|  |
| --- |
| **一、实验目的**  1.创建仓库；  2.增加合作者；  3.添加 issue。 |
| **二、实验设备（或软件平台）**  1．Windows笔记本  2．Git软件  3．GitHub网站 |
| **三、实验内容及步骤**  **1.1 创建标签**  开始操作。首先版本，克隆仓库、配置信息、查看提交历史：      给当前分支当前版本创建一个标签：    这样一个本地标签就创建完成了。  **1.2 查看标签**  执行 git tag 命令显示仓库中的全部标签列表，执行 git show [标签名] 查看标签详情：    **1.3 删除本地标签**    **1.4 将本地标签推送到远程仓库**  首先对两个提交版本创建对应的标签：      执行 git push origin [标签名] 推送标签到远程仓库，注意前面的命令都只涉及本地操作不需要联网，此命令需要联网：    浏览器上打开仓库主目录，点击可以查看 releases 和 tags  点 Tags 按钮查看标签：      查看远程仓库：    **1.5 删除远程仓库标签**  如果标签废弃不用或者写错了，可以使用 git push origin :refs/tags/[标签名] 删除远程仓库的标签，命令中的标签名其实也就是文件名：    再次查看远程仓库，已经没有001了：    好，删除成功。以上就是关于 Git 标签的创建、查看、推送、删除的操作流程。  查看本地仓库的标签列表：    001还在，需要手动删除。  **1.6 签出版本**  首先执行 git checkout [标签名] 切换到之前的某个提交版本，然后执行 git checkout -b [新的分支名] 将此提交版本固定到一个新分支上并切换到此分支：    这样就利用标签完成了提交版本签出的工作。 |
|  |
| **四、实验总结**  答：本次实验为进行有效的版本控制和团队协作提供了宝贵的实践经验，为未来的软件开发工作奠定了坚实的基础。通过这些技能的学习和应用，可以提高代码的可维护性和团队的工作效率 |