

Zadania 1-21 ze strony <https://gitexercises.fracz.com/>

1.Master

git verify

po prostu pushujemy

2.Commit-one-file

git add A.txt

git commit -m "Commit A "

wybieramy który plik chcemy pushowac, dodajemy go do staging area i potem pushujemy

3.commit-one-file-staged

git reset A.txt # cofamy dodanie A

git commit -m "Commit B" #commitujemy B

4.ignore-them

#Towrzymy plik .gitignore za pomocą „echo” lub ręcznie

#Wpisujemy *.o *.exe *.jar libraries/ do pliku .gitignore ręcznie lub za pomocą „echo”

#Dodajemy plik

git add .gitignore

#commitujemy

git commit -m „gitignore”

5.chase-branch

git merge escaped

#mergujemy branch escpaed do chase-branch

6.merge-conflict

naprawiamy merge-conflict ręcznie

git merge another-piece-of-work

#wpisujemy 2+3=5 do pliku equation.txt i wyrzucamy resztę

git add equation.txt

git commit -m “merge-conflict-solved”

7.save-your-work

#Używamy git stash używanego do zmian roboczych bez dodawania ich do historii commitów

git stash

usuwamy linijkę z bugiem

git commit -am „fix1” commitujemy zmianę

git stash pop # ściągamy ze stasha

do pliku bug.txt wpisujemy „Finally, fishied it!”

git commit -am „Finish my work” # commitujemy

8. change-branch-history

Używamy git rebase/ przenoszącego historię commitów

git rebase hot-bugfix #przenosimy commity na aktualnego brancha

9. remove-ignored

git rm igonred.txt # usuwanie pliku z repo i stagea

git commit -am “Deleting” # commitujemy

10.case-sensitivie-filename

git mv File.txt file.txt # git mv zmienia/przenosi nazwy plików w repozytorium

git commit -am “Lowercase f in filename” # commitujemy

11.fix-typo

poprawiamy literówkę w pliku ręcznie

git commit -a --amend # git amend edytuje ostatniego commita podpiętego do heada

zmieniamy literów przy commicie

12.forge-date

#używamy opcji --no-edit ktore nie edytuje wiadomości commita

git commit --amend --no-edit --date="1987-08-03"

13. fix-old-typo

```
git rebase -i HEAD^^
```

```
# zmieniamy opcje z pick->edit dla literówki
```

```
#otwieramy plik z literówką i ją zmieniamy
```

```
git add file.txt
```

```
git rebase --continue # zmieniamy literówkę
```

```
# otwieramy plik file.txt i wyrzucamy wszystko oprócz dwóch przed ostatnich linii
```

```
git add file.txt
```

```
git rebase --continue # nic nie zmieniamy
```

14. commit-lost

```
git reflog # przywracamy poprzedni stan
```

```
git reset --hard HEAD@{1} # przywracamy stageing do momentu ostatniego commita
```

15.split-commits

```
git reset HEAD^ # resetujemy poprzedniego commita
```

```
git add first.txt # dodajemy pierwszy
```

```
git commit -m "First"
```

```
git add second.txt # dodajemy drugi
```

```
git commit -m "Second"
```

16.too-many-commits

```
git rebase -i HEAD^^ # bierzemy dwa ostatnie commity
```

```
# w wyświetlonym pliku zmieniamy opcje edit -> squash która pozwala nam złączyć dwa sąsiednie commity
```

17.executable

```
# dodajemy mozliwosc wykonywania pliku
```

```
git update-index --chmod=+x script.sh # modyfikujemy index(obszar stageing) naszego commita
```

18. commit-parts

git add -p # dodajemy patch, czyli huki (pojedyncze fragmentu kodu)

gdy otworzy nam się opcja edycji wybieramy s – split, wybieram dwa fragmenty do stagea (y) a dwa pomijamy(n) (ten z usuniętą częścią(czerwoną) i ostatni),

git commit -m „cz1”

git add -p

splitujemy, mamy podział na dwa commity wiec dajemy dwa [y]

git commit -m „cz2”

19.pick-your-features

uzywamy cherry-pick ktory przenosi commity z jednej galezi do drugiej bez laczzenia branchy

git cherry-pick feature-a

git cherry-pick feature-b

git cherry-pick feature-c

wchodzimy do pliku program.txt i wyrzucamy wszystko co nie jest tekstem(rozwiązujemy konflikt)

git add program.txt

git cherry-pick –continue # sprawdzamy czy wszystko jest ok

20. rebase-complex

git rebase --onto your-master issue-555 rebase-complex

opcja –onto przenosi commita na innego brancha

wybieram commity od issue-555 do rebase-complex i przenosimy na mastera

21.invalid-order

git log -2 # sprawdzamy które commity powinniśmy zmienić

git rebase -i HEAD~2 # uzywamy interaktywnego rebasa, ~2 oznaczan dwa ostantie commity

wymieniamy recznie dwie pierwsze linijki