开发环境统一配置

I. 旧环境卸载

删除旧环境的目的是消除其对新环境的影响，有旧环境的存在，往往会对新环境造成不可预测的影响，例如不可安装、隐藏的祸患等等。

旧环境主要包含三个 eclipse、Java、MySQL。Java 和 eclipse 的删除都非常简单，控制面板中如果找得到，就在那上面删除，如果找不到直接删除整个文件夹。MySQL 这个的删除就要稍稍复杂一些，如果安装位置有卸载的可执行文件，名字可能是 uninstall.exe，点它就卸载完毕。再例如通过 MySQL 安装软件安装的（和 VS 那个安装软件差不多的那个），打开那个软件就可以卸载。还如果在控制面板能找到，那当然也能直接删除。但往往多数情况下都是自解压安装的老版本，例如 5.7，这样卸载会麻烦一点，只有一种办法，那就是直接删除整个 MySQL 的数据文件（注意你的 data 文件夹，那个是你的数据文件，如果需要请备份——将整个目录拷到其它位置，之后在改一下 my.ini 的配置文件，就可以使用你之前的数据文件了）。直接删除整个文件夹，依然会在服务列表中看见 MySQL 这个选项，这就说明没有删除干净，这可能会导致安装新版本 MySQL 数据库的时候提示已安装，无法正常安装。彻底删除按如下顺序进行不会出错，其时顺序不对并无大碍：

1. 删除注册表信息：regedit，找到\HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\services\eventlog\Application\MySQL 和\HKEY_LOCAL_MACHINE\SYSTEM\ControlSet002\services\eventlog\Application\MySQL。（如果没有跳过这一步）
2. 删除 MySQL 服务，cmd 需要管理员权限运行如下命令：sc delete mysql;。（delete 接服务名字，可一般都是默认的，所以都是 mysql）
3. 删除整个 MySQL 文件夹。

II. 安装及配置 git

安装 git 相当简单，运行那个 exe 文件就行了，除了修改安装位置，其他全部默认，接下来说一下它的配置（关于使用就略过了，因为用的是 github 客户端，其实它就是在执行一些 git 指令，只是可视化了而已）：

1. 设置用户名和邮箱（建议用户名和 github 上面的一样，邮箱必须一样）：直接桌面或者其他位置右键，点击那个 git bash 的位置，之后出现一个类似 cmd 的窗口，在上面输入两条指令，设置你的用户名和邮箱：git config --global user.name "username"和 git config --global user.emil "email@xxx.com"（邮箱得和 github 上的一样，用户名建议一样；--global 是全局的意思，也就是每个仓库的设置都是它，每个仓库可设置不同用户名和邮箱，global 一劳永逸）。
2. 获取 ssh 密钥，并将其发到群上，这一步的用处是让你能够向我的仓库中 push 代码，不进行这一步似乎是没办法 push 的（之前小学期没配置也行，所以我迷糊了，就配置一下吧）。获取 ssh 密钥的指令也在刚刚那个 git bash 的窗口里输入，如下：`ssh-keygen -t rsa -C "email@xxx.com"`（连按几个回车就行了，邮箱和 `github` 上的一样）和 `cat ~/.ssh/id_rsa.pub`（查看产生的 `ssh` 密钥，将其复制发到群上然后@我就行了）

III. 安装及使用 github 客户端

安装 github 客户端相当简单，点击 exe 就行了，重点在如何使用：

1. clone 将远端仓库复制到本地。这个操作简单，远端的地址为：https://github.com/SoftwareDevelopmentPractice/HelpStudent.git（或者 git@github.com:SoftwareDevelopmentPractice/HelpStudent.git），记得本地项目位置，在那个位置会有一个和仓库一模一样名字的文件夹。

2. merge、fetch、pull、commit、push 五个基本操作。

   merge：就是本意，合并，即集合间取并操作，通常指两种：一是分支之间的合并，而是远端仓库和本地仓库之间内容的合并。

   fetch：直接获取远端仓库中的内容到本地仓库，不会进行 merge 操作。

   pull：将远端仓库的内容获取到当前工作空间，即获取内容之后直接进行 merge 操作，即将远端不一样的内容直接添加到当前工作空间。

   commit：也是本意，提交。将当前修改内容提交到本地的 git 仓库中，clone 之后会有一个隐藏的文件夹.git 这个就是本地的 git 仓库，每次修改内容都需先提交，提交的时候写清楚那次提交是做了什么，关于这个是由两个内容组成——摘要和描述，务必每次提交写清楚，弄明白。

   push：推送，即将已提交到本地仓库的改变推到远端相应的分支中，没有 commit 的改变是不会提交到远端分支中的，每次 push 之前都会先 fetch，保证修改之前本地与远端并无不同，如果有不同就必须先 pull，pull 之后才能够 push。

3. 分支开发：master 是主分支，其余的为其他分支，各自在自己的分支上开发，每一次首先将主分支的内容合并到自己的分支，在自己的分支进行修改，每次再将修改推送到自己的远端分支中。github 客户端做得很好，界面上很容易找到 current branch 这个按钮——用来切换 分支（在 fetch、push、pull 操作按钮的旁边），一开始，先将分支切换到 master，然后，在点击 current branch 旁边的按钮，最开始应该是 fetch origin 的字样，点击它之后，有不同就会变成 pull 的字样，点击 pull，这样就将远端主分支内容更新到本地的 master 分支了。接下来，将分支切换到你的工作分支，如果是苟光耀，那么就选择 GouGuangyao。然后，界面最上方菜单栏上有个 branch，点击，选择 merge into current branch，之后选择 master，再点 merge 就行了，这样就把 master 分支的内容更新到你的工作分支了。接下来你就进行更改，更改之后 github 客户端 commit（写好摘要和描述），最会点击菜单栏下方的一栏的右上角的 push（current branch 旁边），这样你的更新就能在远程的相应分支能看见。

4. 首次 clone 之后的做法（针对我们这个项目）。客户端切换到你的分支，然后点开 eclipse，创建项目，选择项目位置，就选择之前 clone HelpStudent 仓库的位置的文件夹。创建时直接更改 eclipse 的项目，不要去写项目的名字，当选择那个 HelpStudent 的文件夹的时候名字会自动填上去（注意当 clone 的时候，会创建一个和仓库一样名字的文件夹，文件夹里的内容就是仓库里的内容），项目创建好了，之后创建一个测试文件的类放在 Test 包内，和我的名字取得一样，然后尝试以上的步骤将其 push 到远端，切记写明白为什么提交。

5. github 上的合并操作（这个我可以来做，但你们也需要了解，毕竟在一起学习）。登录网站，跳到 HelpStudent 仓库，如果上面有 compare&之类的字样，点击就可，如没有，点击 New pull request，之后左边选择 master，右边选择自己的分支，如果自己的分支和 master 有不同，就会

出现 create pull request，直接点击，之后会出现 merge pull request，下面还有个对话框用于交流，交流确认之后，点击 merge pull request，然后 confirm merge，这样你的修改就会同步到 master 分支，如果出现冲突（同一个文件，同时，同一行有不同），需要解决冲突，然后才能 merge，否则会失败，解决冲突，点一下解决，就会出现冲突的地方（冲突的位置，会一上一下排列好，删除不对的，或者综合情况修改，总之就是只留下对的。<<<<<这个标识下方就是冲突的地方，只要冲突两个文件的同一位置都会出现，删掉或修改，即<<<<<删掉，留下对的内容），根据具体修改就行了 。

IV. 安装 jdk13

1. 安装。简单双击那个 exe，修改位置（记住），之后一切默认。

2. 配置环境变量，自动 jdk11 之后便不再有 jre 文件夹，已经将其集成到 jdk 里了所以配置环境变量不需要配置 jre 的部分，此外 CLASS_PATH 的环境变量也不再需要，故目前只需要 path 中添加 jdk 安装目录下的 bin 文件夹就行了。当然也可以配置 JAVA_HOME: ｊｄｋ安装目录，之后在 path 中添加 %JAVA_HOME%\bin;就行了。

V. 安装及配置 eclipse 插件

1. 安装。相当简单，直接解压 eclipse 到你想安装的位置，再创建一个桌面快捷式拖到桌面即可。

2. 配置插件（插件每次安装好，都建议重启 eclipse）。

- 安装 Eclipse Marketplace。help→install new software，输入以下地址：http://download.eclipse.org/mpc/indigo/（如果有时没有开始搜索，地址那里按一下回车），选择 EPP 那个，接下来应该都知道了。
- 安装 Java EE 插件。help→install new software，输入以下地址：http://download.eclipse.org/releases/oxygen，选择最下面那个 web 的选项。
- 安装阿里巴巴代码分析插件。help→install new software，输入以下地址：https://p3c.alibaba.com/plugin/eclipse/update。
- 安装 java 13 插件。help→Marketplace，在搜索框输入 java 13，点击 go，搜索，找到 java 13 support for……（名字太长），install。
- 安装 Window Builder 插件。help→Marketplace ，在搜索框输入 Window Builder，点击 go，搜索，找到 Window Builder 1.9.1，install。
- 更新 eclipse。help→Check for update，更新完之后重启。

3. 其他 eclipse 相关配置

- 工作区的配置。将 org.eclipse.core.runtime 文件夹复制到 eclipse 工作区(开始有目录)/.metadata/.plugins 文件夹下（可能.metadata 文件夹被隐藏了）。这一步的目的在于同步类模板、按键触发、tab 键的解释等设置，如果不想每次都去修改一下类创建时产生的名字（自己创建的类用自己的名字），如下操作：Window→Preferences→java→Code Style→Code Templates，之后右边窗口，code→New java Files→edit 修改@author 之后的内容，改成你的英文名字（汉语拼音），ok、apply close。

- 防空格补全。解压那个防空格补全的压缩包，里面是我修改过后的 jar 包，打开 eclipse 安装位置，将压缩包里的那个 plugins 文件夹直接复制到 eclipse 目录下（那个目录有一个 plugins 的文件夹）它会提示有重名文件，没错，直接覆盖它。

VI. 安装 MySQL 8.0.17

1. 安装。首先双击那个 msi 文件，安装好安装 mysql 的软件，之后运行那个软件，除了修改安装位置，一切默认。它会安装一些附加的软件，我们得装那个 MySQL WorkBench（可视化的数据库软件和 nacivat 是一个软件性质的），其它的我不知道有啥用，如果不默认，可以选择不装。安装过程可能会有一个弹窗，大致意思是没检测到什么软件，询问是否还要安装，总之就是安装，具体是什么我不太记得了。

2. 目录讲解（以未修改安装位置为例，好像修改了目录只会修改其它软件安装位置，即接下来会说的第二点的位置）。通过这种安装软件的安装方式，整个 mysql 文件会分布到很多其它位置（不像解压安装在一个文件夹下）。接下来简单讲解一下 mysql 的目录结构分配：

- data 文件夹及 my.ini 文件在 C:/ProgramData/mysql 文件夹下（ProgramData 文件夹是 C 盘中的一个隐藏文件夹）。单独放出这两个东西，是因为它们很有用处，之前叫备份的那个 data 文件夹可以改个名字，和这个 data 文件夹放在同一个目录下，然后修改 my.ini 的配置文件信息，把那个 datadir 等于后面改成你以前那个 data 文件夹的目录（如果已经移动在了新 data 文件夹的同一目录，只需改个名字），这样重启 mysql 就能使用之前的数据了。
- 其它文件分布在 C:/program/mysql 和 C:/program(x86)/mysql 文件夹下（未更改默认安装位置）。program(x86)/mysql 里面有啥我真不清楚，program/mysql 文件夹下放着 mysql 必要组件，还有一些安装是选择的其它软件，例如上面那个 mysql workbench，此外还有一些其它东西。

至此大家的环境配置完毕，有任何问题联系我，其实我更希望的是你们自己动手，这样也可减少我的工作量，也能让你们学会一些东西。

大家多多联系这些软件的使用，祝我们软件开发顺利，能够体验到一次真正的团队软件开发。