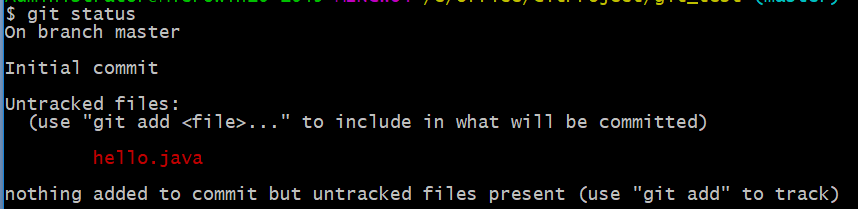
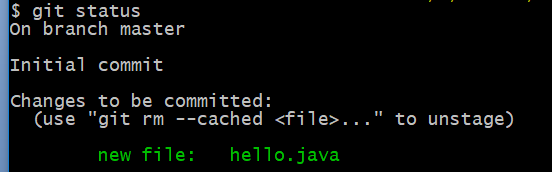
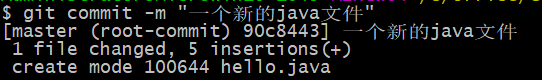
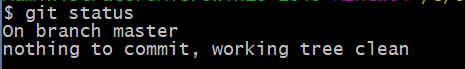
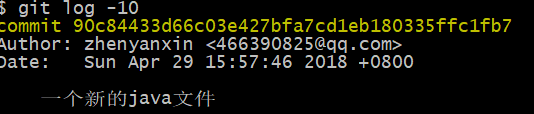
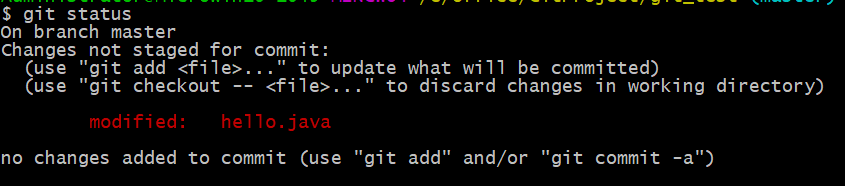
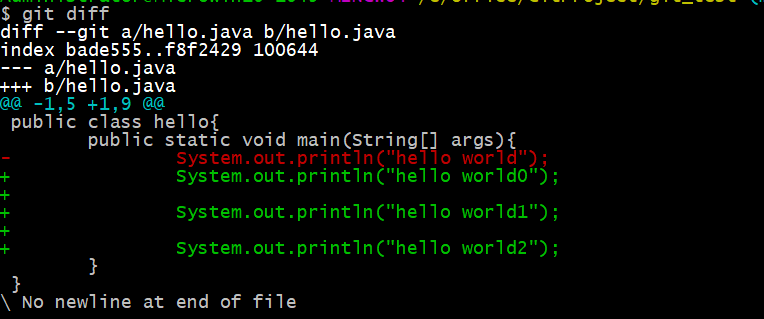
知识点：

1. 掌握GIT工具基本操作流程
2. 理解GIT中工作区、暂存区、主分支的概念
3. 利用GIT实现版本回退功能
4. 利用GIT实现文件的修改操作

具体内容：

1. 设置开发者的个人信息
   1. 一个系统可能存在多个开发者，在GIT中每位开发者需要进行配置GIT的全局信息
   2. 配置全局信息方法：
      1. git config –global user.name “zhenyanxin”
      2. git config –global user.email “466390825@qq.com”
   3. 设置之后不会有任何提示信息，而后用户可以用以下命令查看配置信息
      1. git config –l
      2. 此时这台电脑的所有做出的修改，都会带着用户名和邮箱将内容记录下来，发送到主程序中（相当于日志）。
   4. 创建仓库,初始化仓库:
      1. git init
      2. 如果要开发项目，必须先要有一个仓库
      3. 创建文件夹——>初始化（用i代码）
   5. 添加文件（尽量用UTF-8编码）
      1. 创建一个文件
      2. 查看仓库状态：git status
      3. 通过仓库状态可以看出当前开发处于主分支状态，初始化仓库提交，未标记的文件，随后给出了一些操作的命令信息，未标记文件的列表
      4. 加入暂存库中：git add 文件名
      5. 文件加入暂存库后并没有真正的提交到仓库之中（主分支上）
      6. 提交文件信息（每次更新提交的时候一般都会为其增加一些注释信息使用时使用-m添加注释）：git commit –m “注释信息”
      7. 此时文件已经成功提交到主分支上了，意味着发布成功了。此时么有需要提交的内容了工作目录很干净， git工具也会记录每一次的提交日志
      8. 查看文件的日志信息：git log [文件名] –条数
      9. 如果到底了直接按q图书查看，从上到下信息号，提交者，提交时间。提交前一定要用add进行添加再进行commit，如果不想分两步直接commit的话可以直接使用commit加上-a这个参数
   6. 修改仓库文件
      1. 程序代码的出现就一定会面临修改，git可以控制版本中文件的修改
      2. 现在git直接提示用户，文件没有保存在暂存区之中，而且提示有：要么选择文件暂存，要么直接进行文件恢复，同时给出了已经修改的文件hello.java 
      3. 查看文件修改前后的区别:git diff [文件名] 
      4. 通过代码颜色可以清楚的发现只要是修改了代码都能记录程序员的操作
2. 工作区与仓库
   1. 工作区就是当前电脑的操作目录（包括.git）
   2. 仓库：工作区有一个隐藏的目录.git，这个不算工作区，而是git仓库
   3. git的版本库里面存了很多东西，其中最重要的就是称之为stage的暂存区，还有git为用户自动生成的的主分支master，一级master的HEAD指针
   4. 工作区的文件通过git add将代码提交到暂存区当中，数据保存在暂存区之后git commit进行代码提交到对应分支上，把暂存区的代码提交到分支后，会自动清除暂存区的内容，那么此时再次查询状态，会直接发现没有任何文件修改内容。
3. 版本回退
   1. 每当用户进行代码提交的时候都会生成一个commit id，而这个commit id就是版本回退操作的主要操作方式。
   2. git log –pretty=oneline可以发现所有的commit id并不是顺序的编号，而是系统自动生成的16进制数据，这一概念就和seeson id性质上相似，以防止版本号冲突。
   3. 在分支上会有一个HEAD，这个指针永远指向最后一次提交的位置。若需要回退的话可以直接操作它。
   4. 当使用会退后发现HEAD指针发生改变，如果回退一步，之前的操作不会被删除，但是所有的代码将回归到指定的状态。
   5. 回退一步代码：git reset –hard HEAD~1
   6. 如果还想回到最新状态，又需要找到最新的COMMIT ID
   7. 查找所有的仓库修改操作日志：git reflog
   8. 通过commit id恢复之前状态：git reset –hard [序列号（commit id）]
   9. 此时代码就恢复到commit id的相应状态
4. 文件修改问题
   1. 在有了暂存区和分支的概念之后，就需要回避一个为不同，只有保存在暂存区的文件才能进行相应的修改。
   2. 进行代码修改后对比可以通过 git diff HEAD [文件名]
   3. 撤销修改：代码修改很正常，很多时候修改后发现代码不应该修改
      1. 情况一：工作区代码没有添加在暂存区的时候，要恢复到原来的状态很容易，直接可以在git status里面看到相应的状态。
      2. 处理方式：git checkout - - [文件名]
      3. 情况二：已添加未提交。可通过git reset HEAD [文件名]方式从暂存区中退出内容。
      4. 然后恢复原始状态： git checkout - -[文件名]
5. 文件删除问题
   1. 如果有个文件只用于测试，之后就不再使用了，从git的角度来看删除也是一种修改。
   2. 从磁盘删除文件
   3. 当文件删除后查询状态，会显示文件已删除，文件从工作区删除了，同时也会提示文件被删除了
   4. 进行添加该删除提交（commit -a -m）
   5. 但是如果发现删除错了，可以进行文件的恢复，通过指针对象可以到删除提交前的状态（但是文件删除后只能通过版本控制进行恢复，所以删除文件应该慎重之至）