

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ**

**Отчет о лабораторной работе №1.3 по дисциплине основы программной
инженерии**

Выполнил:
Шальнев Владимир Сергеевич,
2 курс, группа ПИЖ-б-о-20-1,

Проверил:
Доцент кафедры
прикладной математики и
компьютерной безопасности,
Воронкин Р.А.

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2021 г.

ВЫПОЛНЕНИЕ:

```
C:\Users\V0V41K-ПК>F:
F:\>cd F:\pythonProject
F:\pythonProject>git clone https://github.com/HAXF13D/laboratory-3
Cloning into 'laboratory-3'...
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 7 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (7/7), done.
```

Рисунок 1.1 – Клонирование репозитория

```
F:\pythonProject\laboratory-3>git add 1.txt
F:\pythonProject\laboratory-3>git commit -m "add 1.txt file"
[main 1ea989d] add 1.txt file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 1.txt
```

Рисунок 1.2 – Индексирование файла 1.txt и его коммит

```
F:\pythonProject\laboratory-3>git add .
F:\pythonProject\laboratory-3>git commit --amend -m "add 2.txt and 3.txt"
[main e018320] add 2.txt and 3.txt
Date: Thu Sep 23 21:44:44 2021 +0300
3 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 1.txt
create mode 100644 2.txt
create mode 100644 3.txt
```

Рисунок 1.3 – Индексация файлов 2.txt и 3.txt и изменение комитта

```
F:\pythonProject\laboratory-3>git branch my_first_branch
F:\pythonProject\laboratory-3>git checkout my_first_branch
Switched to branch 'my_first_branch'
```

Рисунок 1.4 – создание ветки my_first_branch и переход на нее

```
F:\pythonProject\laboratory-3>git add .
F:\pythonProject\laboratory-3>git commit -m "in_branch.txt"
[my_first_branch b086ea7] in_branch.txt
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 in_branch.txt
```

Рисунок 1.5 – коммит изменений

```
F:\pythonProject\laboratory-3>git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)
```

Рисунок 1.6 – переход на ветку main

```
F:\pythonProject\laboratory-3>git checkout -b new_branch
Switched to a new branch 'new_branch'
```

Рисунок 1.7 – создание и переход на ветку new_branch

```
F:\pythonProject\laboratory-3>git add .  
  
F:\pythonProject\laboratory-3>git commit -m "new row in the 1.txt file"  
[new_branch 2d05070] new row in the 1.txt file  
1 file changed, 1 insertion(+)
```

Рисунок 1.8 – коммит изменений

```
F:\pythonProject\laboratory-3>git checkout main  
Switched to branch 'main'  
Your branch is ahead of 'origin/main' by 1 commit.  
(use "git push" to publish your local commits)  
  
F:\pythonProject\laboratory-3>git merge my_first_branch  
Updating e018320..b086ea7  
Fast-forward  
in_branch.txt | 0  
1 file changed, 0 insertions(+), 0 deletions(-)  
create mode 100644 in_branch.txt  
  
F:\pythonProject\laboratory-3>git merge new_branch  
Merge made by the 'recursive' strategy.  
1.txt | 1 +  
1 file changed, 1 insertion(+)
```

Рисунок 1.9 – переход на ветку main и merge файлов

```
F:\pythonProject\laboratory-3>git branch -d my_first_branch  
Deleted branch my_first_branch (was b086ea7).  
  
F:\pythonProject\laboratory-3>git branch -d new_branch  
Deleted branch new_branch (was 2d05070).
```

Рисунок 1.10 – удаление веток my_first_branch и new_branch

```
F:\pythonProject\laboratory-3>git branch branch_1  
  
F:\pythonProject\laboratory-3>git branch branch_2  
  
F:\pythonProject\laboratory-3>git checkout branch_1  
Switched to branch 'branch_1'
```

Рисунок 1.11 – создание веток и переход на ветку branch_1

```
F:\pythonProject\laboratory-3>git add .  
  
F:\pythonProject\laboratory-3>git commit -m "fix commit 1.txt and 3.txt"  
[branch_1 4218f7c] fix commit 1.txt and 3.txt  
2 files changed, 2 insertions(+), 1 deletion(-)
```

Рисунок 1.12 – коммит изменений

```
F:\pythonProject\laboratory-3>git checkout branch_2  
Switched to branch 'branch_2'
```

Рисунок 1.13 – переход на ветку branch_2

```
F:\pythonProject\laboratory-3>git add .  
  
F:\pythonProject\laboratory-3>git commit -m "My fix commit 1.txt and 3.txt"  
[branch_2 6e32c4c] My fix commit 1.txt and 3.txt  
2 files changed, 2 insertions(+), 1 deletion(-)
```

Рисунок 1.14 – коммит изменений

```
F:\pythonProject\laboratory-3>git checkout branch_1
Switched to branch 'branch_1'

F:\pythonProject\laboratory-3>git merge branch_2
Auto-merging 3.txt
CONFLICT (content): Merge conflict in 3.txt
Auto-merging 1.txt
CONFLICT (content): Merge conflict in 1.txt
Automatic merge failed; fix conflicts and then commit the result.
```

Рисунок 1.15 – переход на ветку branch_1 и merge

```
<<<<<< HEAD
fix in the 1.txt
=====
My fix in the 1.txt
>>>>>> branch_2
|
```

Рисунок 1.16 – решение конфликта в ручном режиме

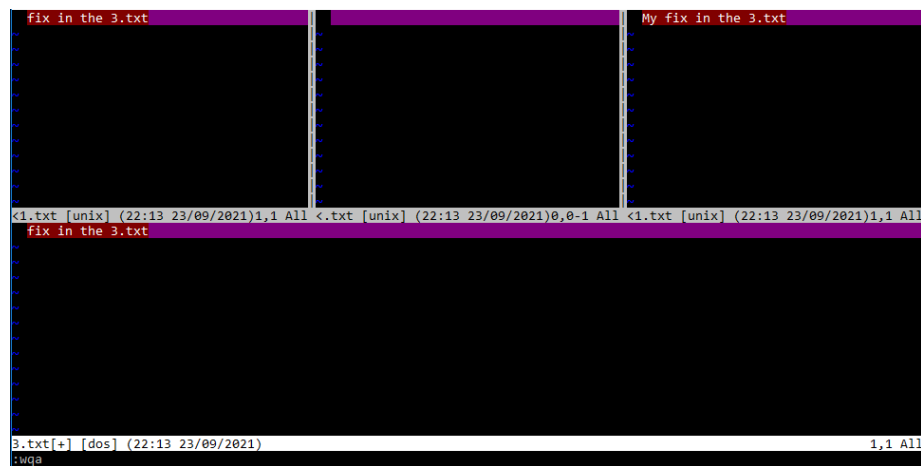


Рисунок 1.17 – решение конфликта с помощью mergetool

```
F:\pythonProject\laboratory-3>git commit -m "Commit merge"
[branch_1 7b40d2b] Commit merge
```

Рисунок 1.18 – Коммит

```
F:\pythonProject\laboratory-3>git push origin branch_1
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 6 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (14/14), 1.19 KiB | 1.19 MiB/s, done.
Total 14 (delta 5), reused 0 (delta 0)
remote: Resolving deltas: 100% (5/5), done.
remote:
remote: Create a pull request for 'branch_1' on GitHub by visiting:
remote:   https://github.com/HAXF13D/laboratory-3/pull/new/branch_1
remote:
To https://github.com/HAXF13D/laboratory-3
 * [new branch]      branch_1 -> branch_1
```

Рисунок 1.19 – отправка ветки branch_1

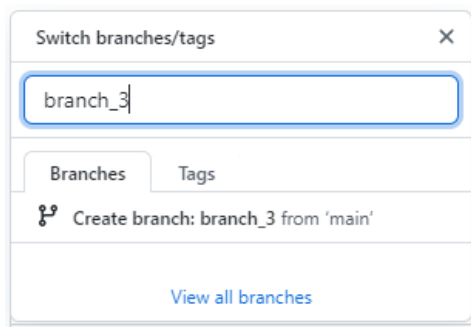


Рисунок 1.20 – создание ветки средствами GitHub

```
F:\pythonProject\laboratory-3>git fetch --all
Fetching origin

F:\pythonProject\laboratory-3>git checkout branch_3
Switched to a new branch 'branch_3'
Branch 'branch_3' set up to track remote branch 'branch_3' from 'origin'.
```

Рисунок 1.21 – создание ветки отслеживания и переход на неё

```
F:\pythonProject\laboratory-3>git add .

F:\pythonProject\laboratory-3>git commit -m "Add 2.txt on branch_3"
[branch_3 ffa53a9] Add 2.txt on branch_3
1 file changed, 1 insertion(+)
create mode 100644 2.txt
```

Рисунок 1.22 – коммит файлов на ветке branch_3

```
F:\pythonProject\laboratory-3>git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 4 commits.
(use "git push" to publish your local commits)

F:\pythonProject\laboratory-3>git rebase branch_2
First, rewinding head to replay your work on top of it...
Fast-forwarded main to branch_2.
```

Рисунок 1.23 – перемещение ветки main на ветку branch_2

```

F:\pythonProject\laboratory-3>git push origin branch_2
Total 0 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'branch_2' on GitHub by visiting:
remote:   https://github.com/HAXF13D/laboratory-3/pull/new/branch_2
remote:
To https://github.com/HAXF13D/laboratory-3
 * [new branch]      branch_2 -> branch_2

F:\pythonProject\laboratory-3>git push origin main
Total 0 (delta 0), reused 0 (delta 0)
To https://github.com/HAXF13D/laboratory-3
 fbc1a93..6e32c4c  main -> main

F:\pythonProject\laboratory-3>git push origin branch_3
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 6 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 346 bytes | 346.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/HAXF13D/laboratory-3
 fbc1a93..ffa53a9  branch_3 -> branch_3

```

Рисунок 1.24 – Отправка изменений на удаленный репозиторий

Default branch				
main	Updated 30 minutes ago by HAXF13D		Default	
Your branches				
branch_3	Updated 9 minutes ago by HAXF13D	5 1	New pull request	
branch_2	Updated 30 minutes ago by HAXF13D	0 0	New pull request	
branch_1	Updated 16 minutes ago by HAXF13D	0 2	New pull request	
Active branches				
branch_3	Updated 9 minutes ago by HAXF13D	5 1	New pull request	
branch_1	Updated 16 minutes ago by HAXF13D	0 2	New pull request	
branch_2	Updated 30 minutes ago by HAXF13D	0 0	New pull request	

Рисунок 1.25 – проверка изменений на GitHub

ОТВЕТЫ НА ВОПРОСЫ:

1. Что такое ветка?

Ветка в Git — это простой перемещаемый указатель на один из таких коммитов. Коммит — содержит метаданные и указатель на объект дерева каталогов.

2. Что такое HEAD?

HEAD — это указатель, задача которого ссылаться на определенный коммит в репозитории. HEAD — это указатель на коммит в вашем репозитории, который станет родителем следующего коммита.

HEAD указывает на коммит, относительно которого будет создана рабочая копия в процессе операции checkout.

3. Способы создания веток.

Первый вариант – командой `git branch <branch_name>`

Второй вариант – используя GitHub

4. Как узнать текущую ветку?

Прописав команду `git branch`, звездочкой будет отмечена текущая ветка.

5. Как переключаться между ветками?

Используя команду `git checkout <branch_name>`

6. Что такое удаленная ветка?

Удалённые ссылки — это ссылки (указатели) в ваших удалённых репозиториях, включая ветки, теги и так далее.

7. Что такое ветка отслеживания?

Ветки слежения — это ссылки на определённое состояние удалённых веток.

8. Как создать ветку отслеживания?

Командой – `git checkout --track <remote>/<branch>`

Или используя команды

`git fetch`

`git checkout <branch>`

Или создать ветку с названием отличным от данной

`git checkout -b <branch> <remote>/<branch>`

9. Как отправить изменения из локальной ветки в удаленную ветку?

`git push <remote> <branch>`

10. В чем отличие команд `git fetch` и `git pull`?

`git fetch` – просто получает с сервера все изменения, которых у вас ещё нет, но не будет изменять состояние вашей рабочей директории.

`git pull` – является командой `git fetch` с тем отличием, что после `git fetch` происходит команда `git merge`.

11. Как удалить локальную и удаленную ветки?

Для удаления локальной ветки необходимо написать команду `git branch -d <branch>`, если необходимо удалить ветку с несохраненными изменениями необходимо прописать команду `git branch -D <branch>`. Для удаления удаленной ветки необходимо прописать команду `git push <remote> --delete <branch>`

12. Изучить модель ветвления `git-flow` (использовать материалы статей <https://www.atlassian.com/ru/git/tutorials/comparing-workflows/gitflow-workflow>, <https://habr.com/ru/post/106912/>). Какие основные типы веток присутствуют в модели `git-flow`? Как организована работа с ветками в модели `git-flow`? В чем недостатки `git-flow`?

В нём присутствуют такие ветки как «feature», «release» и «hotfix» прописать команду «`git flow init`»

```
git flow feature start MYFEATURE
git flow release start RELEASE [BASE]
git flow release publish RELEASE
git flow release track RELEASE
git flow release finish RELEASE
git flow hotfix start VERSION [BASENAME]
git flow hotfix finish VERSION
```

1. Git Flow может замедлять работу, когда приходится ревьюить большие пулл реквесты, когда вы пытаетесь выполнить итерацию быстро.
2. Релизы сложно делать чаще, чем раз в неделю.
3. Большие функции могут потратить дни на мерж и резолв конфликтов и форсировать несколько циклов тестирования.
4. История проекта в гите имеет кучу merge commits и затрудняет просмотр реальной работы.
5. Может быть проблематичным в CI/CD сценариях.

13. На прошлой лабораторной работе было задание выбрать одно из программных средств с GUI для работы с Git. Необходимо в рамках этого вопроса привести описание инструментов для работы с ветками Git, предоставляемых этим средством.

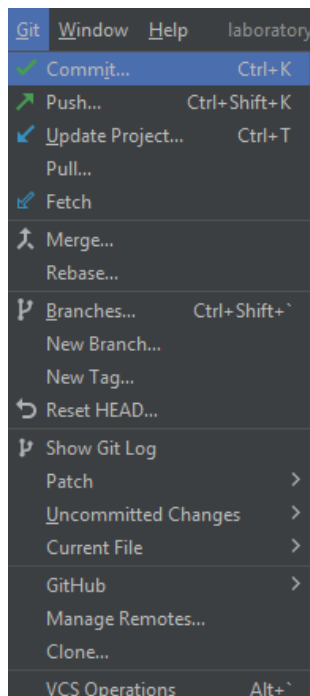


Рисунок 1.26 – окно управления Git Pycharm

В данном окне видны все базовые функции git.

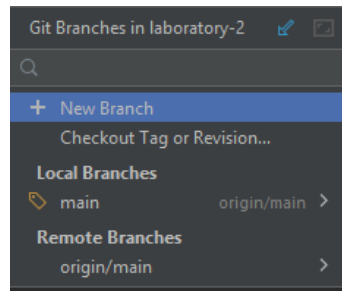


Рисунок 1.27 – окно веток в Pycharm

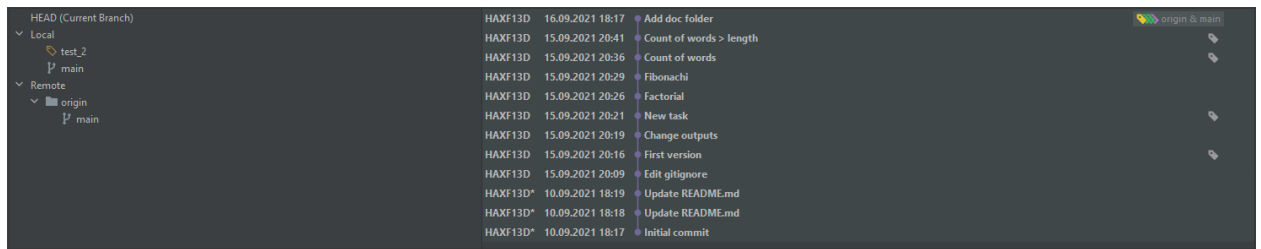


Рисунок 1.28 – окно истории коммитов

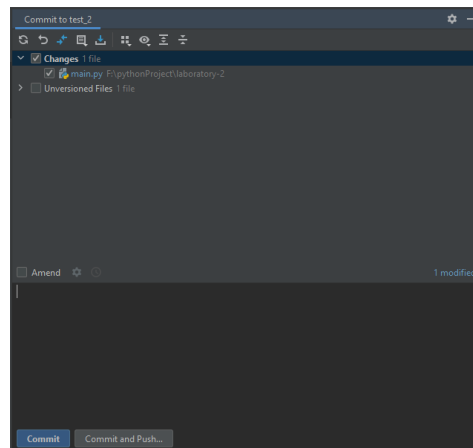


Рисунок 1.29 – окно подтверждения коммита

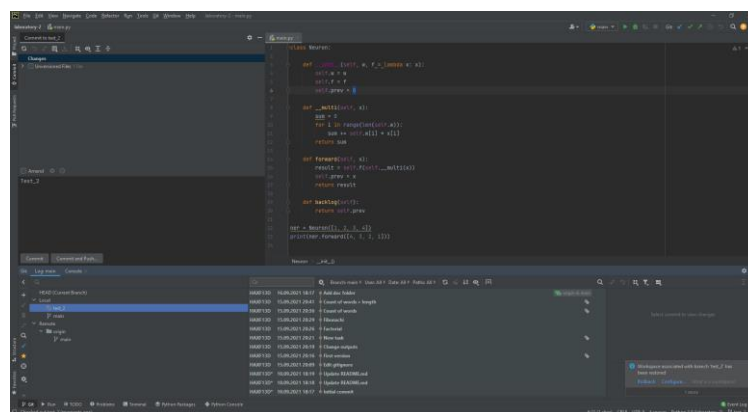


Рисунок 1.30 – ветка test_2

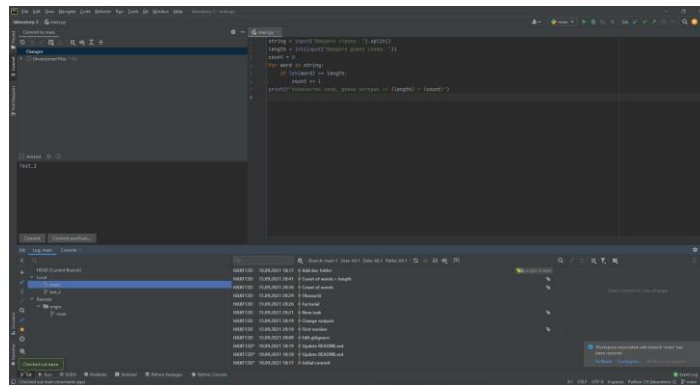


Рисунок 1.31 – ветка main

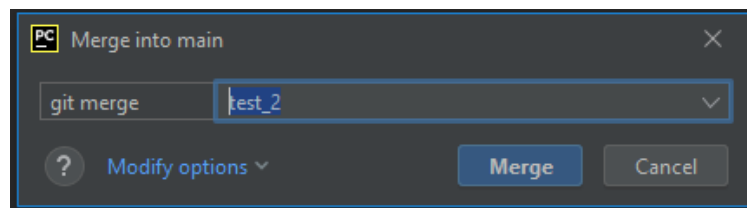


Рисунок 1.32 – окно merge ветки test_2 в ветку main

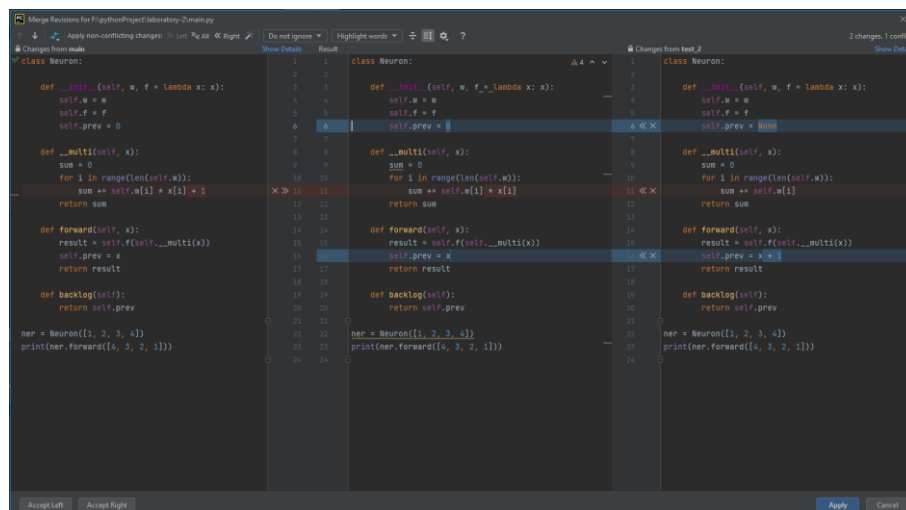


Рисунок 1.33 – окно слияния двух веток