Object-Oriented Language and Theory

Nguyen Thi Thu Trang, trangntt@soict.hust.edu.vn

Hands-on lab guidelines

1. You have to follow the instructions in the hands-on lab and complete all exercises. If you can not finish them at class, please do it at home.
2. Git and Bitbucket tutorial:
   a. Git tutorial:
      i. Learn Git commands in https://git-scm.com/book/en/v2
      ii. https://git-scm.com/docs/gittutorial
      iii. https://www.atlassian.com/git/tutorials
      iv. https://www.youtube.com/watch?v=HVsySz-h9r4&list=PL-osiE80TeTuRUfjRe54Eea17-YfnOOAx
   b. Bitbucket tutorial:
      i. https://www.atlassian.com/git/tutorials/learn-git-with-bitbucket-cloud
      ii. https://bitbucket.org/tutorials/bucket-o-sand
3. Guidelines to git/bitbucket.org to submit your assignment/mini-project:
   a. Install Git (with optional tools: Source Tree, eGit for Eclipse…)
      i. https://www.youtube.com/watch?v=HVsySz-h9r4
   b. Create an account on https://bitbucket.org/
   c. Create a **private** repository naming "OOLT.ICT.20192.StudentID.StudentName" (ICT) or "OOLT.VN.20192.StudentID.StudentName" (VN). If you don't follow this naming convention, your repository will be ignored.
   d. Add "trangntt-for-student" (trangntt.for.student@gmail.com) as a member of your repository
   e. Clone a repository, create a branch (the default and main one is master):

```
cd <local_working_folder>
git clone <repository_url>
cd <repository_name> /* e.g. OOLT.ICT.20192.StudentID.StudentName */
git branch /* let you know which branch you're working on */
git checkout -b <your_branch> /* create and switch to your new branch */
```

Then you can make any changes such as create a new file, modify an existing file, delete a file…

   f. Commit and push after changing some resources:

```
cd <local_working_folder/repository_name>
git add -A /* -A: for all operations or . for without deletions in the current folder */
git status /* check status of the stage */
git commit -m "Your comment for the update" /* commit with comment*/
git push origin <your_branch> /* push from local repo to remote repo */
```

After checking carefully the new branch code, the leader can go to the Bitbucket to *Create merge request*, then *Approve merge request* to merge the code to the master branch.

   g. Pull without conflicts

```
git stash /* run this command if you want to ignore your current change */
git pull origin <your_branch>
```

   h. Pull and resolve conflicts

After you commit and push to the remote repository, if there is any conflict when creating merge request, you need to resolve conflicts.

```
git checkout master
git pull origin master
git checkout <your_branch>
git rebase master
git add -A
git status
git rebase –continue
```

You need to resolve conflicts by checking the resources that have conflicts. There are both versions in the resources so that you can observe and make decisions, e.g.

Having conflig:

```
<<<<<<< HEAD /* master branch in the remote repository */
      suggestor.setClientId(clientID);
      suggestion = suggestor.translate(input).replaceAll("-", " ");
======= /* your branch in the local repository */
      suggestion = suggestor.translate(input);
>>>>>>> Your comment for the update
```

Merging (merge your local repository version with the replaceAll() method, remove the first statement of the remote repository):

```
      suggestion = suggestor.translate(input).replaceAll("-", " ");
```

Then continue, force to push the merged version to the remote repository

```
git push origin <your_branch> -f
```