



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

Lab Manuals for
Software Construction

实验环境与提交指南



School of Computer Science and Technology

Harbin Institute of Technology

Spring 2018

目录

1	实验发布.....	1
2	实验环境.....	1
3	提交方式.....	1
3.1	注册个人 GitHub 账号.....	1
3.2	在 GitHub Classroom 中获取实验的 private repositories.....	2
3.3	本地 Eclipse 开发、Git 管理、提交至 GitHub 仓库	4
4	评分方式.....	5

1 实验发布

所有实验均在 CMS 上发布，请关注“作业/实验”区域所发布的信息（实验手册、提交截止日期）：

<https://cms.hit.edu.cn/course/view.php?id=341>

2 实验环境

- (1) 请阅读 <http://web.mit.edu/6.031/www/sp17/getting-started/>，按该页面列出的指南，在本地机器安装相应的开发环境（JDK、Eclipse、Git）并熟练掌握它们的配置过程。
- (2) 可以从 <http://web.mit.edu/6.031/www/sp17/getting-started/eclipse-faq/> 获取更多的 Eclipse 帮助。
- (3) 阅读 <http://web.mit.edu/6.031/www/sp17/classes/02-basic-java/> 了解 Java 的基本编程特性。
- (4) 关于 Git 的学习手册：<https://git-scm.com/book/en/v2>（英文版）、<https://git-scm.com/book/zh/v2>（中文版）。可使用 <https://www.shiyanlou.com/courses/4> 提供的在线实验环境进行 Git 练习。
- (5) 阅读 http://web.mit.edu/6.005/www/fa16/psets/ps0/#unit_testing，了解单元测试和 JUnit 工具。
- (6) 阅读 <https://github.com/junit-team/junit4/wiki/Download-and-Install>，并在自己的 Eclipse IDE 中安装配置 JUnit。
- (7) 阅读 <https://github.com/junit-team/junit4/wiki/Getting-started>，了解如何使用 JUnit 为 Java 程序编写测试代码并执行测试。

3 提交方式

3.1 注册个人 GitHub 账号

在 GitHub 上注册个人账号，user name 请使用自己的学号（例如 160310301）。如果之前已有 GitHub 账号，请修改 user name 为自己的学号，或者重新注册新账号。

若不遵循该规则，GitHub 上为你创建的 **private repositories** 将无法被持续集成工具检测到，因而无法得到分数。

3.2 在 GitHub Classroom 中获取实验的 private repositories

在实验手册中获取每次实验的 GitHub Classroom URL 地址，例如：

<https://classroom.github.com/a/z9utaaos>

访问该地址，用个人 GitHub 账号登录，同意授权 GitHub Classroom。

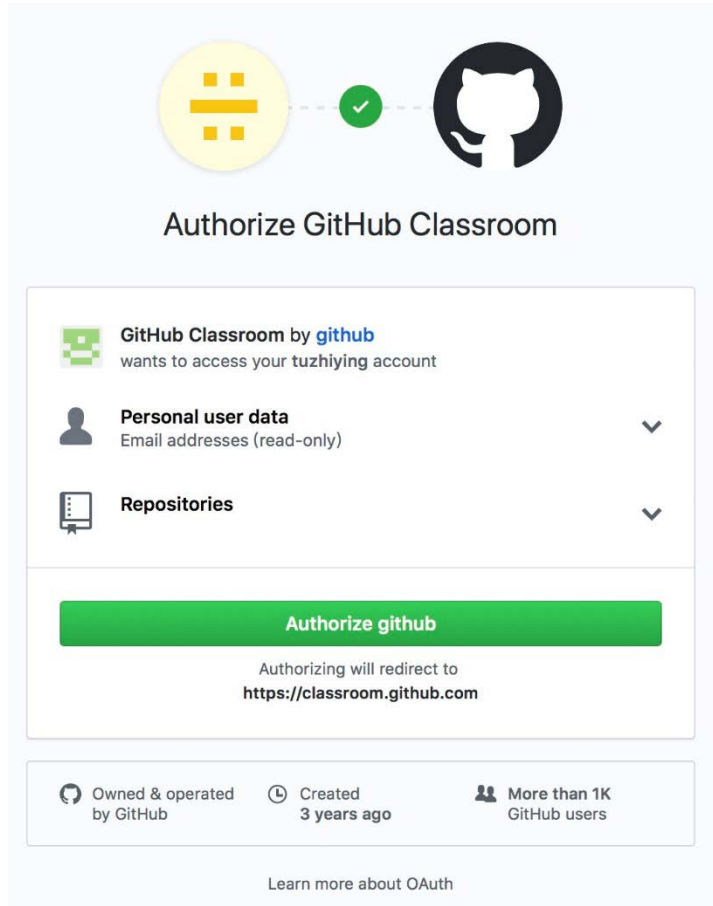


图 1

进入之后，出现以下界面，请选择你的学号，建立该学号与你的 GitHub 账号的关联。

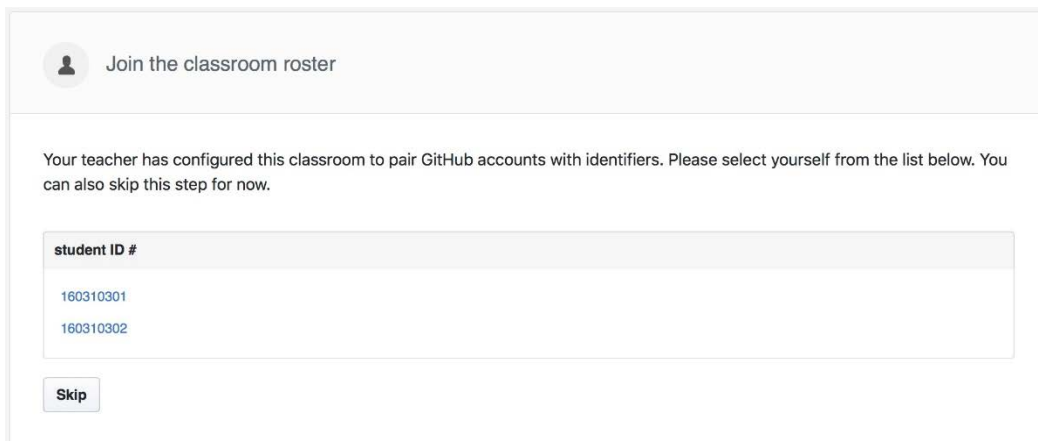


图 2

之后, GitHub classroom 服务为你自动创建了用于提交此实验的 Git repository (见图 3, 仓库名: **Labx-160310301**, x 为实验序号 1~6, 后面为你的学号)。点击 “Accept this assignment”, 即可获得该仓库的信息 (见图 4)。

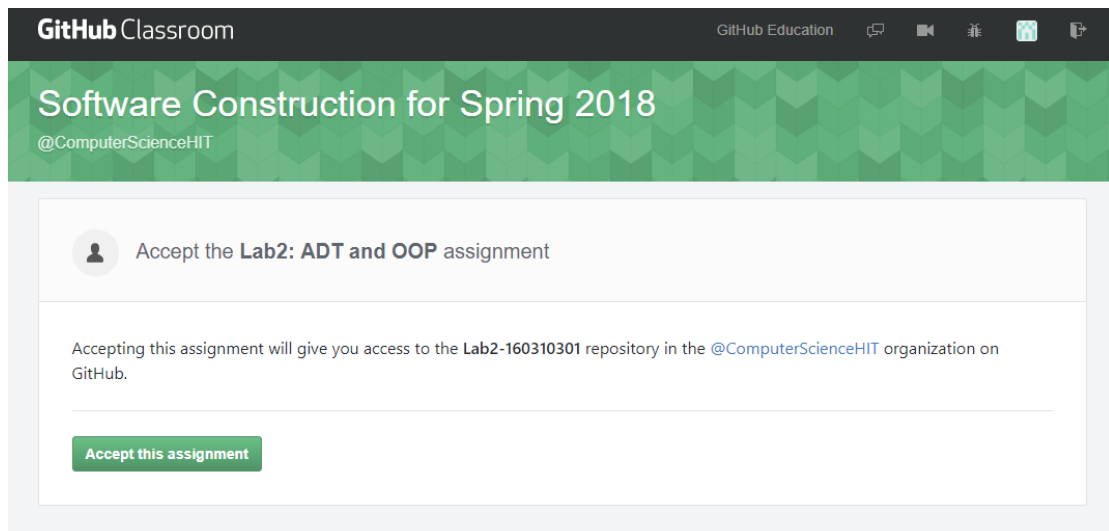


图 3

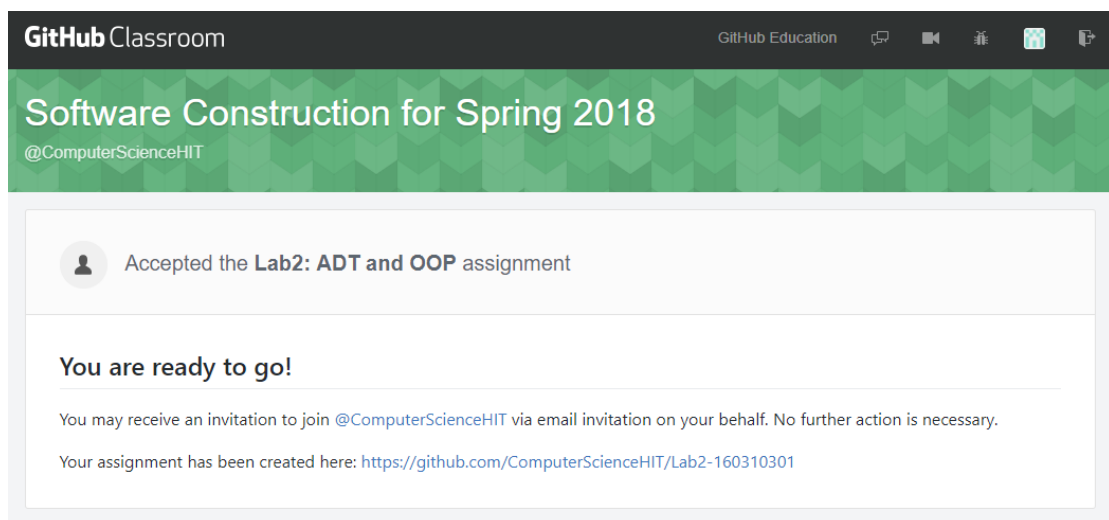


图 4

之后, 你就可以进入该仓库。最初仓库是空的。

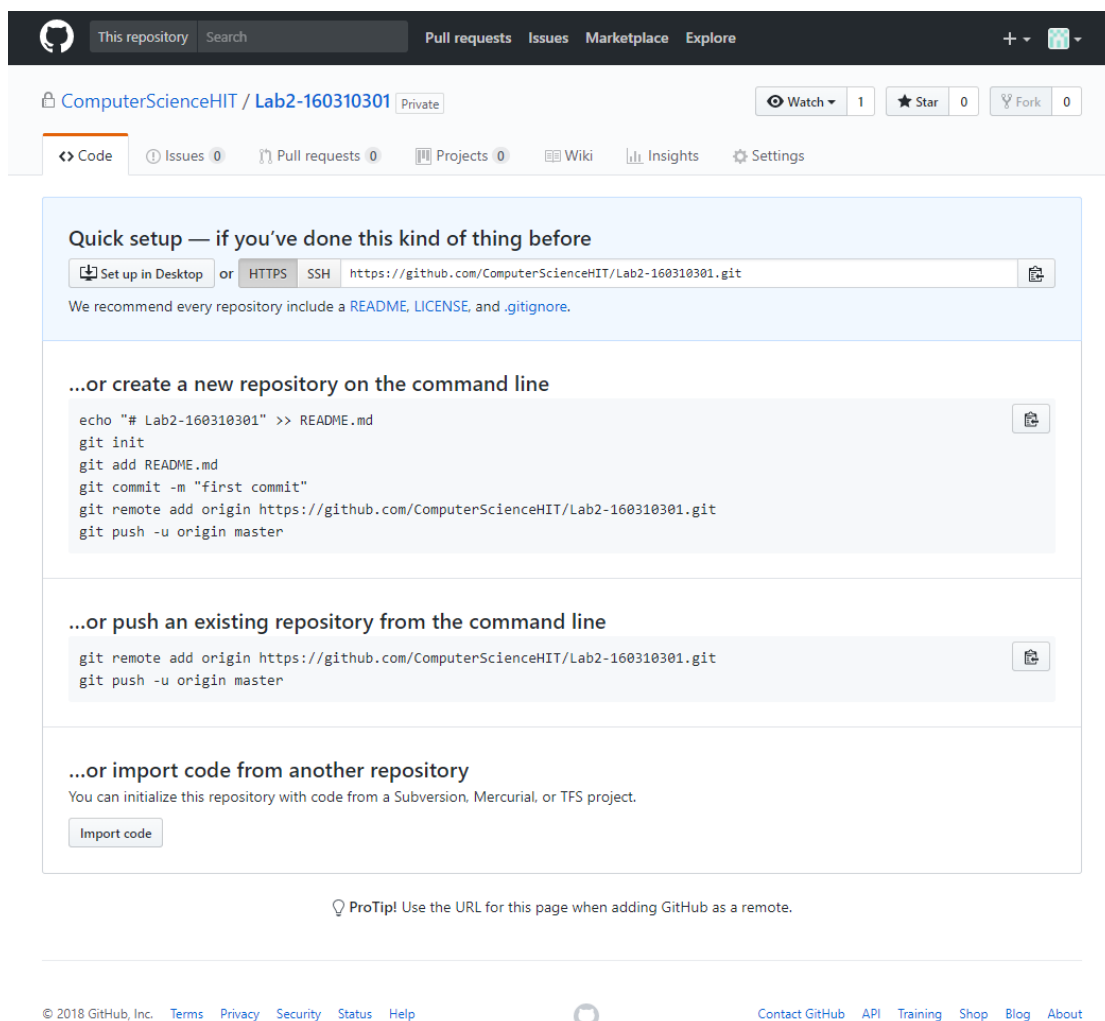


图 5

你对该仓库有完全的控制权，与自己在 GitHub 上创建的仓库完全一样。你可以在本次 Git 进行远程提交，本地 Git 连接该仓库可使用 https 或 SSH 的方式，具体配置请自行探索。

注意：该仓库是 **private** 的，即仅对自己和教师可见。请不要将各实验的代码发布到 **public** 的 GitHub 仓库。如果因为将自己的代码发布到 **public** 仓库导致被其他人抄袭借鉴，那么双方均无成绩。

3.3 本地 Eclipse 开发、Git 管理、提交至 GitHub 仓库

按照实验要求，在本地机器使用 Eclipse、Git 进行开发，本地 Git 仓库保存所有提交历史。在截止日期前推送至上述 GitHub 仓库。

各实验的仓库目录结构如下：

Labx-160310301	根目录
src	java 文件子目录，其内部可以包含子目录
test	JUnit 测试程序子目录，可以包含子目录
lib	程序所使用的的所有外部库文件，内部不分子目录

doc
...
实验报告
你自己设置的其他目录

注意：

- (1) 请使用 JDK 8。
- (2) 请遵循该目录规范，若不满足，教师评分使用的持续集成工具无法构建你的实验。
- (3) 在 Eclipse 中通过配置加入项目的各种外部库（通常是.jar 形式），本地构建时一般是通过 classpath 引入项目，但在提交 GitHub 仓库之前，请将代码中所有用到的 jar 文件拷贝至项目的 lib 目录，否则在 GitHub 上构建时无法成功。
- (4) 在提交至 GitHub 仓库前，请将实验代码从 Eclipse 环境脱离开来，自行使用 JDK、Ant (<http://ant.apache.org>)、Maven (<http://maven.apache.org>)、Gradle (<https://gradle.org>)等工具进行 build，或者在提交至 GitHub 仓库之后使用 Travis-CI (<https://travis-ci.org>) 进行在线 build，以确保提交后可通过自动化构建。
- (5) 提交到 GitHub 仓库时，构建程序会读取 master 分支。所以如果你在其他分支上开发，提交前请务必将最新修改合并至 master 分支。

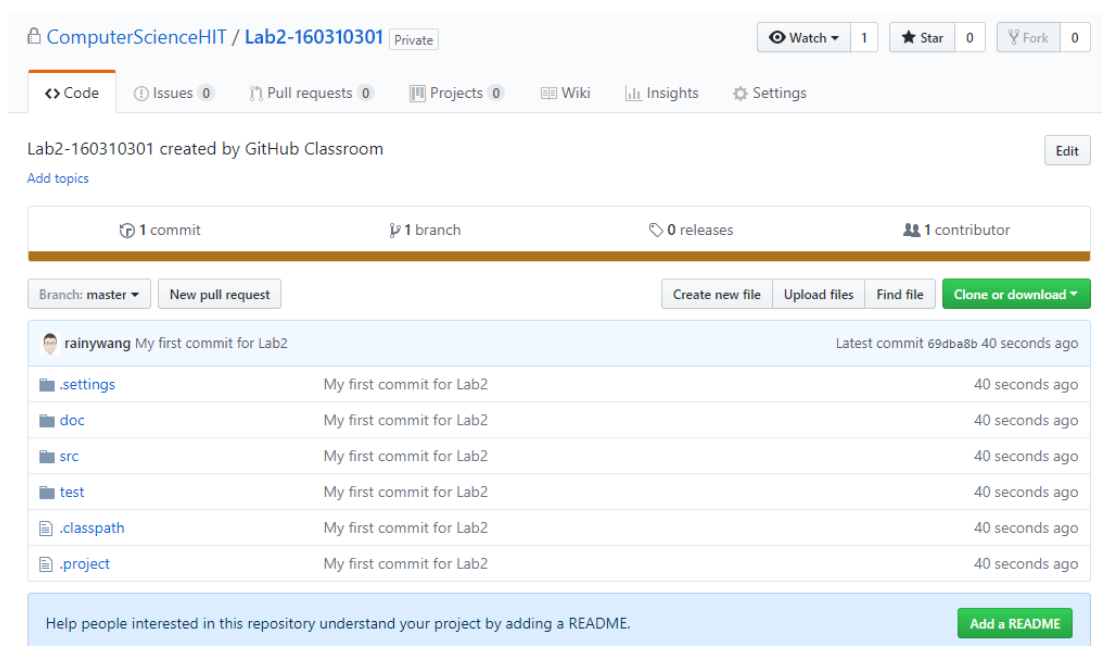


图 6

4 评分方式

在各实验的截止日期之后，教师可能会使用持续集成工具，对你的代码进行 build、执行 JUnit 测试用例、使用诸如 CheckStyle 和 FindBugs 等静态代码检查工具对代码规范进行检查。你的 GitHub 账号使用的 email 会收到构建报告。

注意：实验截止日期（周日夜里 23:55）之后，GitHub 仓库将不再允许提交。

若自动构建失败，分数为 0。若自动构建成功，根据 JUnit 测试报告进行自动评分。根据教师设定的多个测试用例（不向学生公开，在自动构建时执行）的结果，计算得到分数。CheckStyle 和 FindBugs 检查报告中的不符合代码规范或有潜在 bug 的缺陷，考虑其数量和严重程度，计入实验分数。

对某些实验，TA 通过人工测试的方式，对你的实验结果进行打分。

TA 阅读 doc 目录下的实验报告，人工打分。