



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 2020

## 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.....	1
3.3	本地 Eclipse 开发、Git 管理、提交至 GitHub 仓库 .....	7
4	评分方式.....	8

# 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=184>

# 2 实验环境

- (1) 请阅读 <http://web.mit.edu/6.031/www/fa18/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/fa18/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](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，或者直接使用之前已有的 GitHub 账号。

## 3.2 在 GitHub Classroom 中获取实验的 private repositories

GitHub Classroom 是 GitHub 为学校提供的教学环境，教师在这里布置作业，

为每个学生创建私有仓库，学生在仓库里提交代码，仓库对教师和 TA 可见。



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/z99eb2ot>

访问该地址，用个人 GitHub 账号登录，同意授权 GitHub Classroom。第一次授权之后，后续实验就无需再授权。

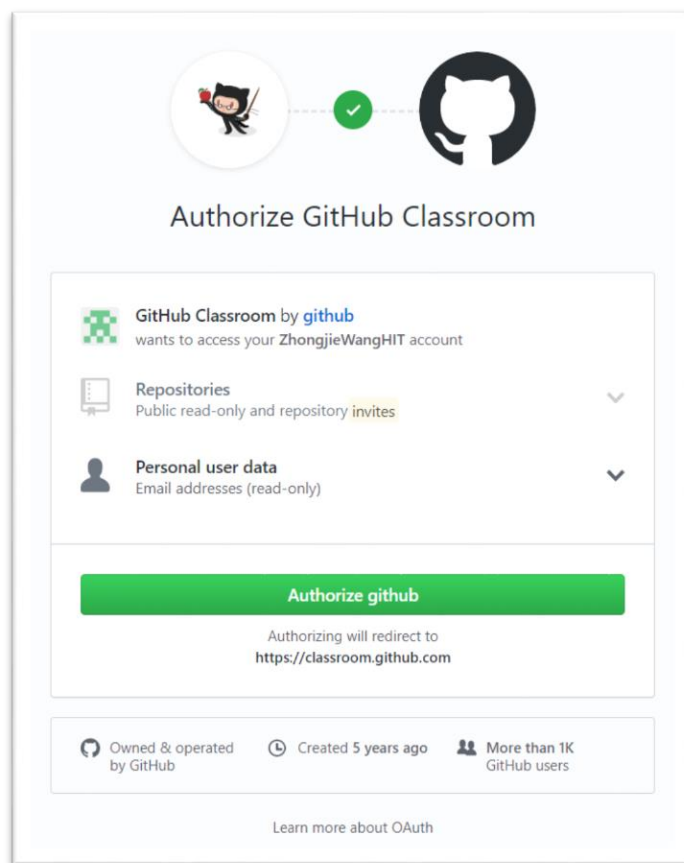


图 1

进入之后，出现以下界面，请在 **Identifiers** 区域选择你的学号，建立该学号与你的 **GitHub** 账号的关联。请不要使用 **Skip**，若在列表中找不到你的学号，请第一时间与教师联系。第一次绑定学号之后，后续实验无需再次绑定。

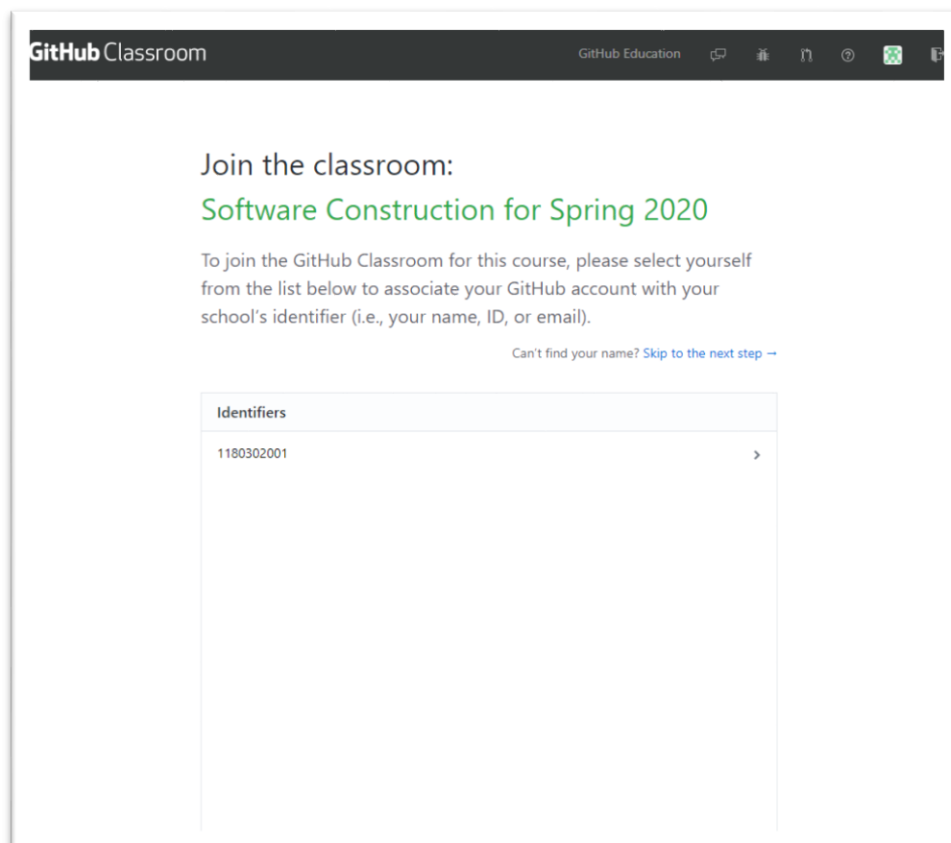


图 2

之后，进入图 3 界面，点击“Accept this assignment”，GitHub classroom 服务为你自动创建了用于提交此实验的 Git repository。进入之后，即可获得仓库的信息（见图 4）。

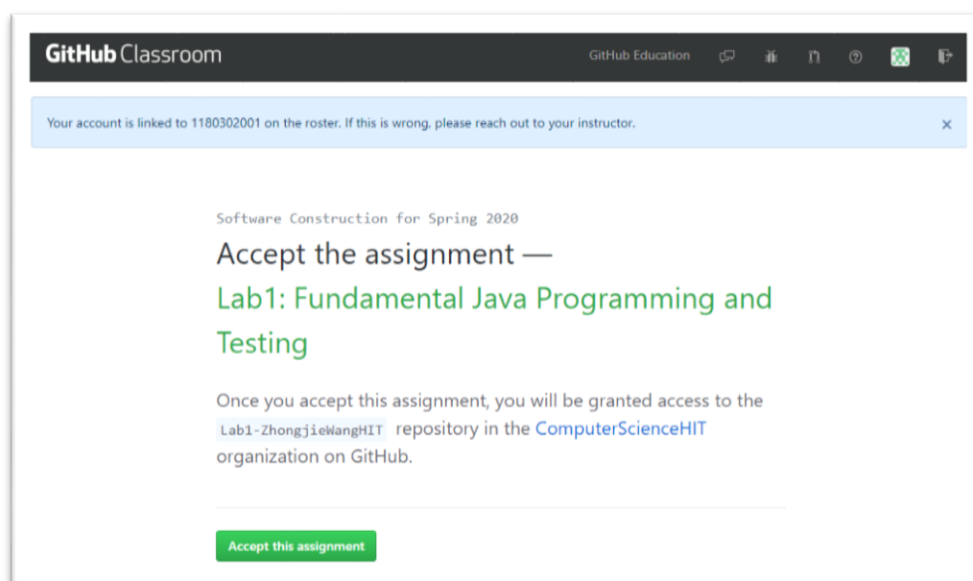


图 3

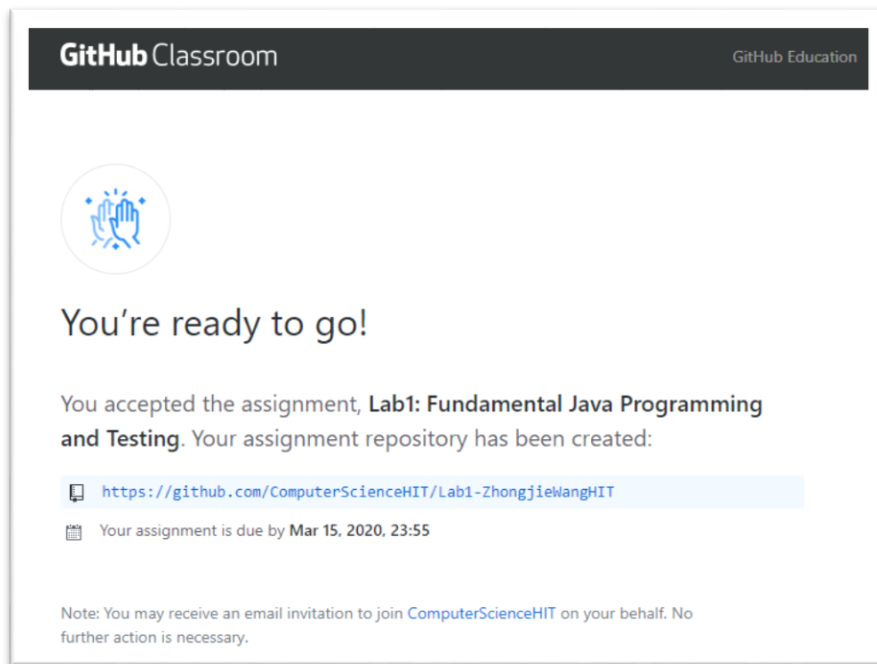


图 4

之后，你就可以进入该仓库。最初仓库是空的。

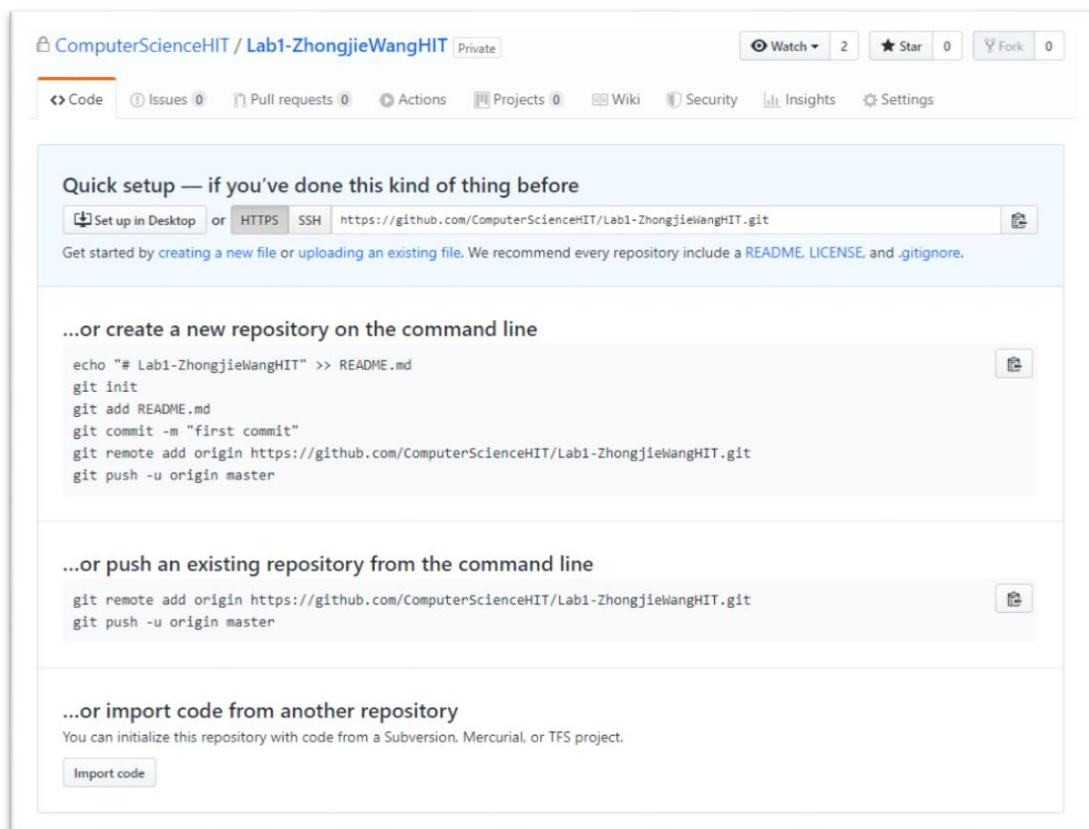


图 5

为了便于 TA 评分，请首先进入该仓库的 Settings 界面，将仓库名字修改为

Labx-1180302001 的形式，其中 x 为 1-4 表示实验序号，后面跟随的是你的学号（见下图所示）。请务必确保仓库名符合该规则，否则 TA 无法找到你的仓库，无法评分！

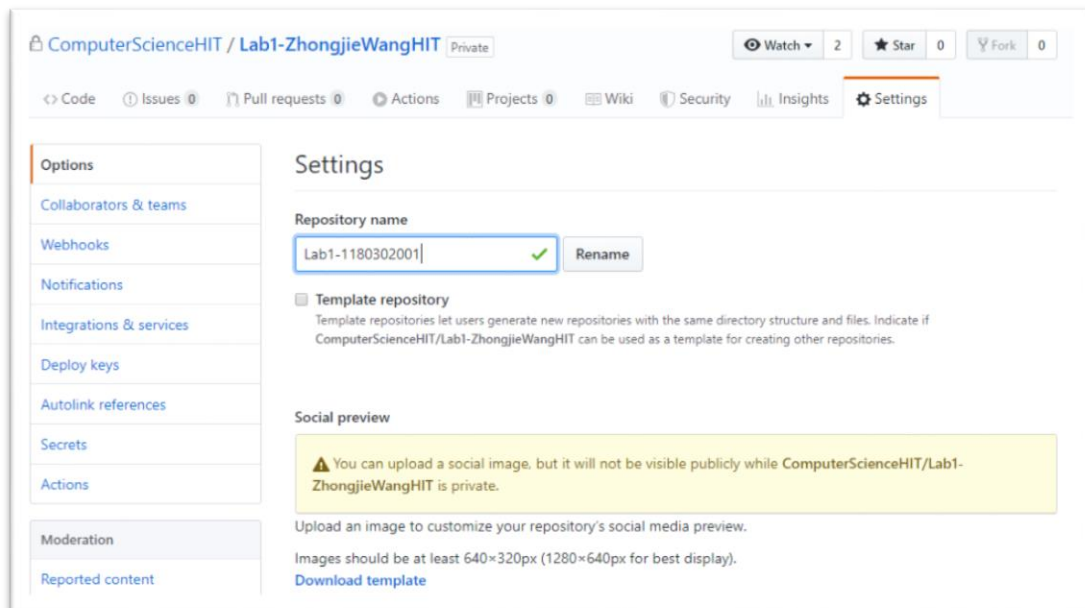


图 6

为了避免每个实验都要手动修改仓库名，你可以将你的 GitHub 账户的 username 改为自己的学号。

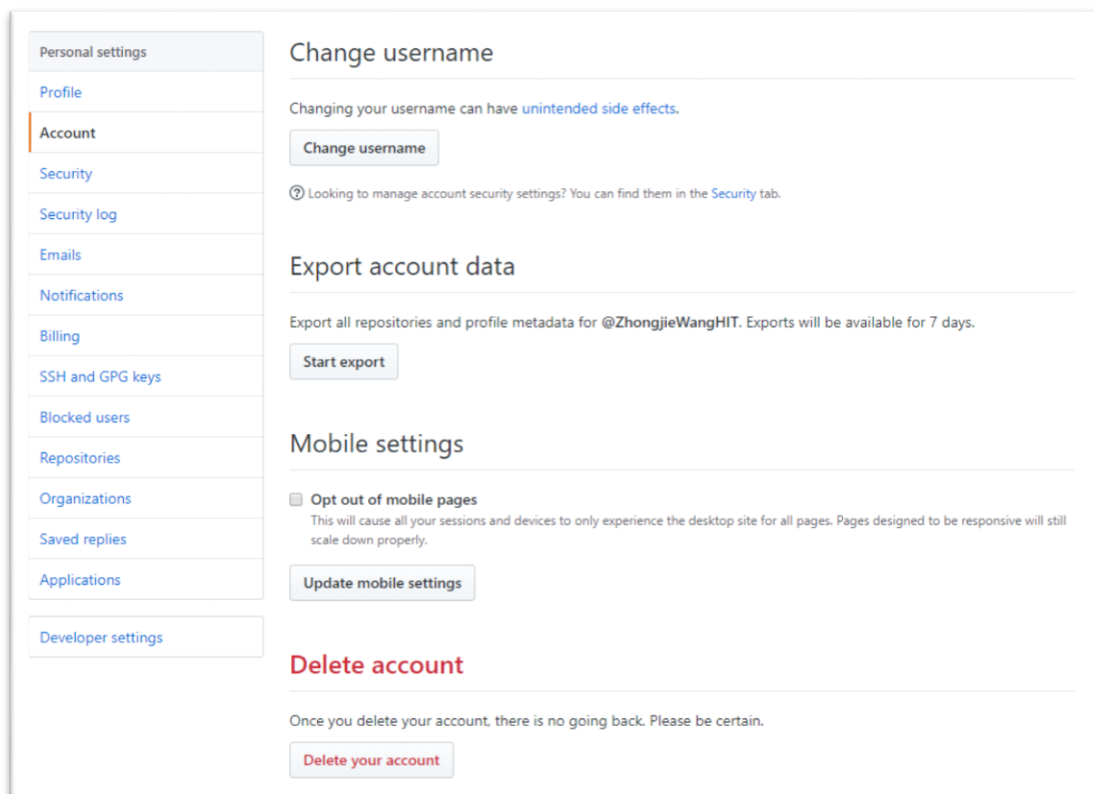


图 7



请务必确认你的仓库名符合如下图的规则。

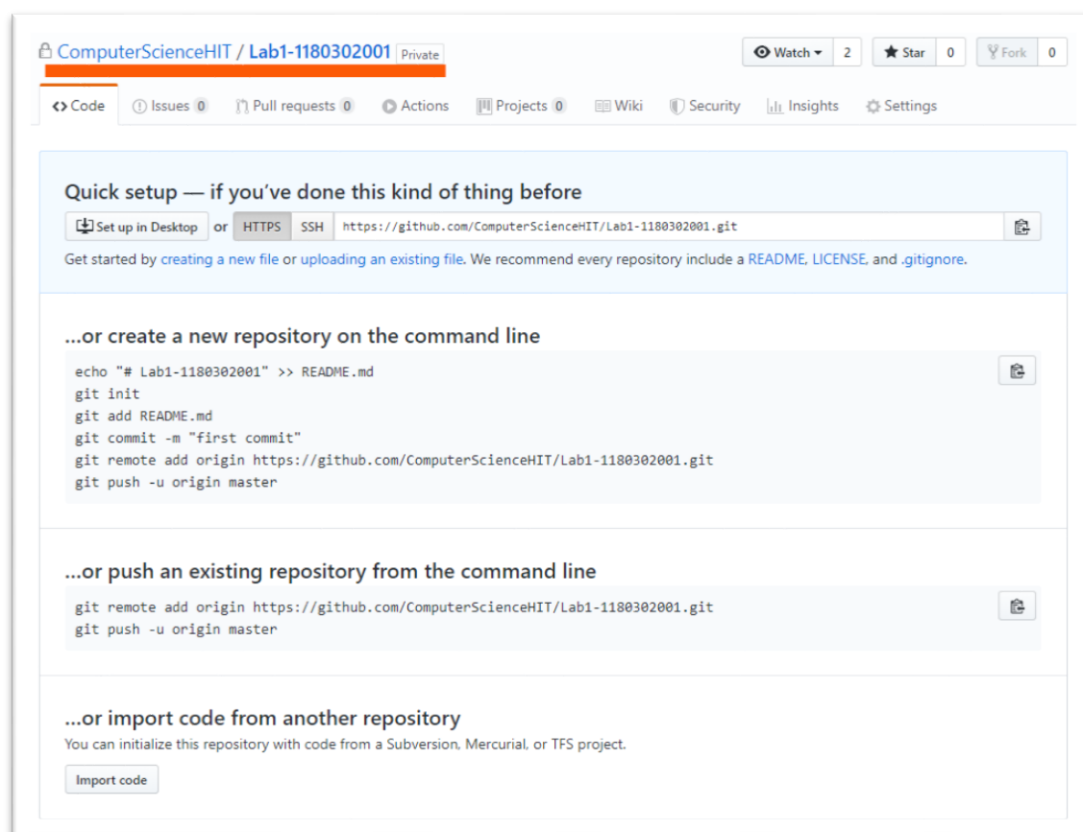


图 8

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，具体配置已显示在图 5 的说明部分，请自行探索。

**注意：**该仓库是 **private** 的，即仅对自己和教师可见。请坚决不要将各实验的代码发布到 **public** 的 GitHub 仓库中。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：

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、JDK 9 或 JDK10。不同版本的 Java 存在差异，有些实验内容可能受到影响。
- (2) 请遵循该目录规范，若不满足，TA 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 目录下，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，建议你自行使用 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。如果因为缺少某些库文件导致你的程序无法运行，TA 不再为其评分。
- (5) 提交到 GitHub 仓库后，TA 会读取你的 master 分支。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。如果实验要求你设置其他分支，也请务必将各分支指向正确的 commit，TA 根据分支名来获取你的代码。

## 4 评分方式

在各实验的截止日期之后，教师可能会使用某些工具对你的代码进行 build、执行 JUnit 测试用例、使用诸如 CheckStyle 和 SpotBugs 等静态代码检查工具对代码规范进行检查，再通过人工测试的方式，对你的实验结果进行打分。

此外，TA 阅读 doc 目录下的实验报告，对其进行人工打分。

**注意：**实验截止日期（周日夜里 23:55）之后，GitHub 仓库虽允许继续提交，但教师和 TA 仍以截止日期前的最后一次提交为准。