定。由此,即完成了 Checkstyle 的配置。

### 2.2 PMD

在本节中,将描述在 IntelliJ 中安装以及配置 PMD 的方法。

与 Checkstyle 的安装方式基本一致,首先我们在 Setting 中的 Plugins 选项卡内输入“PMD”,点击 Search in repositories,在出现的窗口中选择 PMDPlugin,点击 Install 进行安装,等待安装完成。安装完后如图2-4所示。

由于本次实验中我们将直接使用 PMD 默认的配置,因此不需要进行额外的配置。

### 2.3 FindBugs

在本节中,将描述在 IntelliJ 中安装 FindBugs 的方法。

与 Checkstyle 的安装方式基本一致,首先我们在 Setting 中的 Plugins 选项卡内输入“FindBugs”,点击 Search in repositories,在出现的窗口中选择 FindBugs-IDEA,点击 Install 进行安装,等待安装完成。安装完后如图2-5所示。

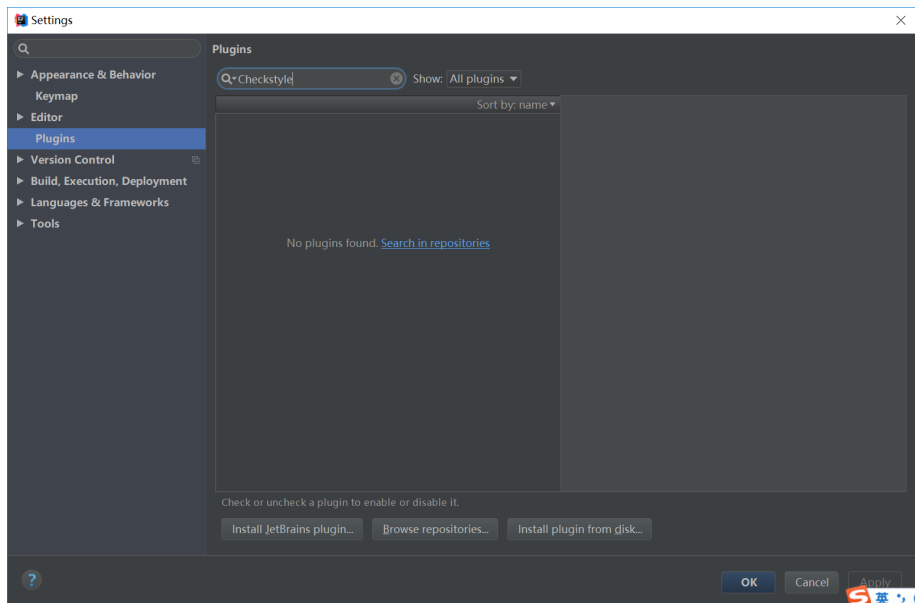


图 2-1 Plugins 选项卡

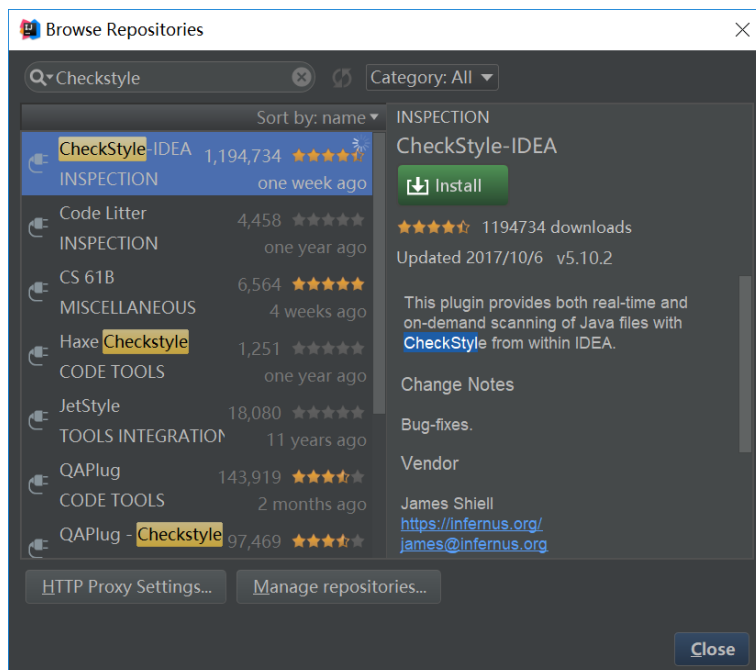


图 2-2 Browse Repositories 界面

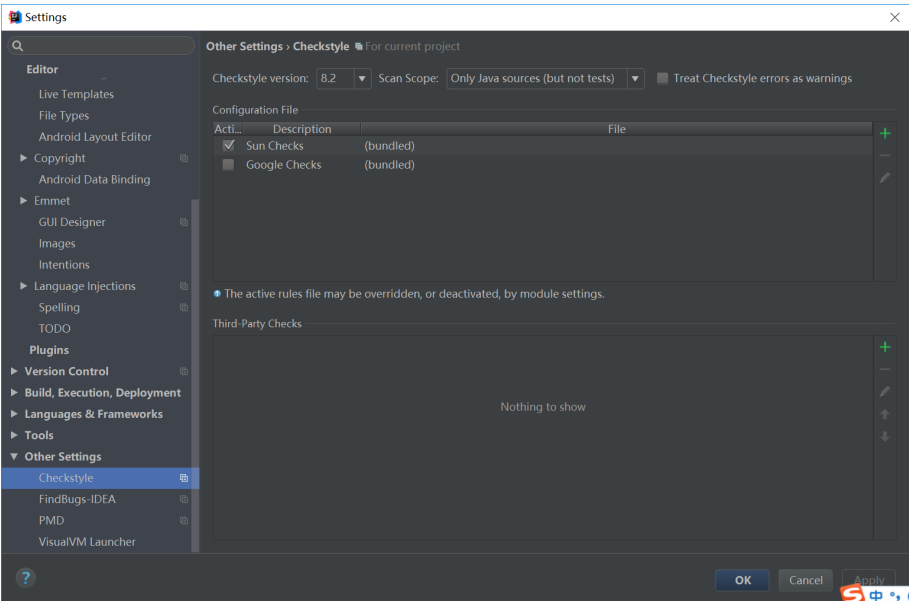


图 2-3 设置检查标准

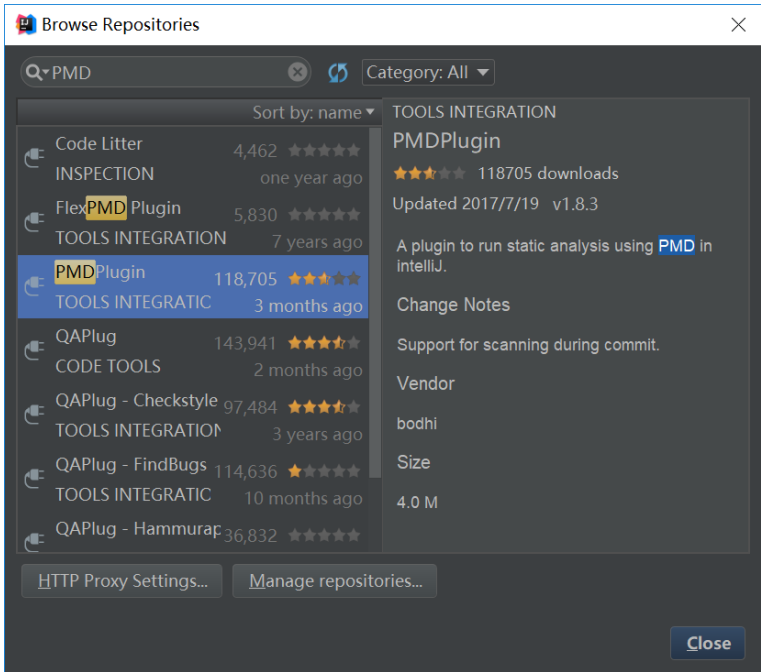


图 2-4 安装 PMD

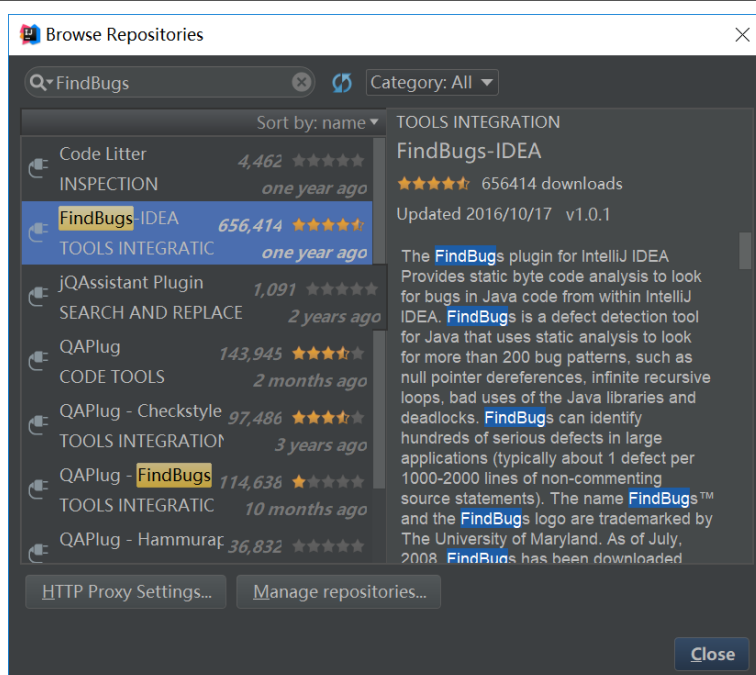


图 2-5 安装 FindBugs

## 2.4 VisualVM

在本节中，将描述在 IntelliJ 中安装 VisualVM 的方法。

由于 JDK 中自带 VisualVM 软件，因此不需要额外安装 VisualVM。不过，为了能够在 IntelliJ 中方便地使用 VisualVM，我们选择安装 VisualVM Launcher 插件。该插件的安装过程与 Checkstyle 的安装方式基本一致，首先我们在 Setting 中的 Plugins 选项卡内输入“VisualVM”，点击 Search in repositories，在出现的窗口中选择 VisualVM Launcher，点击 Install 进行安装，等待安装完成。安装完后如图2-6所示。



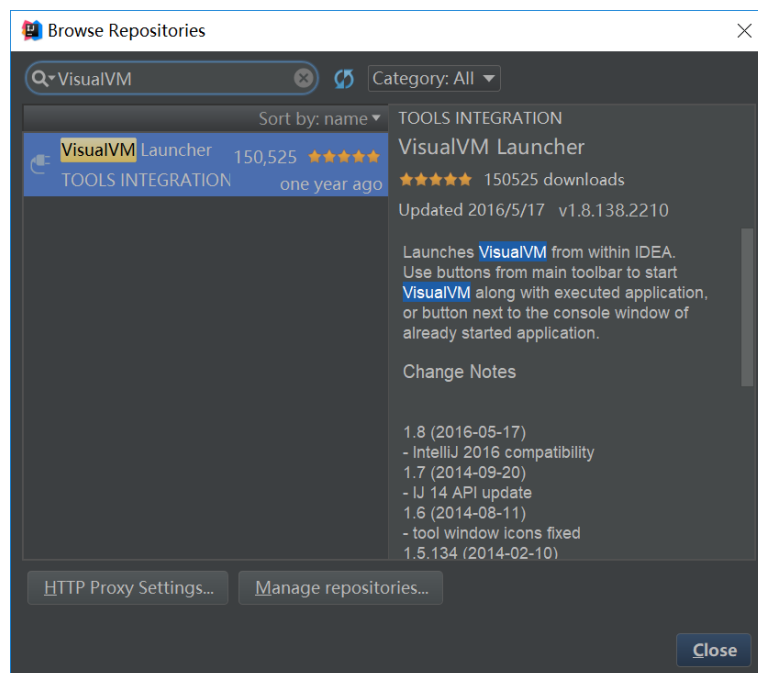


图 2-6 VisualVM Launcher 安装

## 第 3 章 本次实验所评审的代码

## 第 4 章 代码 review 记录

## 第 5 章 Checkstyle 所发现的代码问题清单及原因分析

## 第 6 章 PMD 所发现的代码问题清单及原因分析

## 第 7 章 FindBugs 所发现的代码问题清单及原因分析

## 第 8 章 VisualVM 性能分析结果

在本章中，将利用 VisualVM 工具对项目的执行时间以及内存占用情况进行分析，并根据分析得出的结果对项目代码进行改进。最后将测试改进后的代码的执行时间以及内存占用情况，并与为改进之前的情况进行对比。

### 8.1 执行时间的统计结果与原因分析

在本节中，将分别分析项目在生成与展示有向图，查询桥接词，根据桥接词生成新文本，计算两个单词之间的最短路径，以及随机游走这五个功能方面的耗时，并对耗时的原因进行分析。

需要说明的是，虽然在 Lab1 的原始要求中“读取并生成有向图”以及“展示有向图”为两个功能，但是在该项目中完成文件的读取以及有向图的生成后即自动展示了有向图，因此在此将原始要求中的两个功能需求合并为“读取并展示有向图”一个功能，并对这一个功能进行分析。

#### 8.1.1 生成与展示有向图

在生成与展示有向图的时间测试中，我们采用 Lab1 验收时的测试数据作为输入进行测试，耗时结果的统计见图8-1。

需要首先特别说明的是，MenuListener.actionPerformed() 的耗时为用户选择所要读取的文件时的耗时，因此不算在程序的运行时间内。在耗时图像中，Graphviz.get\_img\_stream() 为调用外部程序 Graphviz 绘制图像，并从

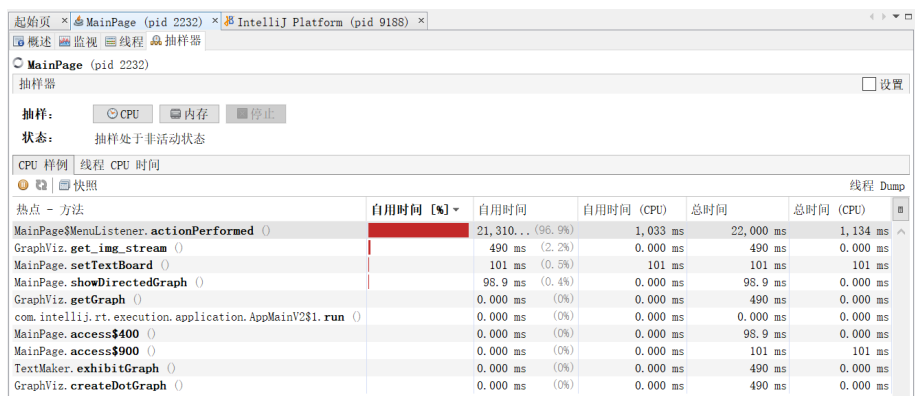


图 8-1 生成与展示有向图耗时

Graphviz 读取其返回的图像的二进制串的函数, `Graphviz.createDotGraph()` 为接受 Graphviz 源程序作为输入, 并最终将生成的图片写入到文件中的函数, `TextMaker.exhibitGraph()` 为生成程序中图的结构对应的 Graphviz 源程序, 并最终将生成的图片写入到文件中的函数, `MainPage.setTextBoard()` 为配置 GUI 左侧展示文本框的函数, `MainPage.showDirectedGraph()` 为配置 GUI 中展示有向图部分的 GUI 元素函数。

## 8.2 内存占用的统计结果与原因分析

## 8.3 代码改进之后的执行时间统计结果

## 8.4 代码改进之后的内存占用统计结果



## 第 9 章 利用 Git/GitHub 进行协作的过程

## 第 10 章 评述

### 10.1 对代码规范方面的评述

### 10.2 对代码性能方面的评述

## 第 11 章 计划与实际进度

## 第 12 章 小结

## 参考文献

- [1] 林来兴. 空间控制技术 [M]. 北京: 宇航出版社, 1992:25–42.
- [2] 辛希孟, 中国科学院文献信息中心, 孟广均, 等. 信息技术与信息服务国际研讨会论文集 [C]. 北京: 中国社会科学出版社, 1994:45–49.
- [3] 赵耀东. 新时代的工业工程师 [M/OL]. 台北: 天下文化出版社, 1998[1998-09-26]. [http://www.ie.nthu.edu.tw/info/ie.newie.htm\(Big5\)](http://www.ie.nthu.edu.tw/info/ie.newie.htm(Big5)).
- [4] 湛颖. 空间最优交会控制理论与方法研究 [D]. 哈尔滨: 哈尔滨工业大学, 1992:8–13.
- [5] Kanamori H. Shaking without Quaking[J]. Science, 1998, 279(5359):2063–2064.
- [6] Mlot C. Plant Biology in the Genome Era[J/OL]. Science, 1998, 281(5375):331–332[1998-09-23]. <http://www.sciencemag.org/cgi/collection/anatmorp>.
- [7] Lamport L. Paxos made simple[J]. ACM Sigact News, 2001, 32(4):18–25.