西南民族大学

**实验报告**

**2019------2020**学年第**2**学期

课程名称：软件工程

学院：计算机科学与技术 专业：软件工程

年级：2019级 班级：1902

学号：201931101064 姓名：黑起华

同组人：无

|  |
| --- |
| 西南民族大学学生实验报告  教学单位：计科学院 实验室名称：BS-222 实验时间：2021 年 9 月 29 日  姓名：黑起华 专业：软件工程 班级：1902 学号:201931101064 |
| 实验项目名称：SE实践1 实验成绩： 教师签名：周绪川 |
| **一、实验目的**  1.通过自主学习配置实验所需环境。  2.安装git工具且申请账号，通过网络自主学习其基本使用方法。  3.回顾学习UML图并可熟练画出UML对象图及时序图。  **二、材料与方法**  PC机一台、GitHub、ProcessOn在线作图工具等 。  **三、实验主要过程与结果**  1.配置实验环境  （1）语言要求  C/C++、JAVA等。  （2）开发环境  内存8G及以上。  （3）IDE  选择使用IDEA。  （4）数据库  选择使用MySQL。  （5）辅助工具  选择ProcessOn在线作图工具。  （所需工具在此之前都进行过配置环境并安装，因此在此不进行赘述。）  2.安装git工具并进行自主学习。  （1）在其官方网站上寻到下载地址，并点击下载。  （2）下载完成后根据提示进行安装。  23  图1.1 阅读信息 图1.2 选择安装路径  45  图1.3 选择组件 图1.4 选择在开始菜单的文件夹  67  图1.5 调整路径环境 图1.6 配置行尾转换  8 9  图1.7 将终端模拟器配置与Git Bash一起使用 图1.8 额外的配置选项  10  图1.9 完成安装1 图1.10 完成安装2  （3）申请账号完成后进行登陆。    图1.11 成功登陆  （4）学习github的基本使用方法  ①创建自己的一个仓库  A.在页面的最右上角，头像的左边，点击“＋”，在弹出的下拉选项中选择“New repository”。  B.在仓库名称输入框中输入仓库名称：“mycode”并写一个简短的描述。  C.勾选“Initialize this repository with a README”，点击“Create repository”按钮，完成创建。    图1.12 创建仓库1    图1.13 创建仓库2  ②新建一个子分支  A.进入你的“X-A”仓库。  B.点击文件列表上方的写着“branch: master”的下拉框。  C.在输入框中输入新的分支名字“Anna-edits”。  D.点击下方蓝色背景的分支创建框或直接按键盘的“Enter”键。    图1.14 新建子分支editor    图1.15 成功新建子分支editor  **③做出修改并提交**  **A.点击“README.md”文件，切换到“README.md”内容页面。**  **B.点击编辑框右上角的铅笔按钮，进入编辑界面，在编辑框中，写一些自己随意编辑的信息。**  **C.在下方的“提交更改”框中，输入你做出此次修改的备注说明。**  **D.点击下方的“Commit changes”按钮。**  **（这些更改只是针对于位于“Anna-edits”子分支中的 “README.md ”文件，所以现在这个分支上包含的内容和主分支上已经有所不同。）**    图1.15 进入README.md    图1.16 做出修改    图1.17 提交修改  **④为更改的README文件发出“请求代码合并”请求**  **A.点击“Pull Request”选项卡按钮，切换到请求代码合并页面，点击绿色的“New pull request”按钮。**  **B.选择你创建的“Anna-edits”分支，与主分支进行比较。**  **C.在对比页面查看这些更改，确定它们就是你想要提交的。当你确定这些更改就是你所要的时候，点击绿色的“Create Pull Request”按钮。**  **D.**为更改做一个简单的标题和描述，当写完备注描述之后点击“Create pull request”按钮。    **图1.18请求代码合并页面**    **图1.19 创建的子分支与主分支比较/查看更改**    **图1.21 为更改写标题及描述**  **⑤合并你的“请求代码合并”请求**  **A.点击绿色的“Merge pull request”按钮，将这些修改合并到主分支。**  **B.点击“Confirm merge”按钮。**  **（由于这个分支的更改已经被合并了，所以还需要点击紫色框中的“Delete branch”按钮，删除这个分支。）**    **图1.22 将修改合并到主分支**    **图1.23 删除子分支**  （5）将仓库连入本地文件夹下  ①在D盘建立了一个Test空文件夹，双击进入文件夹。    **图1.24 建立Test空文件夹**  **②双击打开桌面上的Git Bash图标，使用命令行进入项目所在目录。**  **cd D:/Test（**或者先进入所在目录，然后右击，在所弹出的操作列表里点击Git Bash Here**）**    **图1.25 进入项目所在目录**  **③复制自己的git链接，并以git clone https://github.com/MASKok/MASKok.git的形式输入到命令框。**    **图1.26 将仓库连入本地**    **图1.27 查看Test文件夹**  （test文件夹下面多了一个项目的文件夹，里面的文件就是我们在github上的所有文件。）  （6）进行简单测试并建立远程仓库  ①进行用户设置    图1.28 用户设置并测试  ②ls命令    图1.29 ls命令测试  ③建立远程仓库  A.寻找本地SSH keys值。  sss  图1.30 寻找SSH keys值  B.在网页版中点击进入设置settings并找到SSH and GPG keys选项。    图1.31 找到SSH and GPG keys选项  C.进入后点击网页右上角的New SSH keys，并输入密钥名称和密钥（将.pub后缀的文件中的内容复制） 。    图1.32 输入密钥名称和密钥  a  图1.33 图1.34 成功建立新的SSH  （点击Add SSH key后出现了如图1.33所示的情况，将登陆密码输入即可。）  D.在Git Bash中输入ssh -T git@github.com检测是否成功绑定。    图1.35 成功绑定  E.进行验证。  1）在本地文件/D/Test/A-X文件夹下创建测试.txt。    图1.36 创建测试.txt  2）在Git Bash中进行验证。    图1.37 验证成功  3）输入git add 测试.txt之后输入然后git commit -m "cc" ,引号内的内容可以随意改动，这个语句的意思是给你刚刚上传的文件一个备注，方便查找记忆。    图1.38  4）输入git push origin master会出现如图1.39所示的情况，将用户名及密码输入。  2  图1.39    图1.40  出现如图1.40所示该情况的话，就是登陆失败了，这时候需要输入GitHub的账号名称然后输入密码即可。    图1.41 再次输入登陆密码    图1.42 上传成功    图1.42 上传验证成功  3.用UML工具画出我校每人每天体温测量上报系统的一个对象图和一个时序图。  （1）UML对象图  体温提交对象图体温提交对象图  （2）UML时序图  体温提交时序图2  4.个人Git链接：https://github.com/MASKok/201931101064-1  **四、分析讨论**  1.通过自主学习逐步熟悉了Git工具并学会基本的使用方法。  2.Git命令  （1）git pull命令  用来拉取该分支内容，与本地做为对比，防止冲突。  （2）git add命令  用来将所要更新的文件，添加到git队列中，以备后续的提交。  （3）git commit -am命令  用来做提交准备，-am后面跟上注释，可以让其他人看到本次为什么做出更改，此时并没有提交上去。  （4）git push命令  真正地将所更改的文件提交到分支中。  3.通过自主回顾UML图的分类，再次熟悉UML类图、对象图以及时序图。  4.通过对本章节的实验让我初步的队git工具和GitHub的使用犹豫了初步的了解。  **五、教师评阅** |

注：实验报告的内容及格式可由学院根据学科专业特点确定；全校各专业必须使用学校统一封面。