**Hướng dẫn làm câu git THCM**

Tạo folder tên là **Cau3** bằng cơm hoặc dùng git bash:

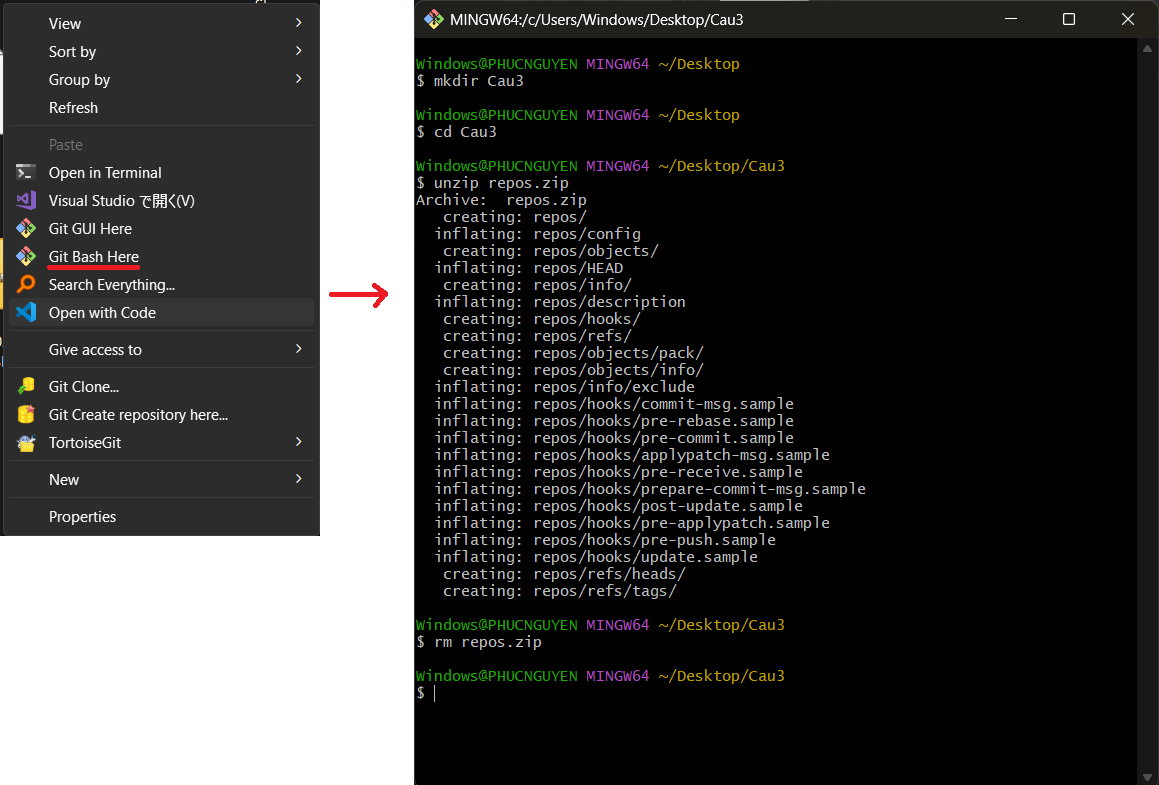
click chuột phải chọn **Git Bash Here** rồi gõ **mkdir Cau3**

**cd Cau3** ---> di chuyển vào folder **Cau3**

Coppy file **repos.zip** của đề cho vào folder **Cau3**

**unzip repos.zip** ---> giải nén file repos.zip

**rm repos.zip** ---> xoá file zip

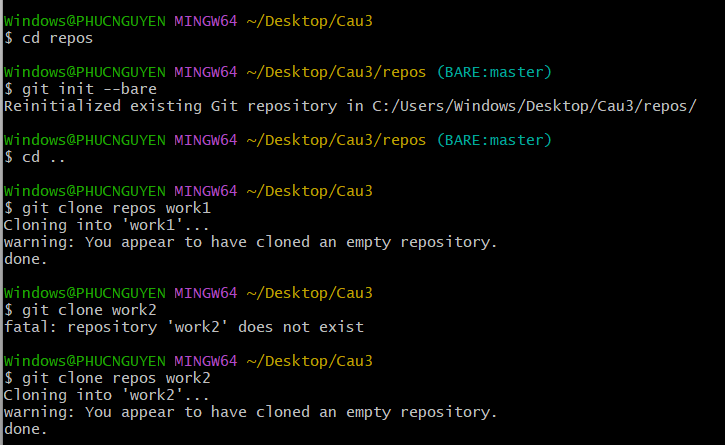
****

**cd repos** ---> di chuyển vào folder **repos**

**git init --bare** ---> khởi tạo repo

**cd ..** ---> back về folder cha (**Cau3**)

**git clone repos work1**

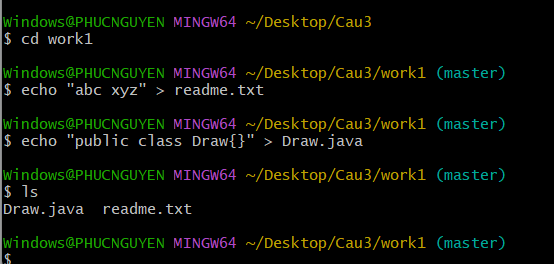
**git clone repos work2**

**cd work1** ---> di chuyển tới folder **work1**

**echo “abc xyz” > readme.txt** ---> tạo file **readme.txt** với nội dung “abc xyz”

**echo “public class Draw{}” > Draw.java** ---> tạo file **Draw.java**

lệnh **ls** để show danh sách file trong folder đang trỏ tới

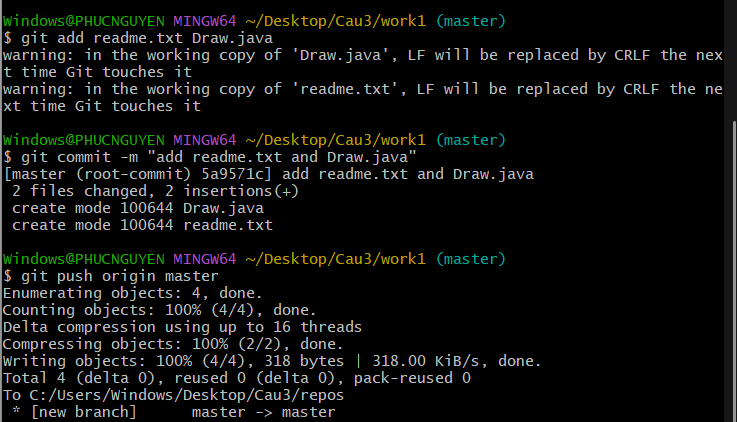
(check xem tạo file thành công chưa, ko check cũng được)

**git add readme.txt Draw.java** ---> add 2 file vào vùng chờ

Lưu ý: trước khi **git commit** phải **git add tên\_file**

hoặc **git add .** (add tất cả file vào vùng chờ)

**git commit -m “noi dung commit”**

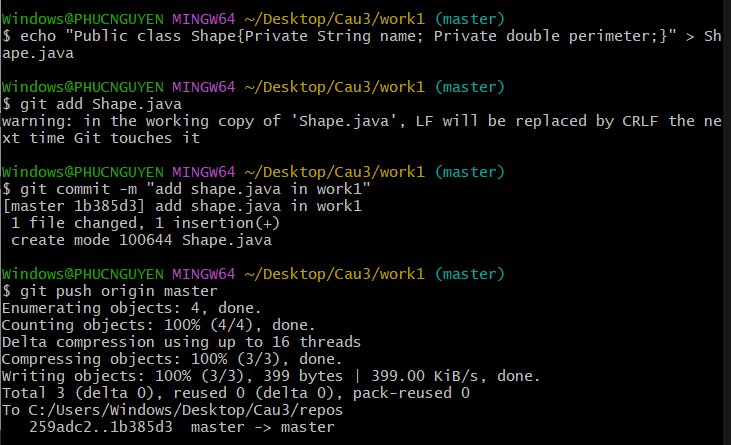
**git push origin master**

Tạo file Shape.java trong **work1** và push lên:

**echo “Public class Shape { Private String name; Private double perimeter;}” > Shape.java**

**git add Shape.java**

**git commit -m “add shape.java in work1”**

**git push origin master**

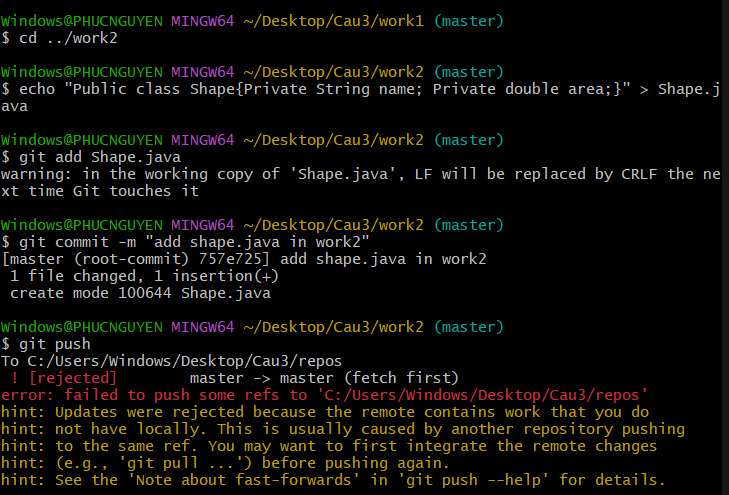
back về **work2** tạo file Shape.java và push lên:

**cd ../work2**

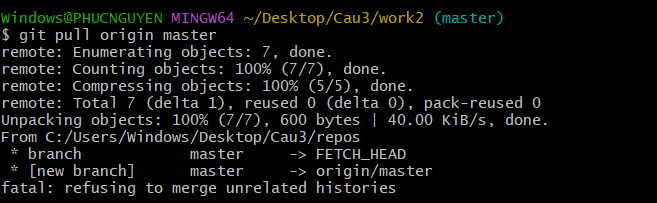
**echo “Public class Shape {Private String name; Private double area;}” > Shape.java**

**git add Shape.java**

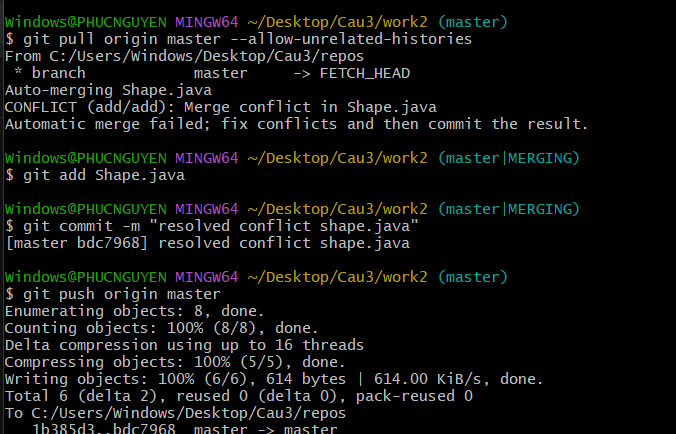
**git commit -m “add shape.java in work2”**

**git push origin master**

lúc này khi push lên sẽ có conflit ---> xử lí conflict

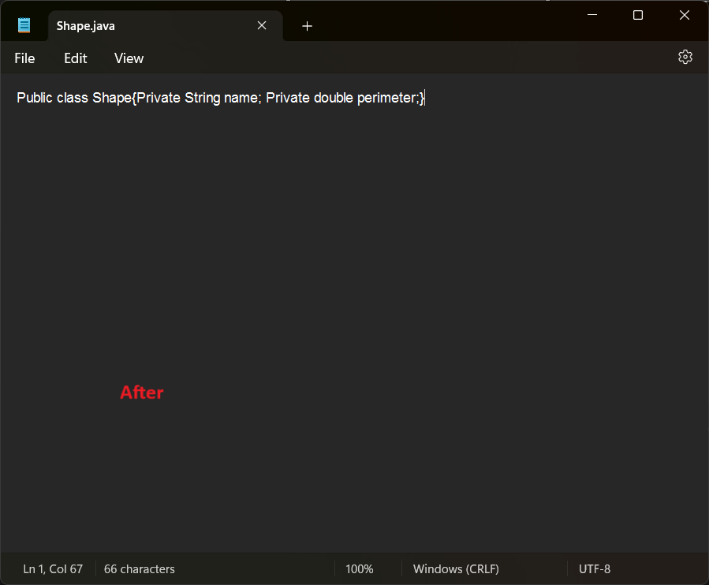
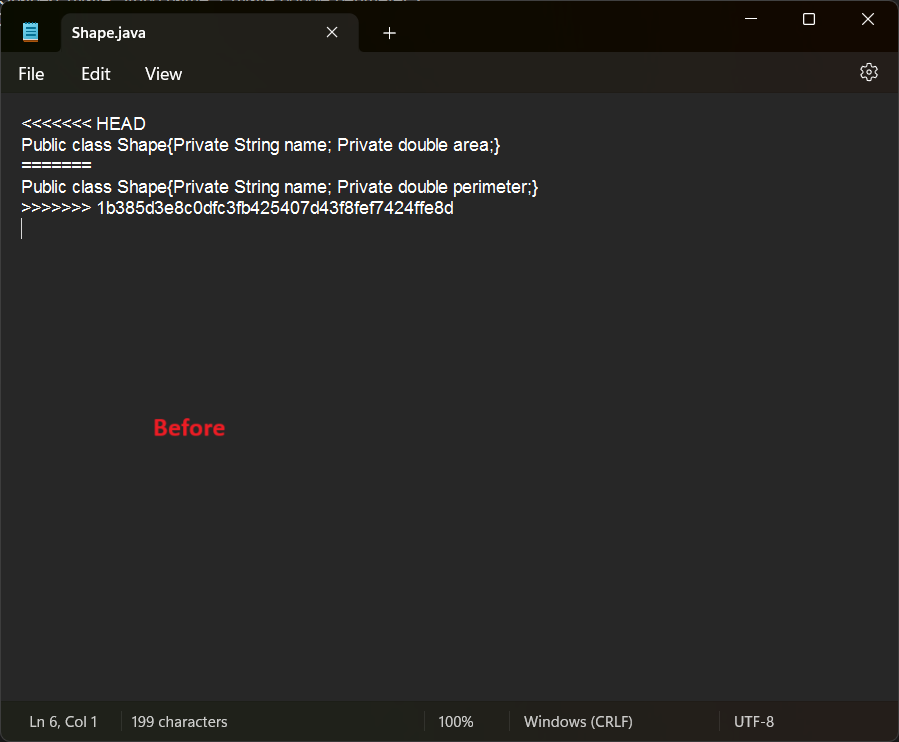
**git pull origin master**

nếu báo lỗi refusing to merge unrelated histories

---> **git pull origin master --allow-unrelated-histories**

Mở file Shape.java trong work2 (Open with > Notepad fix bằng cơm)

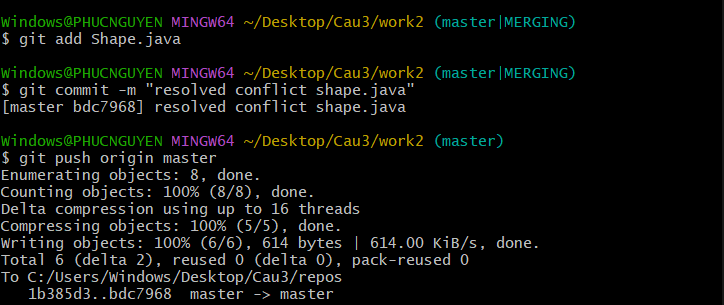
Chưa biết conflict là gì thì đọc thêm link này

<https://topdev.vn/blog/cach-xu-ly-conflict-khi-git-merge/>

Sau khi fix bằng cơm xong thì add rồi commit, push lại:

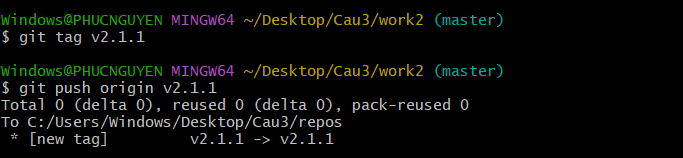
**git add Shape.java**

**git commit -m “resolved conflict shape.java”**

**git push origin master**

tạo snapshot v2.1.1 và push lên:

**git tag v2.1.1**

**git push origin v2.1.1**

đổi tên file Draw.java -> DrawApp.java và push

**cd ../work1**

**mv Draw.java DrawApp.java**

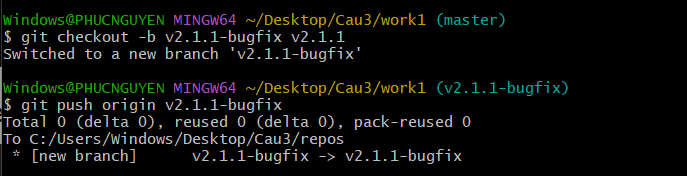
**git add DrawApp.java**

**git commit -m “rename Draw.java to DrawApp.java”**

**git push**

tạo branch v2.1.1-bugfix và push

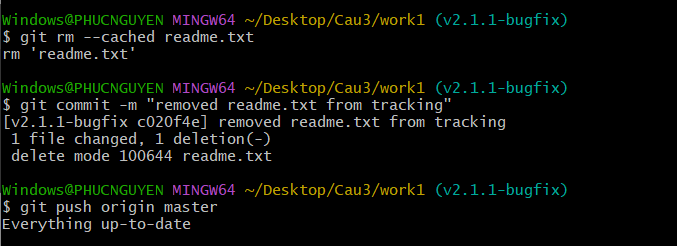
**git checkout -b v2.1.1-bugfix v2.1.1**

**git push origin v2.1.1-bugfix**

loại bỏ readme.txt khỏi danh sách theo dõi của work1

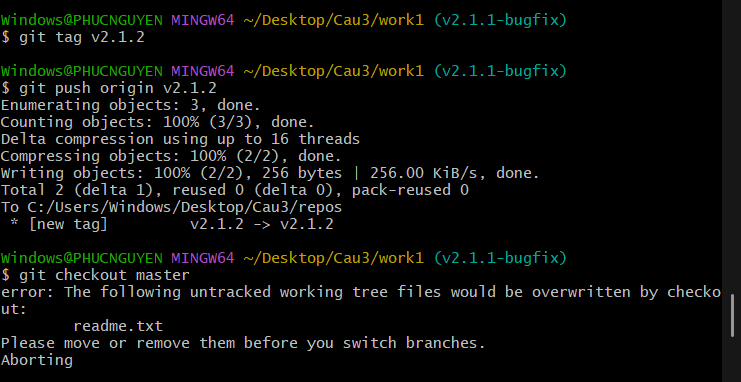
**git rm --cached readme.txt**

**git commit -m “removed tracking readme.txt”**

**git push origin master**

tạo snapshot v2.1.2

**git tag v2.1.2**

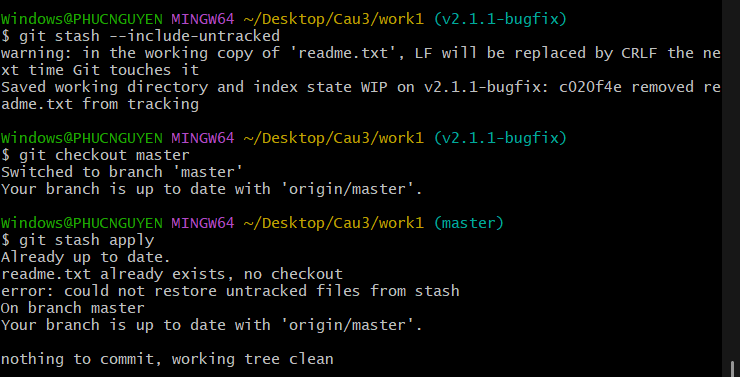
**git push origin v2.1.2**

checkout qua branch master lúc này sẽ bị lỗi

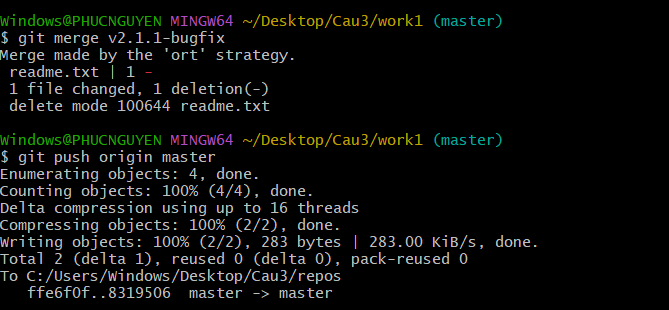
lưu file readme.txt vào stash (bộ nhớ đệm rồi mới chuyển qua branch master được)

**git stash --include-untracked**

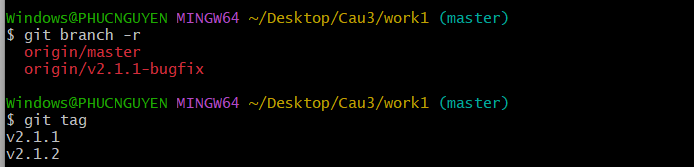
**git checkout master**

**git stash apply** ---> switch qua branch master thành công thì stash apply lại

git merge v2.1.1-bugfix

git push origin master

**git branch -r** ---> danh sách branch

**git tag** ---> danh sách snapshot tạo nảy giờ

Done, zip cục **Cau3** lại