使用Github进行协作

学习目的

Git 是当今最好的版本控制软件，在软件工程中常被用来管理代码仓库（repository），对多人协作非常友好。

Github 是基于 Git 的现今最好的在线协作平台，几乎是所有程序员都会使用的项目或代码托管平台，大多数人选择在 GitHub上 公开自己的项目或者代码，在上面可以找到很多优秀的开源项目。

我们今后的作业交付都会基于 Github ，所以我们要先学会 Github 的基本使用。

自学任务

1. 注册一个Github账户。

- 通过 注册链接 创建一个 Github 账户，将用户名发送在班级群，并 @ 教练和辅导员

2. 创建一个自己的代码仓库，学会使用分支（Branch）和 Pull Request

1. 阅读参考资料1
2. 创建一个名为 helloworld 的仓库，并在仓库内创建两个分支 master 和 develop
3. 在 develop 分支中创建一个名为 develop.txt 的文件，提交一个 commit
4. 从 develop 提交一个 Pull Request 到 master 分支，并将其合并(merge)
5. 在master分支中再创建一个名为 master.txt 的文件，提交一个 commit
6. 从 master 提交一个 pull request 到 develop 分支，并将其合并

3. 在自己的本地电脑，安装 Git 桌面客户端可视化工具，用于管理远程仓库

1. 阅读参考资料 2，3

2. 通过 下载地址 安装 Github Desktop 桌面客户端
3. 在自己创建的 helloworld 仓库中找到 clone with https 的地址
4. 通过 Github Desktop 将 helloworld 仓库 clone 到本地仓库
5. 在本地仓库中创建一个名为 local.txt 的文件

> 请通过电脑自带的文件管理工具完成创建文件/文件夹，Github 桌面客户端没有创建文件和文件夹的功能

6. 将本地仓库新增的文件提交为一个 commit
7. 通过 Github Desktop 将本地电脑的变更推送到自己账户下的 helloworld 仓库

> 对于基础较好并且对 Git 操作和原理有所了解的同学，推荐安装 SourceTree
>
> SourceTree 和 Github Desktop 都是在本地电脑管理 Git 仓库的工具
>
> Github Desktop 更简单，非常适合没有基础的新人，SourceTree 相对更专业，比较适合有一定基础的学员

4. 在其他项目中贡献自己的代码

1. 阅读参考资料4
2. 将 公共作业仓库 fork 到自己账户下
3. 向辅导员或教练确认自己的班级目录，如 19100101
4. 将自己账户下的作业仓库 clone 到本地电脑
5. 在本地仓库班级目录下，创建一个用自己的 Github 用户名命名的文件夹（这个文件夹是将来写作业的地方），并在自己用户名的文件夹下创建一个名为 README.md 的文件（这个 README.md 文件可用来记录分享自己的学习心得）
6. 在自己用户名的文件夹下创建一个名为 d1_exercise_helloworld.txt 的文件，在文件中写入任意内容
7. 将本地仓库关于本次作业的变更提交为一个 commit
8. 通过 Github 桌面客户端将本地电脑的变更推送到自己账户下的作业仓库
9. 回到 Github 页面，在自己账户下的作业仓库页面向远程公用作业仓库的 master 分支发起Pull Request, 在提交的 Pull Request 中 @自己的教练 提醒他检查作业

5. 在 Github 的仓库中提交 Issue

1. 阅读参考资料5
2. 在自己创建的 helloworld 仓库中提交一个 issue

3. 为自己创建的 issue 打上任意一个标签

参考资料

1. Github 官方写给新人的教程
2. 如何使用 Github Desktop Clone 仓库
3. Github Desptop 官方教程
4. Github Forking 功能介绍
5. Github Issue 功能介绍

拓展学习（可选）

1. Git 手册
2. Octotree - Code tree for GitHub
3. How-To-Ask-Questions-The-Smart-Way
4. 在 SourceTree 中提交，推送，拉取变更记录