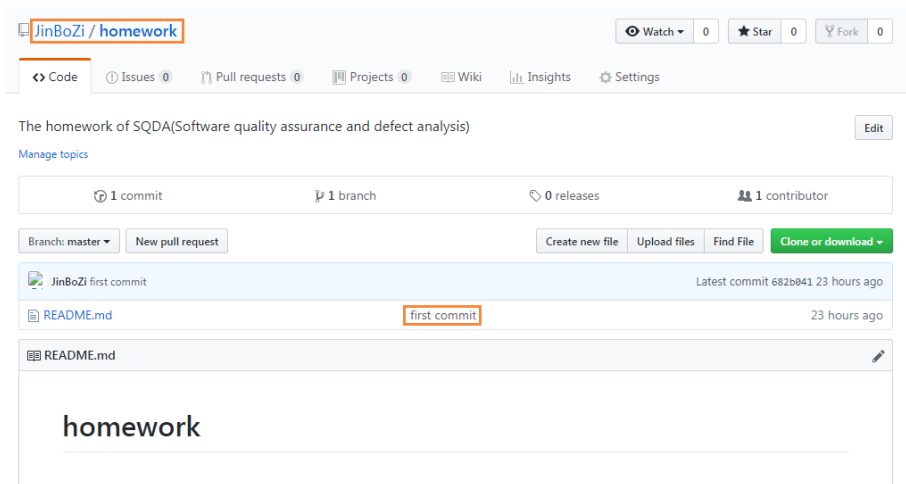


作业完成过程

1.准备阶段

在github上建立仓库，完成软件配置管理任务：<https://github.com/JinBoZi/homework.git>



2.查阅博客

通过搜索引擎，初步了解最基本的，工程上常用的软件度量及其工具；查阅博客，了解已有的自动化软件度量工具、插件，进而统计这些工具涉及的度量元。

- [SourceMonitor](#)
- [JDepend](#)
- [inFusion](#)

3.学术性较强的软件度量

通过博客与知网的过渡，了解软件度量学，理解比较出名的几组度量值；例如LK、CK、MOOD

- [邢大红,曹佳冬,汪和才,刘宗田.软件度量学综述\[J\].计算机工程与应用,2001\(01\):17-19.](#)
- [简介LK、CK、MOOD的博客](#)

4.新提出的软件度量

由于CK度量是面向对象中很经典的度量方法，故查看了一下近几年引用CK度量的论文，并通过这些论文的引用图谱，了解了一些新提出或者改进的软件度量。

- CK度量：Chidamber S R, Kemerer C F. A metrics suite for object oriented design[J]. IEEE Transactions on software engineering, 1994, 20(6): 476-493.
- ISCOM度量：Kansal D, Aher T, Joshi R K. Sensitivity and Monotonicity in Class Cohesion Metrics[C]//Proceedings of the 12th Innovations on Software Engineering Conference (formerly known as India Software Engineering Conference). ACM, 2019: 24.
- 有一个静态度量总结表：AbuHassan A, Alshayeb M. A metrics suite for UML model stability[J]. Software & Systems Modeling, 2019, 18(1): 557-583.
- 验证了几种度量的准确性：Pantiuchina J, Lanza M, Bavota G. Improving code: The (mis) perception of quality metrics[C]//2018 IEEE International Conference on Software Maintenance and Evolution (ICSME). IEEE, 2018: 80-91.
- C3度量：Marcus A, Poshyvanyk D, Ferenc R. Using the Conceptual Cohesion of Classes for Fault Prediction in Object-Oriented Systems[J]. IEEE Transactions on Software Engineering, 2008, 34(2):287-300.

其他未获取太多信息，没有仔细阅读的论文：

- AOP相关的度量，感觉与之前提到的度量重复度很高，因为这些度量都适用于OOP，这篇论文主要讲述了AOP的改进：Ghareb M I, Allen G. State of the art metrics for aspect oriented programming[C]//AIP Conference Proceedings. AIP Publishing, 2018, 1952(1): 020107.
- 名为综述，但确实没找到软件度量指标：Padhy N, Satapathy S, Singh R P. State-of-the-art object-oriented metrics and its reusability: a decade review[M]//Smart Computing and Informatics. Springer, Singapore, 2018: 431-441.
- Smart Contracts Software Metrics,没能理解：Tonelli R, Destefanis G, Marchesi M, et al. Smart contracts software metrics: a first study[J]. arXiv preprint arXiv:1802.01517, 2018.
- 代码可读性的度量，没有具体计算方式，用的模型，所以未仔细研读：Buse R P L, Weimer W R. Learning a Metric for Code Readability[J]. IEEE Transactions on Software Engineering, 2010, 36(4):546-558.

5.实现一种软件度量

由于时间原因，而且学术性软件度量所需的数据结构过于复杂，因此只计划实现**代码行**的度量。很多学术性的软件度量都未找到工具或者插件，实际工程中应该不常使用。

LOC功能极为简单，源代码与程序都在 .\LOC\ 目录下，在命令行模式下为其提供一个源代码路径，即可返回路径下各文件的代码行数

```
管理员: C:\Windows\system32\cmd.exe
H:\git\student\2018\刘彬彬\01课程学习\软件质量检测与缺陷分析\honeework\LOC>LOC H:\osdc
FileName Null lines Lines of code Annotation lines
H:\osdc\CMakeLists.txt 0 9 0
H:\osdc\Filer.cc 69 377 38
H:\osdc\Filer.h 42 229 31
H:\osdc\Journaler.cc 235 1135 207
H:\osdc\Journaler.h 69 348 120
H:\osdc\ObjectCacher.cc 343 2139 210
H:\osdc\ObjectCacher.h 124 570 73
H:\osdc\Objecter.cc 735 4256 298
H:\osdc\Objecter.h 286 2518 253
H:\osdc\Striper.cc 48 340 37
H:\osdc\Striper.h 18 55 40
H:\osdc\WritebackHandler.h 6 37 13
H:\git\student\2018\刘彬彬\01课程学习\软件质量检测与缺陷分析\honeework\LOC>
```

6.思考

因为还有别的作业，也希望尽快投入导师的任务，所以没有投入过多的时间与精力，希望老师谅解。尽管如此，我觉得通过完成这次作业还是有不小的收获的：

- 完整的自己使用了软件配置管理，通过github管理本次实践
- 了解到各种形式的软件度量，也了解到工程中经常使用的软件度量工具
- 学会了查阅论文的方法，通过经典论文的引用图谱找到更多相关资料

但相应的，不足之处也有很多：

- 只是自己使用github，并未与他人合作，也没有版本回退的操作，只是单纯的进行了资料托管，对软件配置管理工具的使用还很浅显
- 没有仔细去找学术性软件度量的工具，没有真正使用找到的工具
- 很多论文没有仔细阅读，其实应该还有许多软件度量没有继续往下找
- LOC度量工具的实现较为简陋