

Лабораторная работа №3

По дисциплине: Основы программной инженерии

Духно Михаил

ПИЖ-б-о-20-1

```
C:\Users\я\Desktop\Папки\Программирование\Python\lessons>git add README.md test.rb LICENSE
C:\Users\я\Desktop\Папки\Программирование\Python\lessons>git commit -m "initial commit of my project"
[main 5fabfee] initial commit of my project
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 test.rb
```

Рисунок 3.1 – Инициализация коммита проекта

```
C:\Users\я\Desktop\Папки\Программирование\Python\lessons>git log --oneline
5fabfee (HEAD -> main) initial commit of my project
55820e8 (origin/main, origin/HEAD) 8 git
fd3ce85 7 git
37352db 6 git
145b1fe 5 git
8b2c9e3 4 git
47d8daf 3 git
d75f64b 2 git
c26e119 1 git
7f6e4bb Initial commit
```

Рисунок 3.2 – Просмотр истории изменений

```
C:\Users\я\Desktop\Папки\Программирование\Python\lessons>git checkout origin
Note: switching to 'origin'.
```

Рисунок 3.3 – Переход на репозиторий origin

```
C:\Users\я\Desktop\Папки\Программирование\Python\lessons>git log --oneline
55820e8 (HEAD, origin/main, origin/HEAD) 8 git
fd3ce85 7 git
37352db 6 git
145b1fe 5 git
8b2c9e3 4 git
47d8daf 3 git
d75f64b 2 git
c26e119 1 git
7f6e4bb Initial commit
```

Рисунок 3.4 – Просмотр истории изменений

```
C:\Users\я\Desktop\Папки\Программирование\Python\lessons>git checkout fd3ce85
Previous HEAD position was 55820e8 8 git
HEAD is now at fd3ce85 7 git
```

Рисунок 3.5 – Переход на репозиторий fd3ce85

```
C:\Users\я\Desktop\Папки\Программирование\Python\lessons>git log --oneline
fd3ce85 (HEAD) 7 git
37352db 6 git
145b1fe 5 git
8b2c9e3 4 git
47d8daf 3 git
d75f64b 2 git
c26e119 1 git
7f6e4bb Initial commit
```

Рисунок 3.6 – Просмотр истории изменений

```
C:\Users\я\Desktop\Папки\Программирование\Python\lessons>git checkout 37352db
Previous HEAD position was fd3ce85 7 git
HEAD is now at 37352db 6 git
```

Рисунок 3.7 – переход на репозиторий 37352

```
C:\Users\я\Desktop\Папки\Программирование\Python\lessons>git log --oneline
37352db (HEAD) 6 git
145b1fe 5 git
8b2c9e3 4 git
47d8daf 3 git
d75f64b 2 git
c26e119 1 git
7f6e4bb Initial commit
```

Рисунок 3.8 – Просмотр истории изменений

```
C:\Users\я\Desktop\Папки\Программирование\Python\lessons\.git>dir
Том в устройстве C имеет метку OS
Серийный номер тома: F66A-7E04

Содержимое папки C:\Users\я\Desktop\Папки\Программирование\Python\lessons\.git

16.10.2021 12:49          29 COMMIT_EDITMSG
10.09.2021 18:11        294 config
10.09.2021 18:11        73 description
16.10.2021 12:53        41 HEAD
10.09.2021 18:11    <DIR>      hooks
16.10.2021 12:53       361 index
10.09.2021 18:11    <DIR>      info
10.09.2021 18:11    <DIR>      logs
16.10.2021 12:49    <DIR>      objects
10.09.2021 18:11       112 packed-refs
10.09.2021 18:11    <DIR>      refs
                   6 файлов          910 байт
                   5 папок   130 968 555 520 байт свободно
```

Рисунок 3.9 – Команда dir

```
C:\Users\я\Desktop\Папки\Программирование\Python\lessons\.git>type HEAD
37352db2d2b736b441c08dab22ef06467f393532
```

Рисунок 3.10 – Просмотр HEAD

```
C:\Users\я\Desktop\Папки\Программирование\Python\lessons>git branch testing
C:\Users\я\Desktop\Папки\Программирование\Python\lessons>git log --oneline --decorate
37352db (HEAD, testing) 6 git
145b1fe 5 git
8b2c9e3 4 git
47d8daf 3 git
d75f64b 2 git
c26e119 1 git
7f6e4bb Initial commit
```

Рисунок 3.11 – Переход на ветку testing, просмотр истории изменений

```
C:\Users\я\Desktop\Папки\Программирование\Python\lessons>git merge hotfix
Already up to date.

C:\Users\я\Desktop\Папки\Программирование\Python\lessons>git branch -d hotfix
Deleted branch hotfix (was 55820e8).
```

Рисунок 3.12 – слияние кода при помощи git merge и удаление ветки

Контрольные вопросы:

1. Что такое ветка?

Ветка в Git — это простой перемещаемый указатель на один из таких коммитов. По умолчанию, имя основной ветки в Git — master . Как только вы начнёте создавать коммиты, ветка master будет всегда указывать на последний коммит.

2. Что такое HEAD?

HEAD в Git - это указатель на текущую ссылку на ветвь, которая, в свою очередь, является указателем на последний сделанный вами коммит или последний коммит, который был извлечен в ваш рабочий каталог. Это также означает, что это будет родитель следующего коммита, который вы делаете.

3. Способы создания веток.

git checkout -b <имя ветки>

4. Как узнать текущую ветку?

Узнать текущую ветку в Git можно путем просмотра списка веток через команду `git branch`

5. Как переключаться между ветками?

используйте команду `git checkout` для переключения между ветками

6. Что такое удаленная ветка?

Удалённые ветки действуют как закладки для напоминания о том, где ветки в удалённых репозиториях находились во время последнего подключения к ним. Они выглядят как [имя удал. репоз.]/[ветка]

7. Что такое ветка отслеживания?

Ветки слежения — это ссылки на определённое состояние удалённых веток.

8. Как создать ветку отслеживания?

`git branch --set-upstream-to=<имя ветки>`

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

`git push origin dev`

`git push --set-upstream origin dev`

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

`git pull` — это, по сути, команда `git fetch`, после которой сразу же следует `git merge`.

Команда `git fetch` получает изменения с сервера и сохраняет их в каталог `refs/remotes/`. Это действие (`fetch`) не влияет на локальные ветки и текущие изменения, просто изменения с удаленного сервера скачиваются в директорию локального репозитория.

Git merge сливает код в указанной репозитории

11. Как удалить локальную и удаленную ветки?

`git branch -d` для локальных

`git push -d` для удалённых

12. Изучить модель ветвления git-flow (использовать материалы статей

[https://www.atlassian.c](https://www.atlassian.com/ru/git/tutorials/comparing-workflows/gitflow-workflow)

[om/ru/git/tutorials/comparing-workflows/gitflow-workflow](https://www.atlassian.com/ru/git/tutorials/comparing-workflows/gitflow-workflow), [https://habr.com/ru/post/10691](https://habr.com/ru/post/106912/)

2/). Какие основные типы веток присутствуют в модели git-flow? Как организована работа с ветками в модели git-flow? В чем недостатки git-flow?

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.