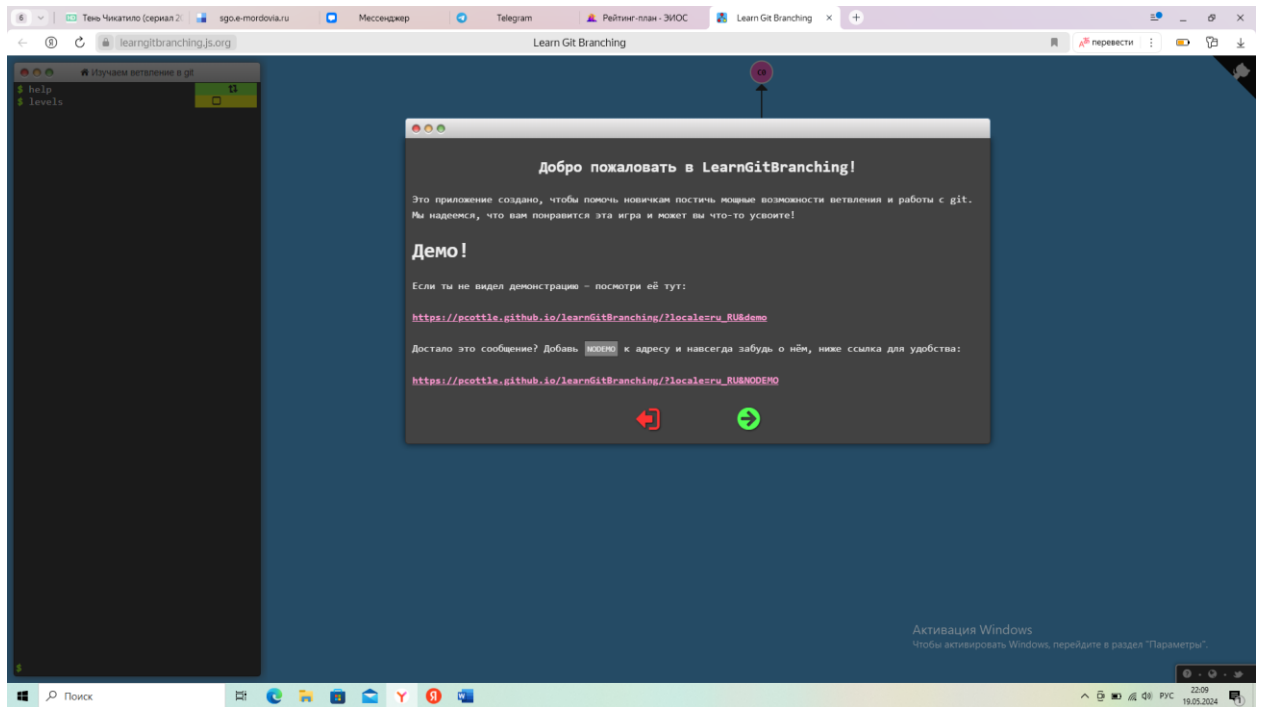
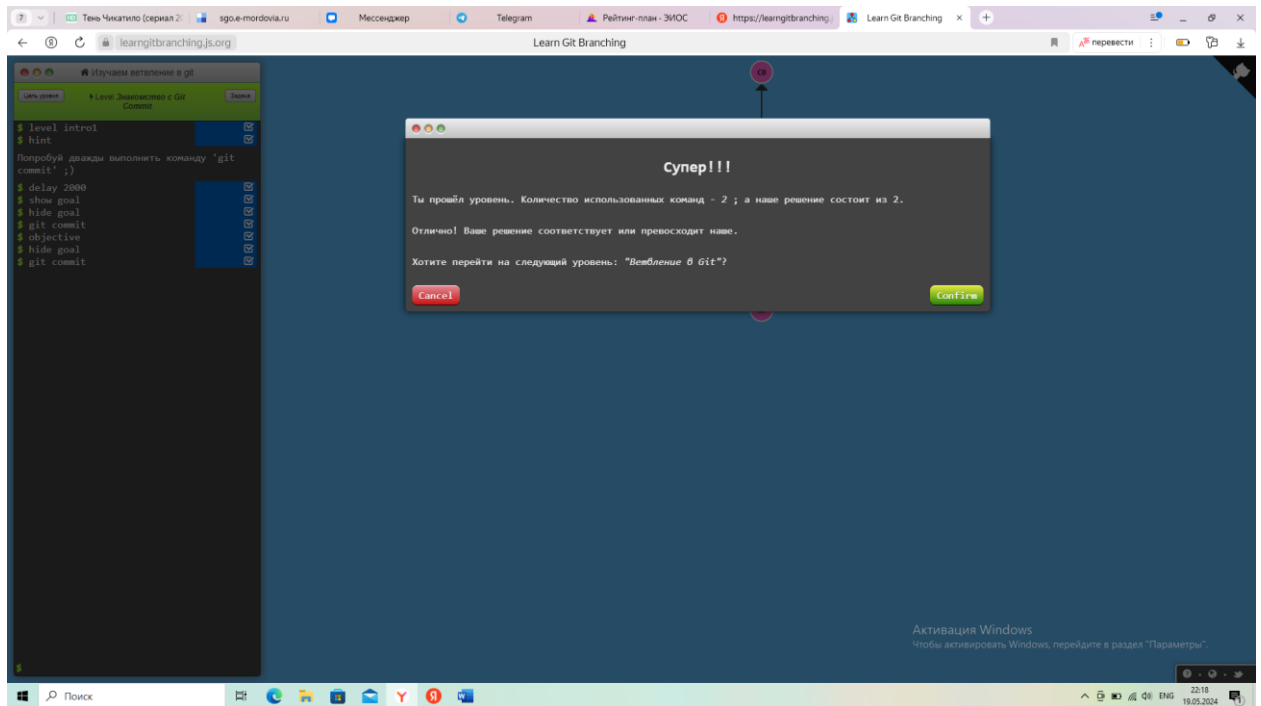


Вдовин Евгений Васильевич 214В



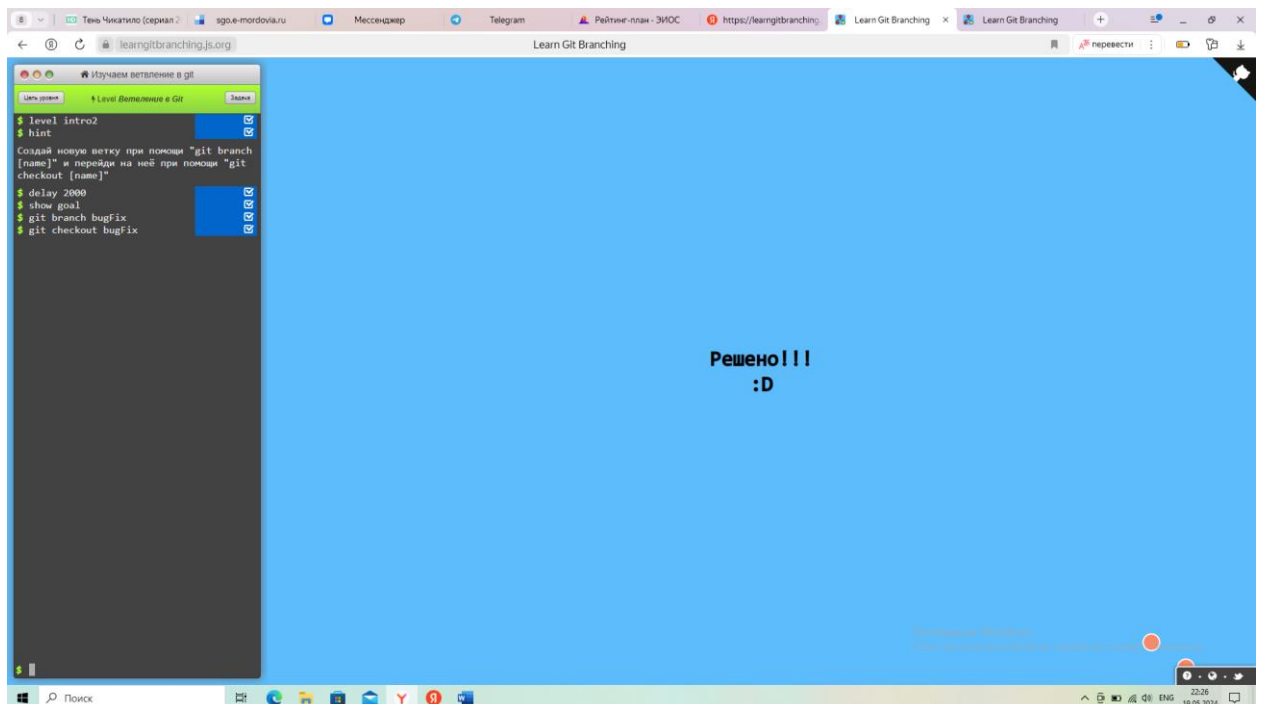
1.1

```
$ level intro1
$ hint
Попробуй дважды выполнить команду 'git commit' ;)
$ delay 2000
$ show goal
$ hide goal
$ git commit
$ objective
$ hide goal
$ git commit
```



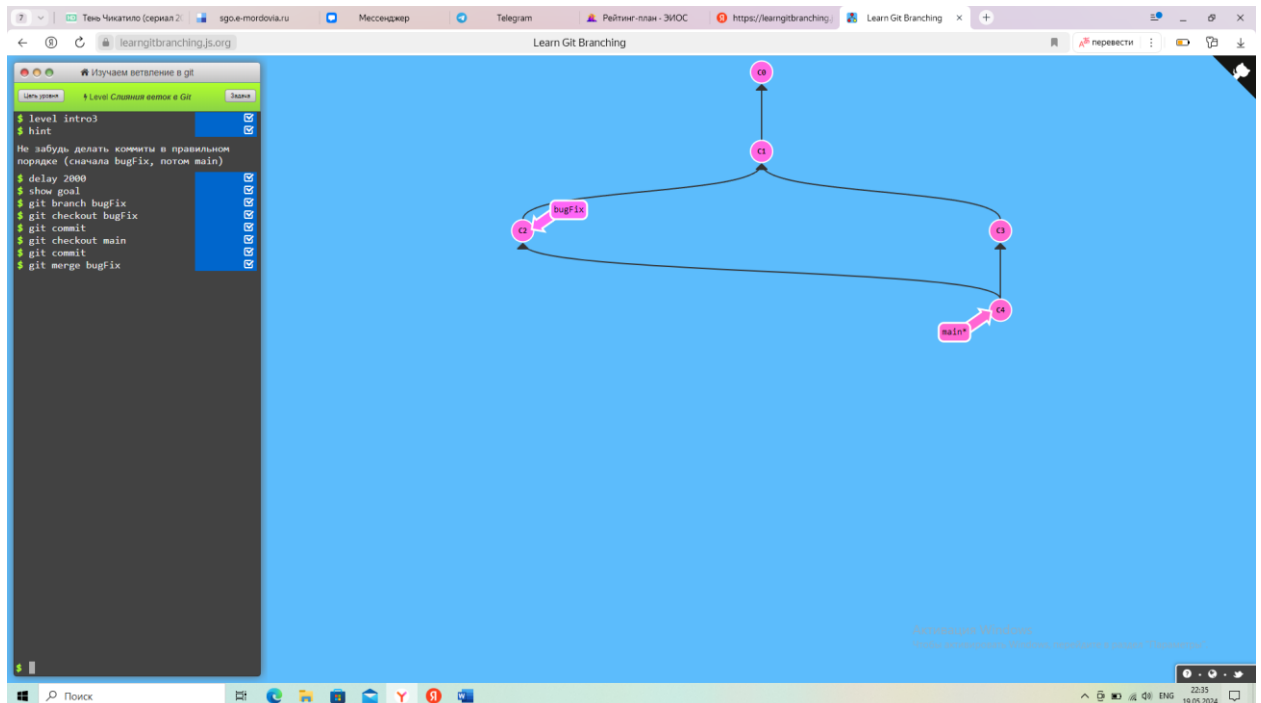
1.2

```
$ level intro2
$ hint
Создай новую ветку при помощи "git branch [name]" и перейди на неё при помощи "git checkout [name]"
$ delay 2000
$ show goal
$ git branch bugFix
$ git checkout bugFix
```



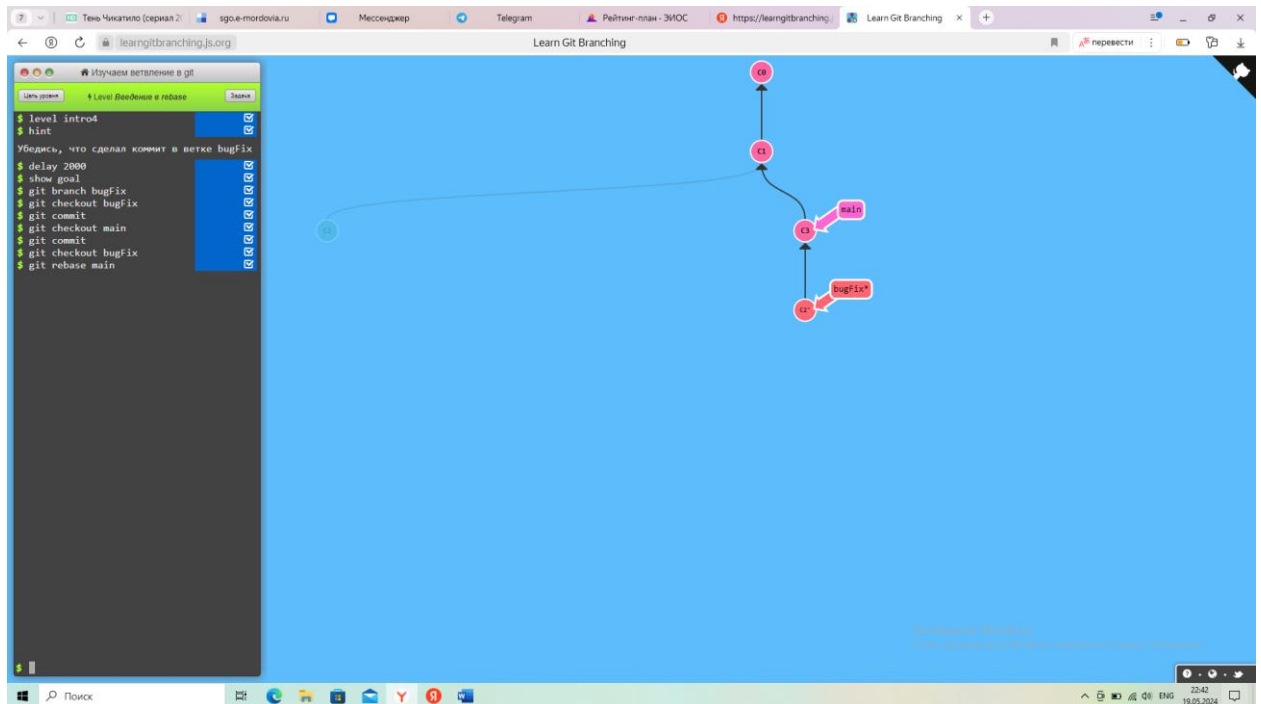
1.3

```
$ level intro3
$ hint
Не забудь делать коммиты в правильном порядке (сначала bugFix,
потом main)
$ delay 2000
$ show goal
$ git branch bugFix
$ git checkout bugFix
$ git commit
$ git checkout main
$ git commit
$ git merge bugFix
```



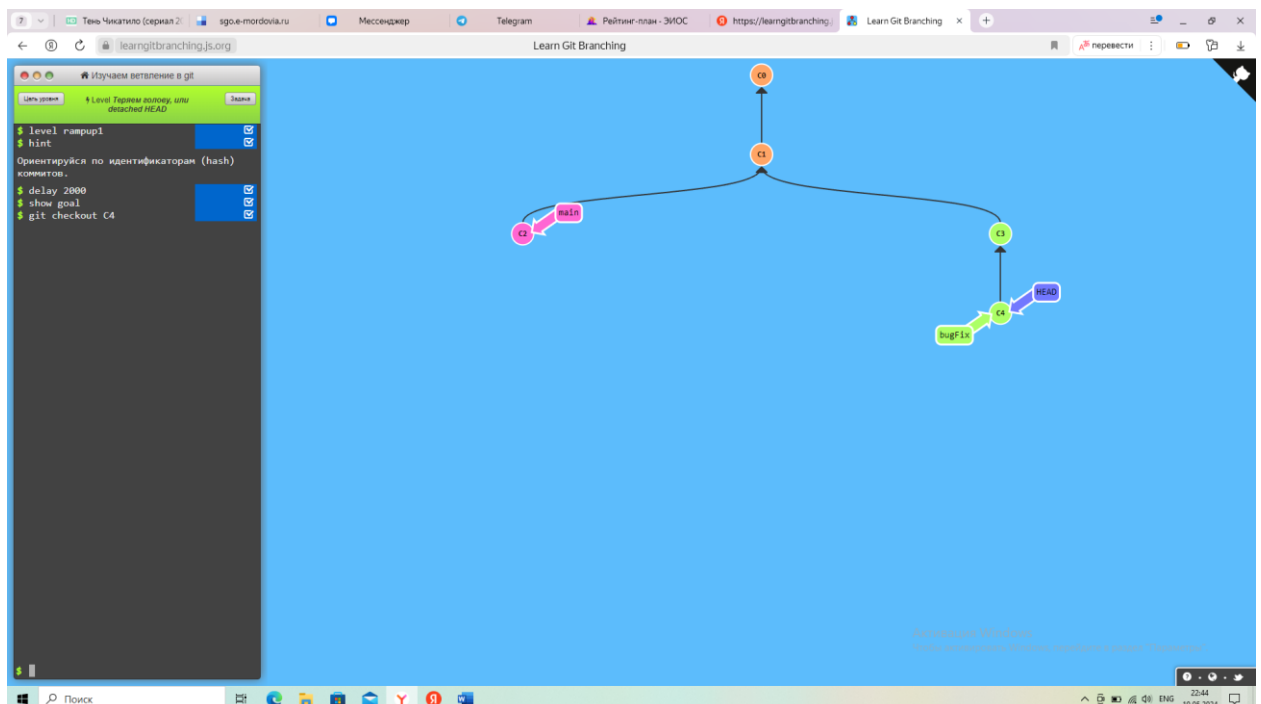
1.4

```
$ level intro4
$ hint
Убедись, что сделал коммит в ветке bugFix
$ delay 2000
$ show goal
$ git branch bugFix
$ git checkout bugFix
$ git commit
$ git checkout main
$ git commit
$ git checkout bugFix
$ git rebase main
```



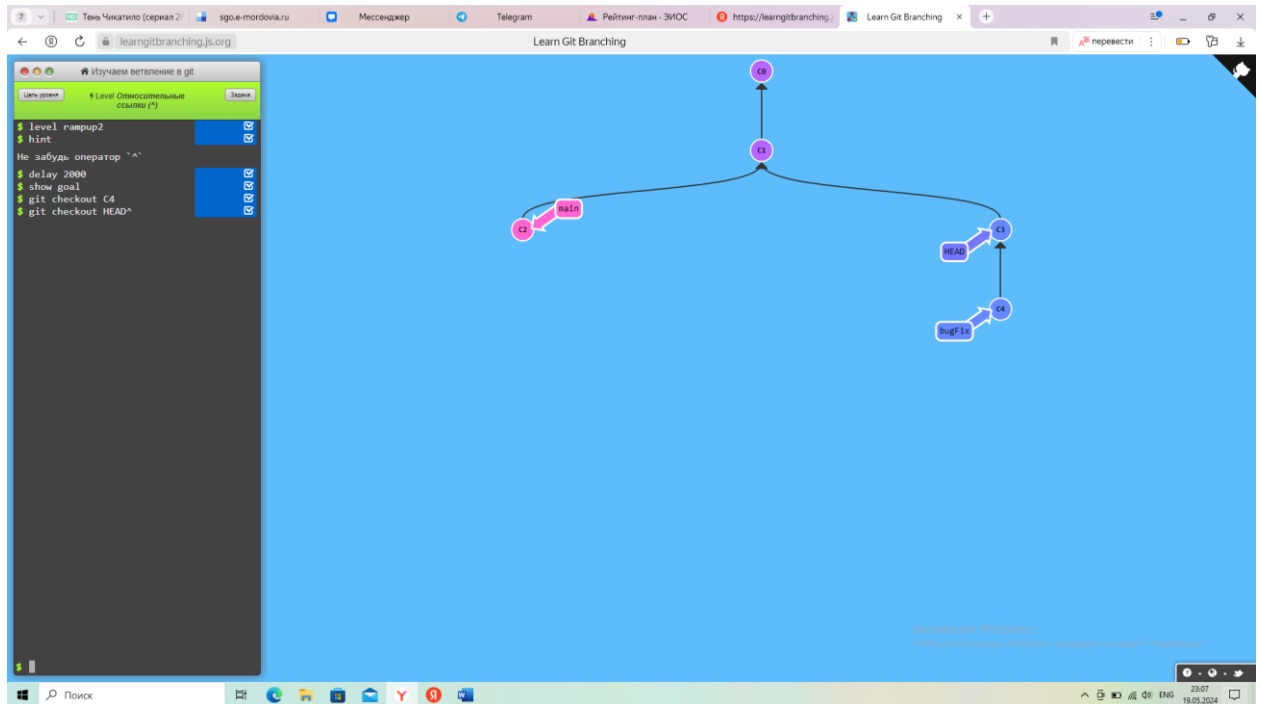
2.1

```
$ level rampup1
$ hint
Ориентируйся по идентификаторам (hash) коммитов.
$ delay 2000
$ show goal
$ git checkout C4
```



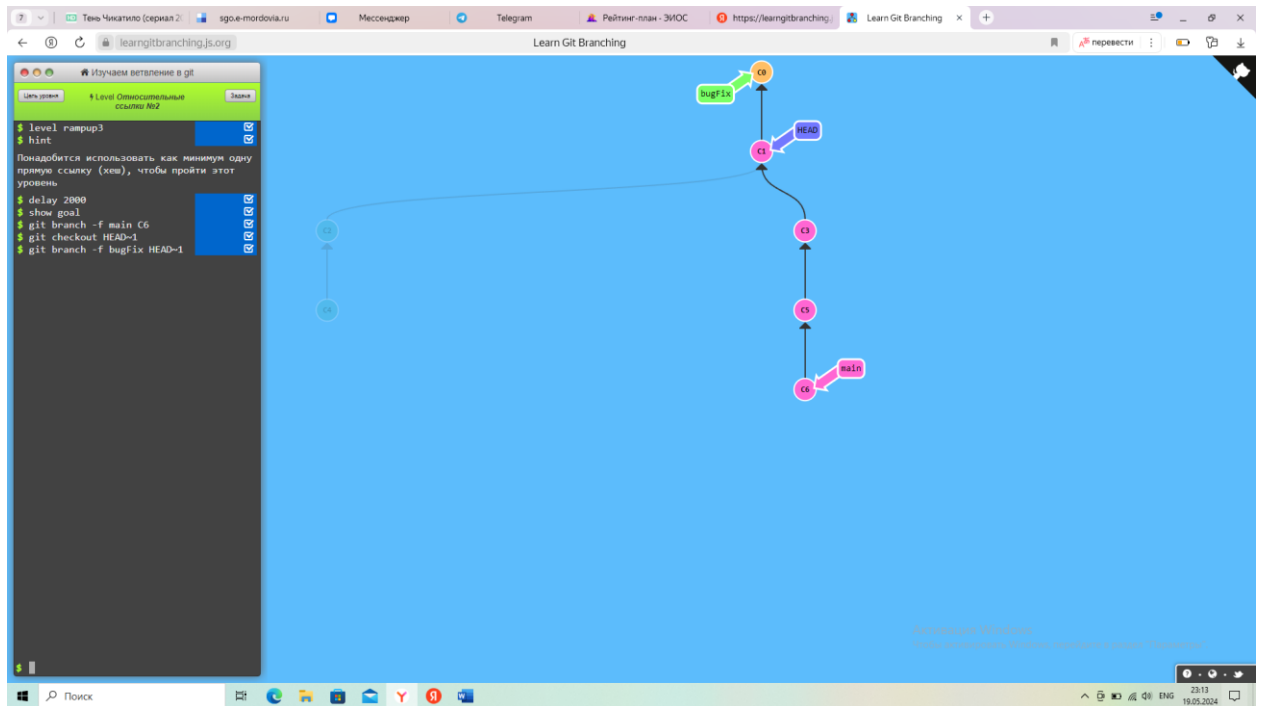
2.2

```
$ level rampup2
$ hint
Не забудь оператор `^`
$ delay 2000
$ show goal
$ git checkout C4
$ git checkout HEAD^
```

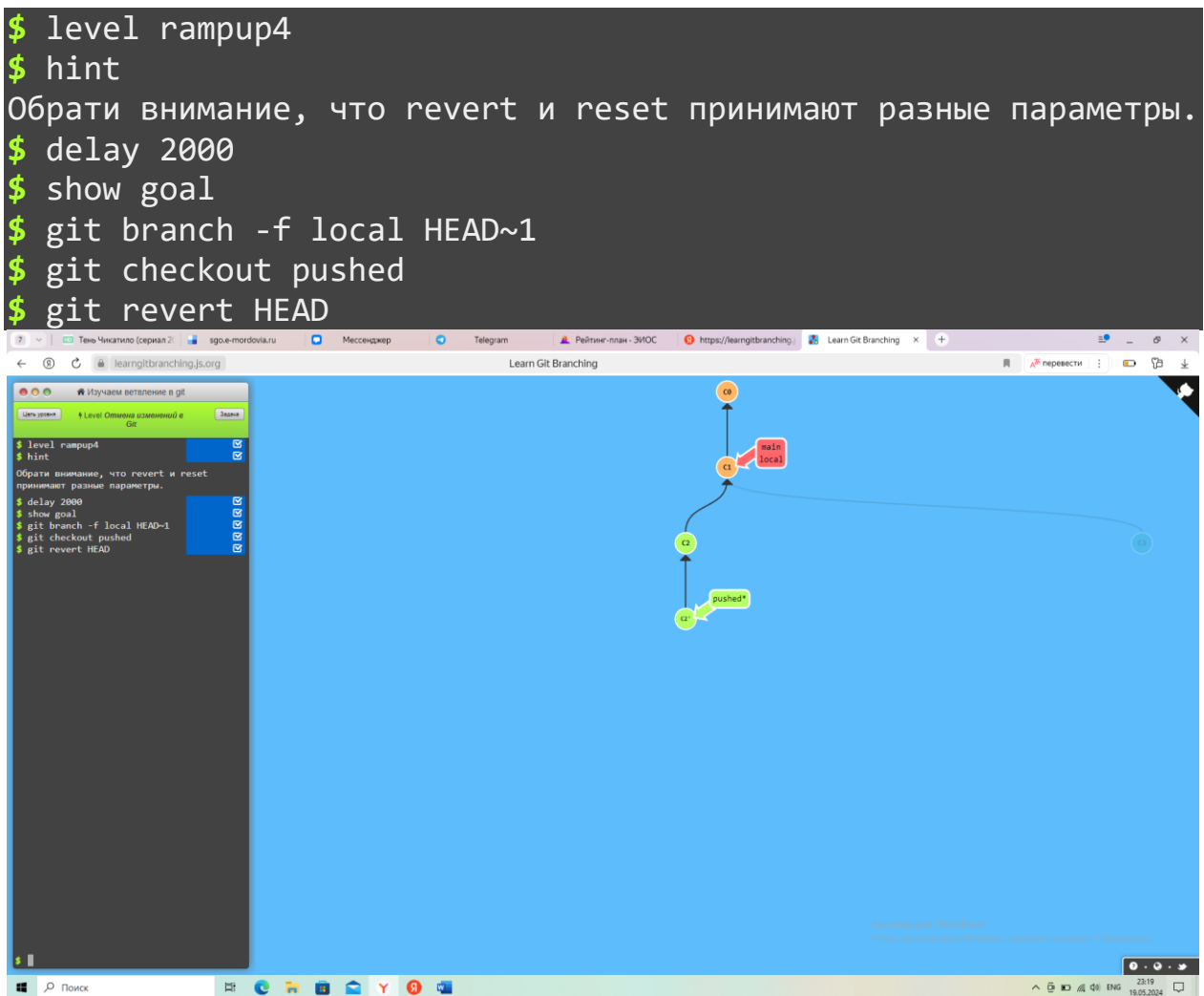


2.3

```
$ level rampup3
$ hint
Понадобится использовать как минимум одну прямую ссылку (хеш),
чтобы пройти этот уровень
$ delay 2000
$ show goal
$ git branch -f main C6
$ git checkout HEAD~1
$ git branch -f bugFix HEAD~1
```



2.4

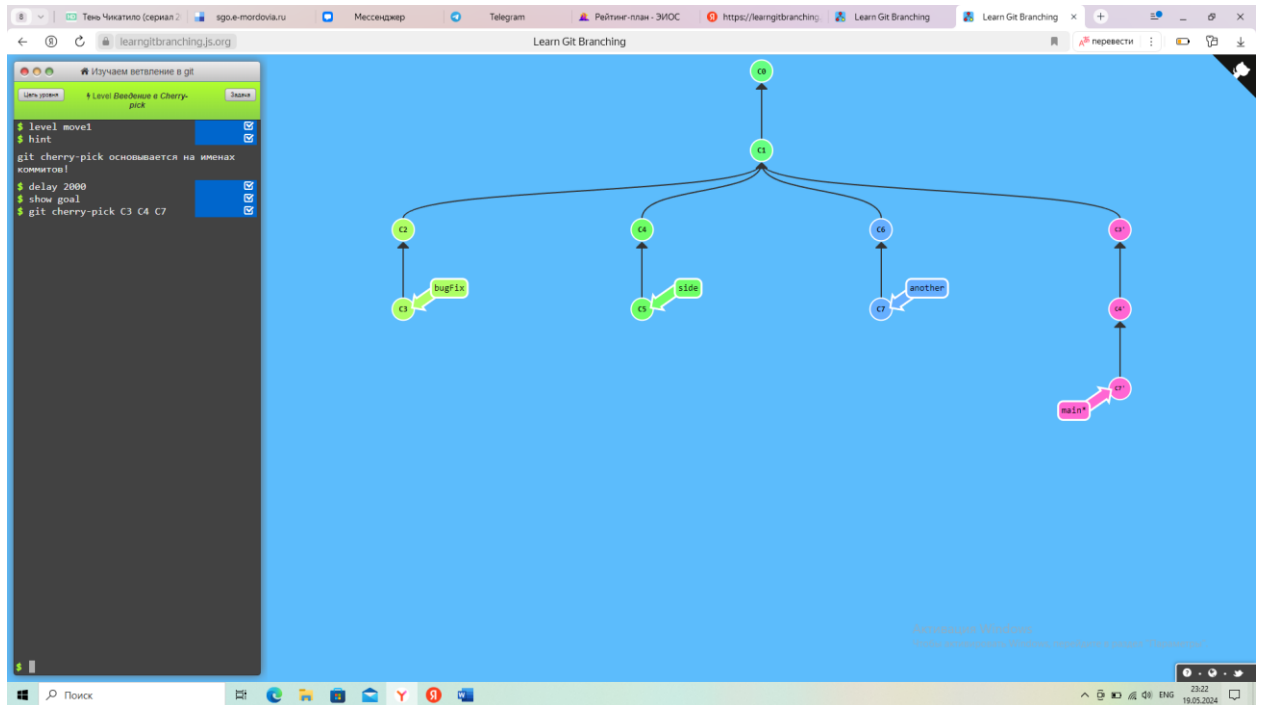


3.1

```

$ level move1
$ hint
git cherry-pick основывается на именах коммитов!
$ delay 2000
$ show goal
$ git cherry-pick C3 C4 C7

```

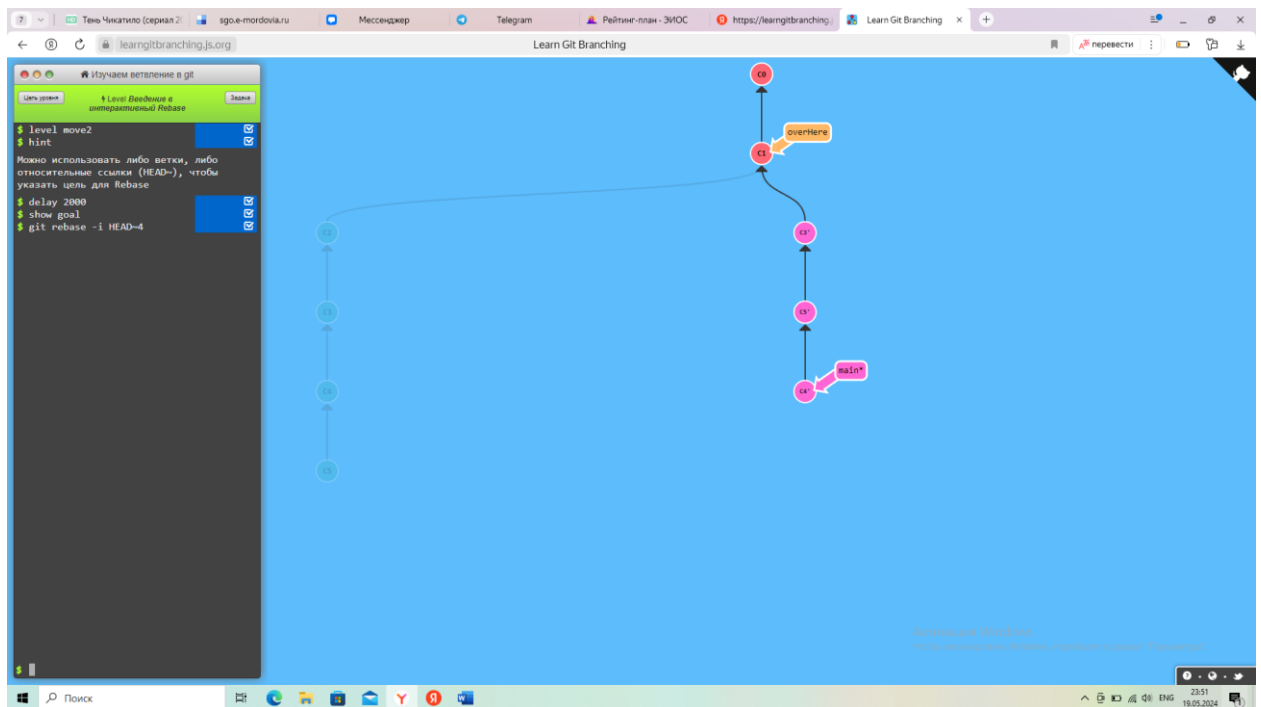
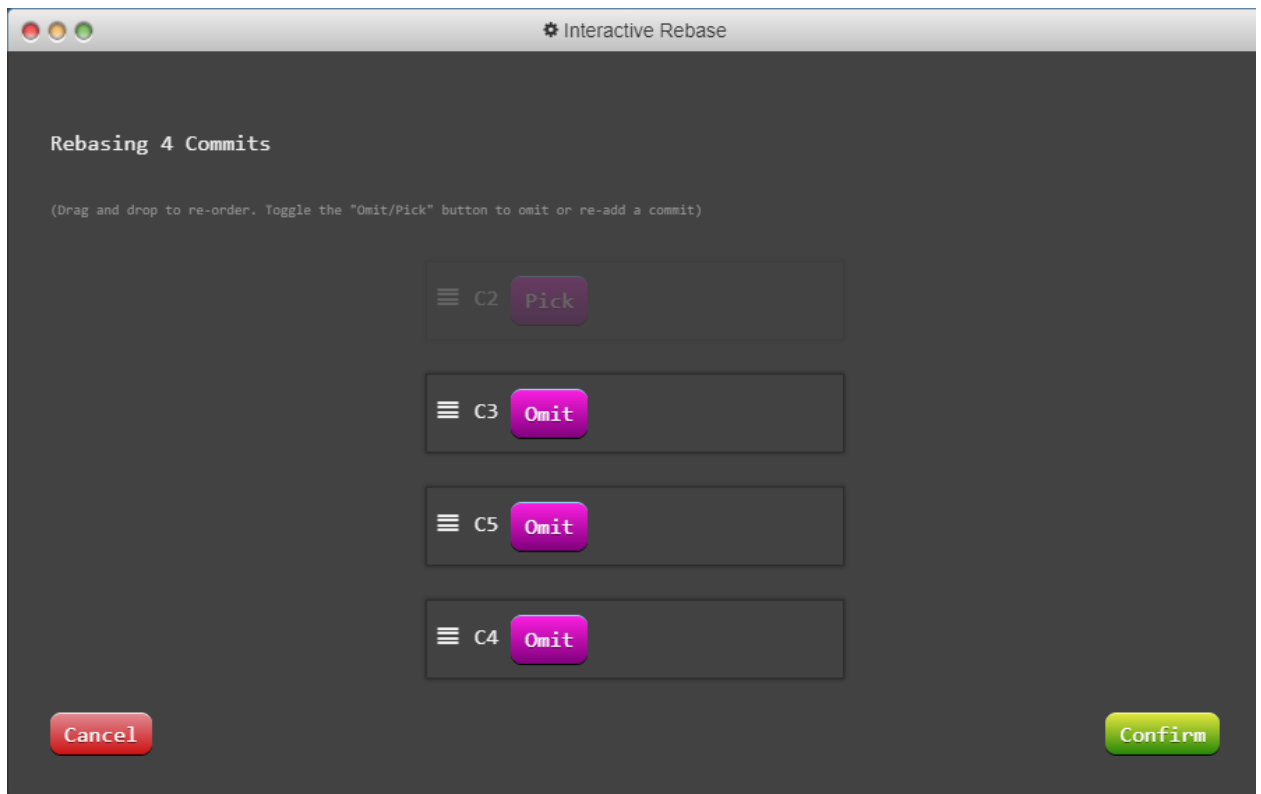


3.2

```

$ level move2
$ hint
Можно использовать либо ветки, либо относительные ссылки
(HEAD~), чтобы указать цель для Rebase
$ delay 2000
$ show goal
$ git rebase -i HEAD~4

```

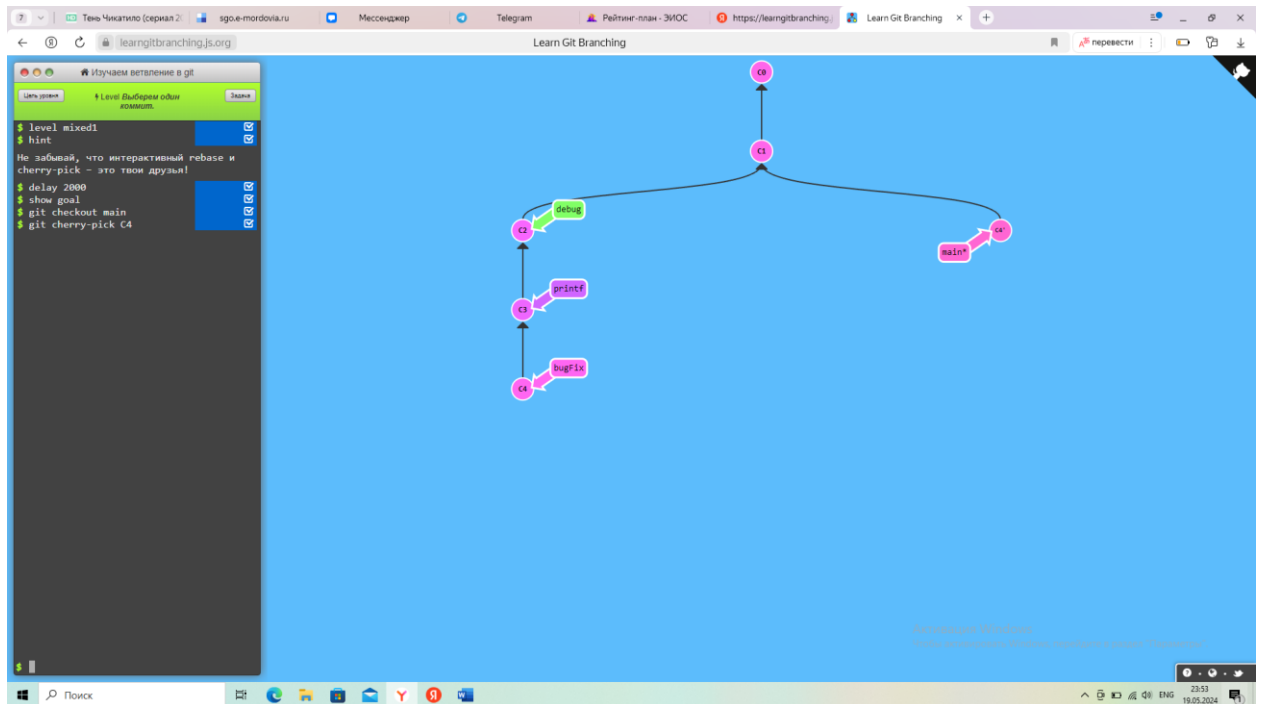


4.1

```
$ level mixed1
$ hint
Не забывай, что интерактивный rebase и cherry-pick – это твои друзья!
$ delay 2000
$ show goal
$ git checkout main
```

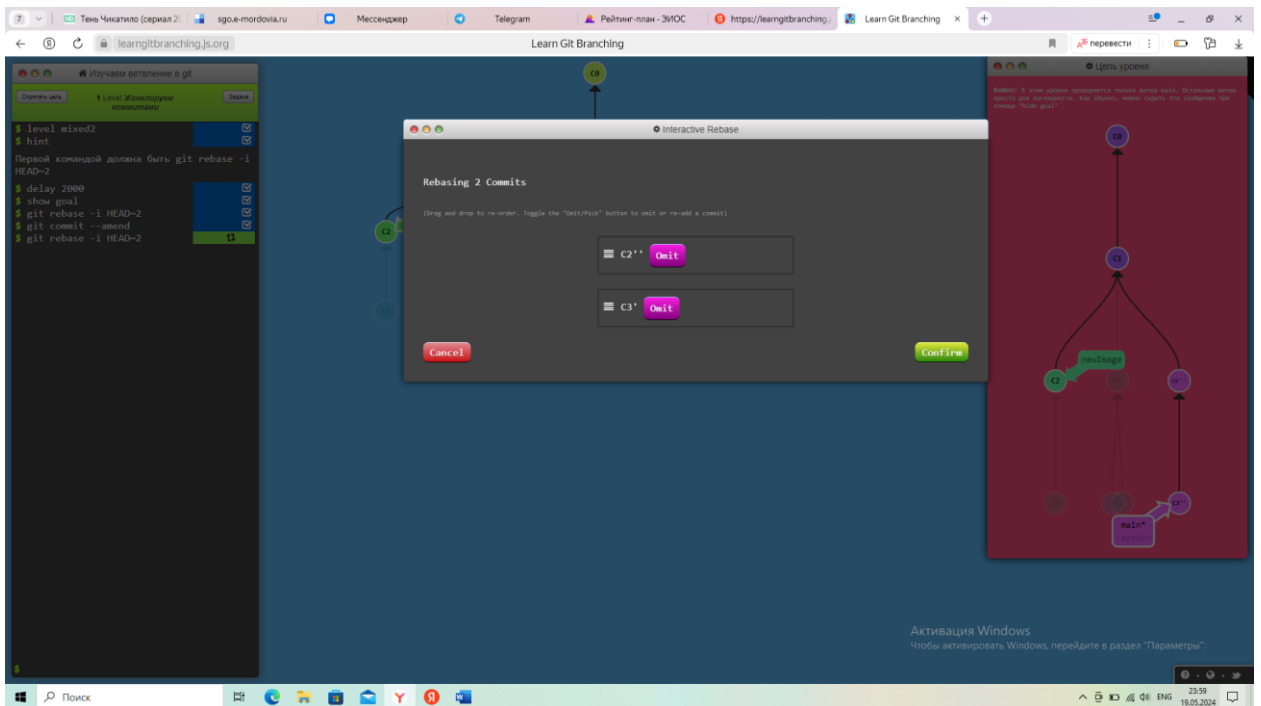
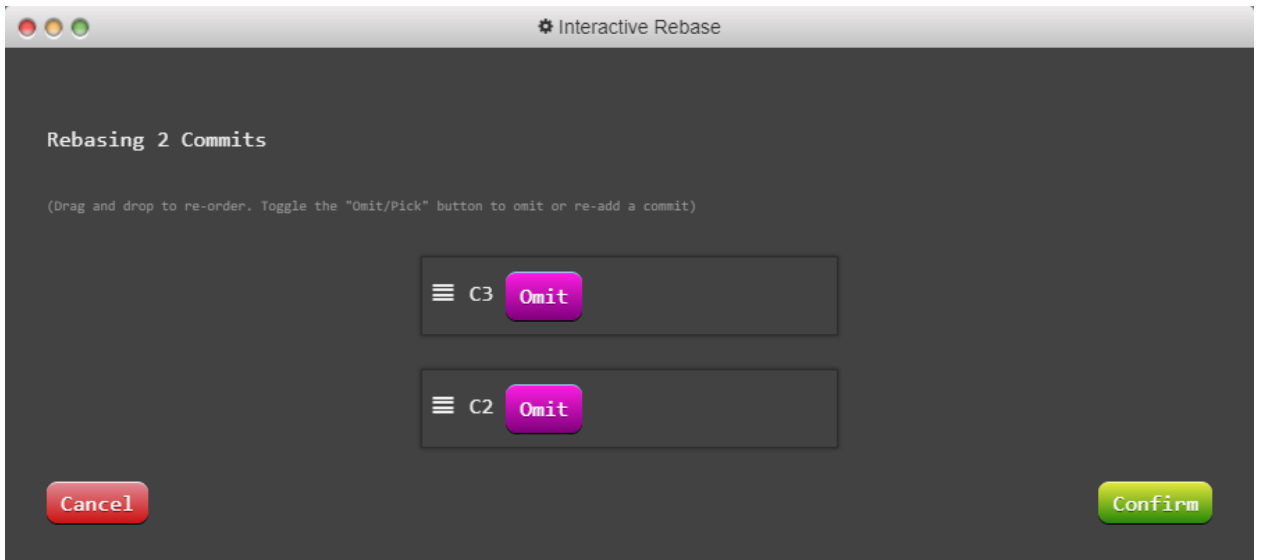


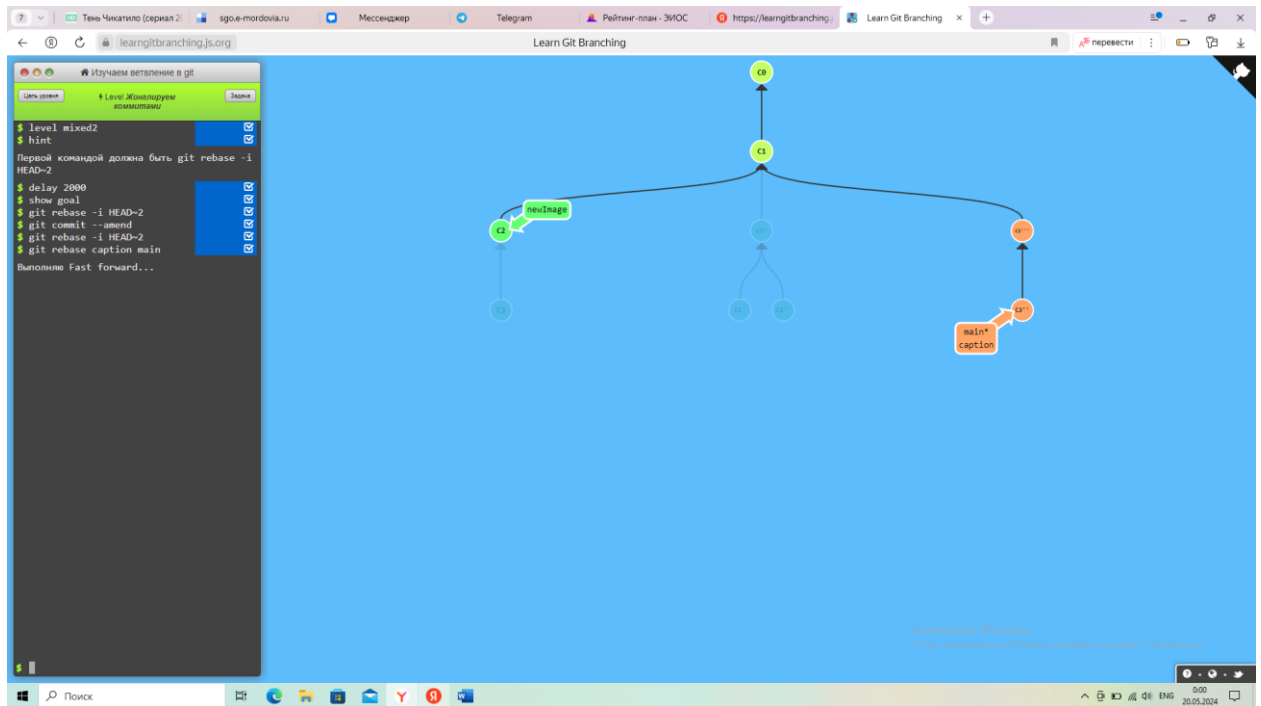
```
$ git cherry-pick C4
```



4.2

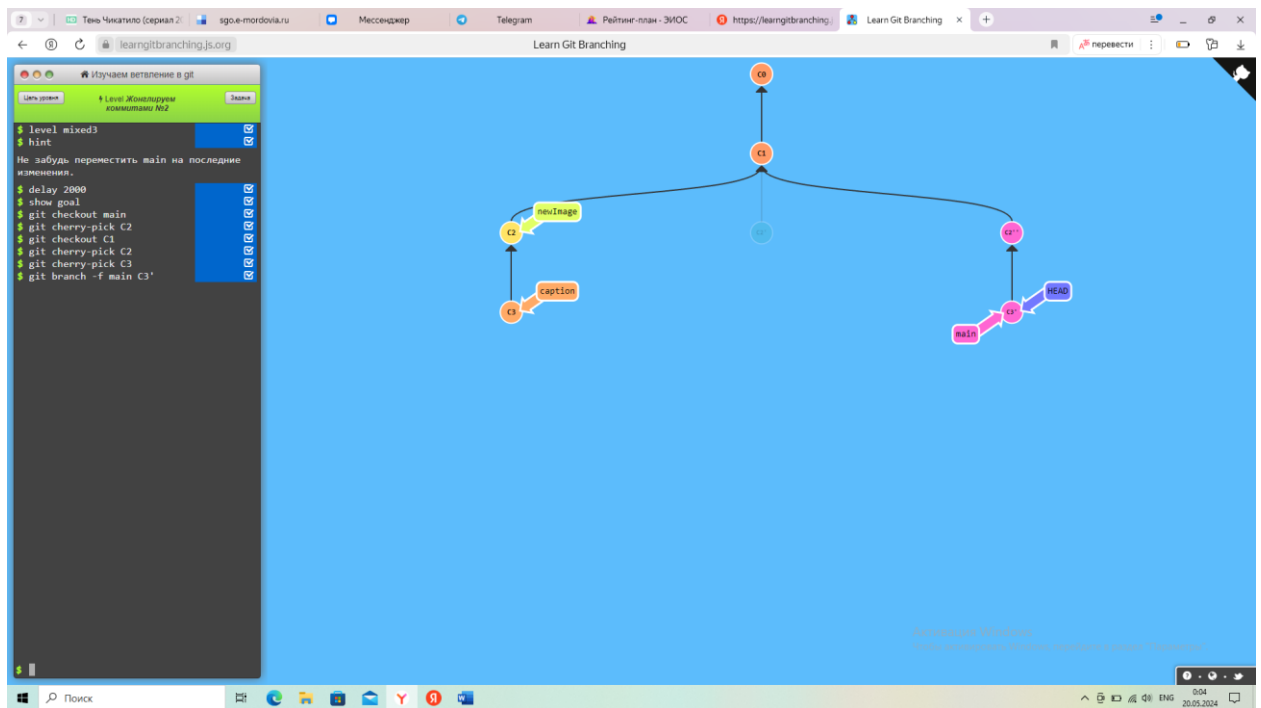
```
$ level mixed2
$ hint
Первой командой должна быть git rebase -i HEAD~2
$ delay 2000
$ show goal
$ git rebase -i HEAD~2
$ git commit --amend
$ git rebase -i HEAD~2
$ git rebase caption main
Выполняю Fast forward...
```





4.3

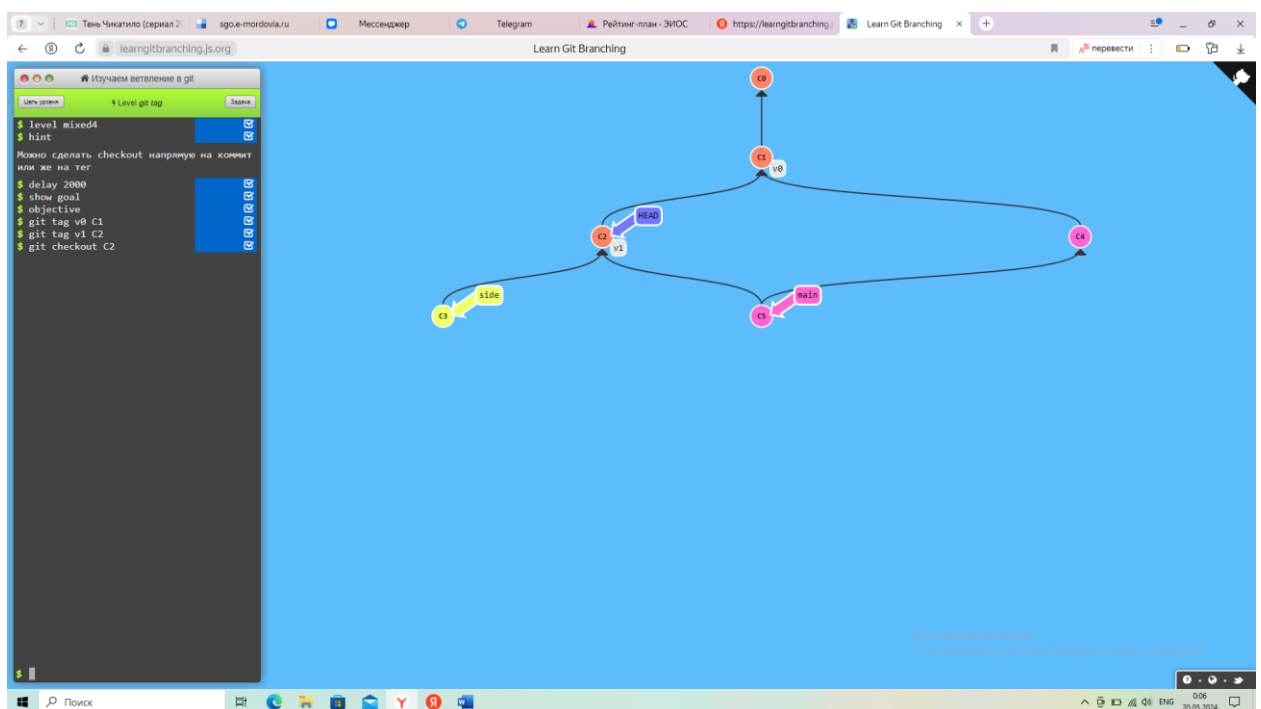
```
$ level mixed3
$ hint
Не забудь переместить main на последние изменения.
$ delay 2000
$ show goal
$ git checkout main
$ git cherry-pick C2
$ git checkout C1
$ git cherry-pick C2
$ git cherry-pick C3
$ git branch -f main C3'
```



4.4

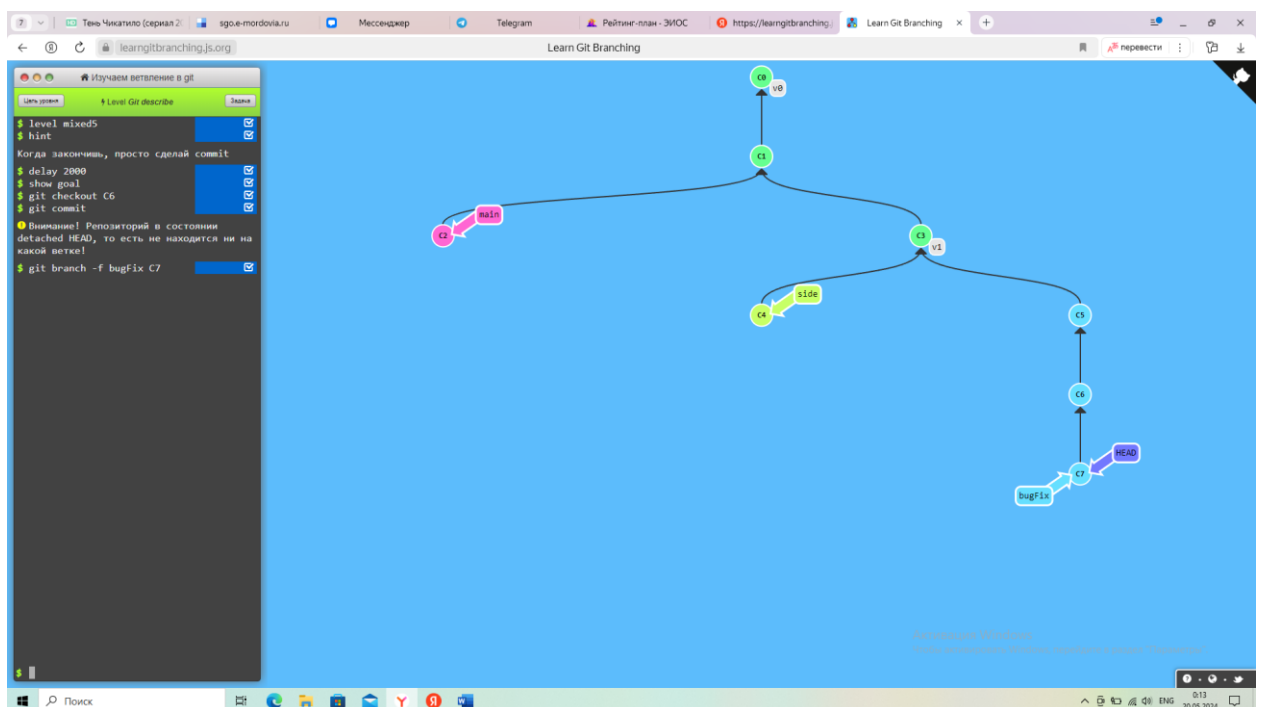
```

$ level mixed4
$ hint
Можно сделать checkout напрямую на коммит или же на тег
$ delay 2000
$ show goal
$ objective
$ git tag v0 C1
$ git tag v1 C2
$ git checkout C2
  
```



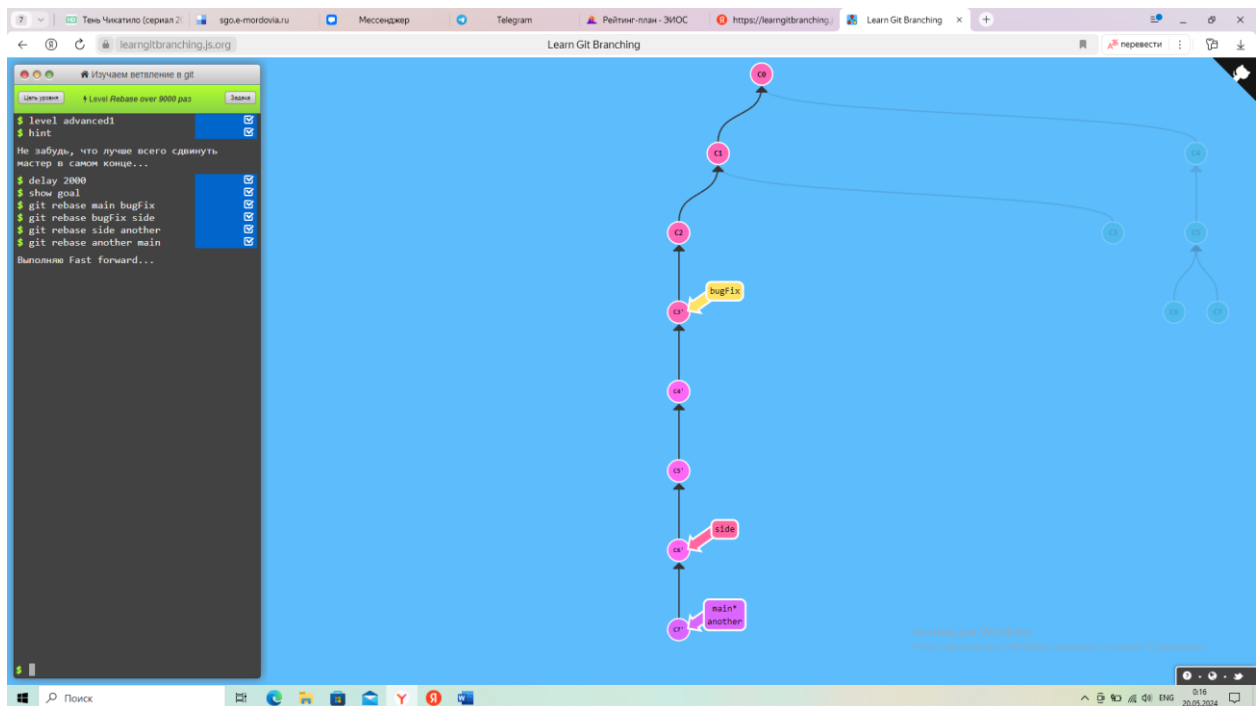
4.5

```
$ level mixed5
$ hint
Когда закончишь, просто сделай commit
$ delay 2000
$ show goal
$ git checkout C6
$ git commit
Внимание! Репозиторий в состоянии detached HEAD, то есть не
находится ни на какой ветке!
$ git branch -f bugFix C7
```



5.1

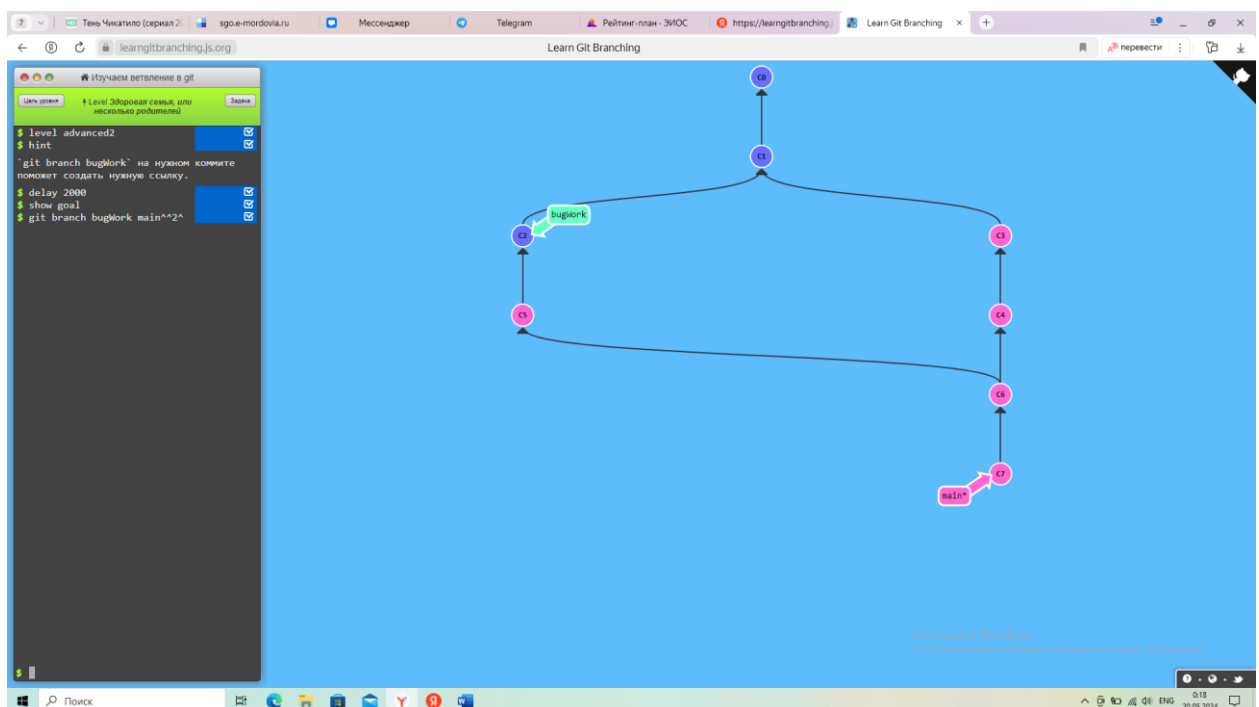
```
$ level advanced1
$ hint
Не забудь, что лучше всего сдвинуть мастер в самом конце...
$ delay 2000
$ show goal
$ git rebase main bugFix
$ git rebase bugFix side
$ git rebase side another
$ git rebase another main
Выполняю Fast forward...
```



5.2

```

$ level advanced2
$ hint
`git branch bugWork` на нужном коммите поможет создать нужную
ссылку.
$ delay 2000
$ show goal
$ git branch bugWork main^^2^
  
```



5.3

```

$ level advanced3
$ hint
Убедись, что у нас всё по порядку! Сначала ветка `one`, потом
`two` и только потом `three`
$ delay 2000
$ show goal
$ git checkout C1
$ git cherry-pick C4
$ git checkout C4'
$ git cherry-pick C3
$ git cherry-pick C2
$ git checkout C1
$ git cherry-pick C5
$ git cherry-pick C4
$ git cherry-pick C3
$ git cherry-pick C2
$ git branch -f one C2'
$ git branch -f three C2
$ git branch -f two C2''

```

