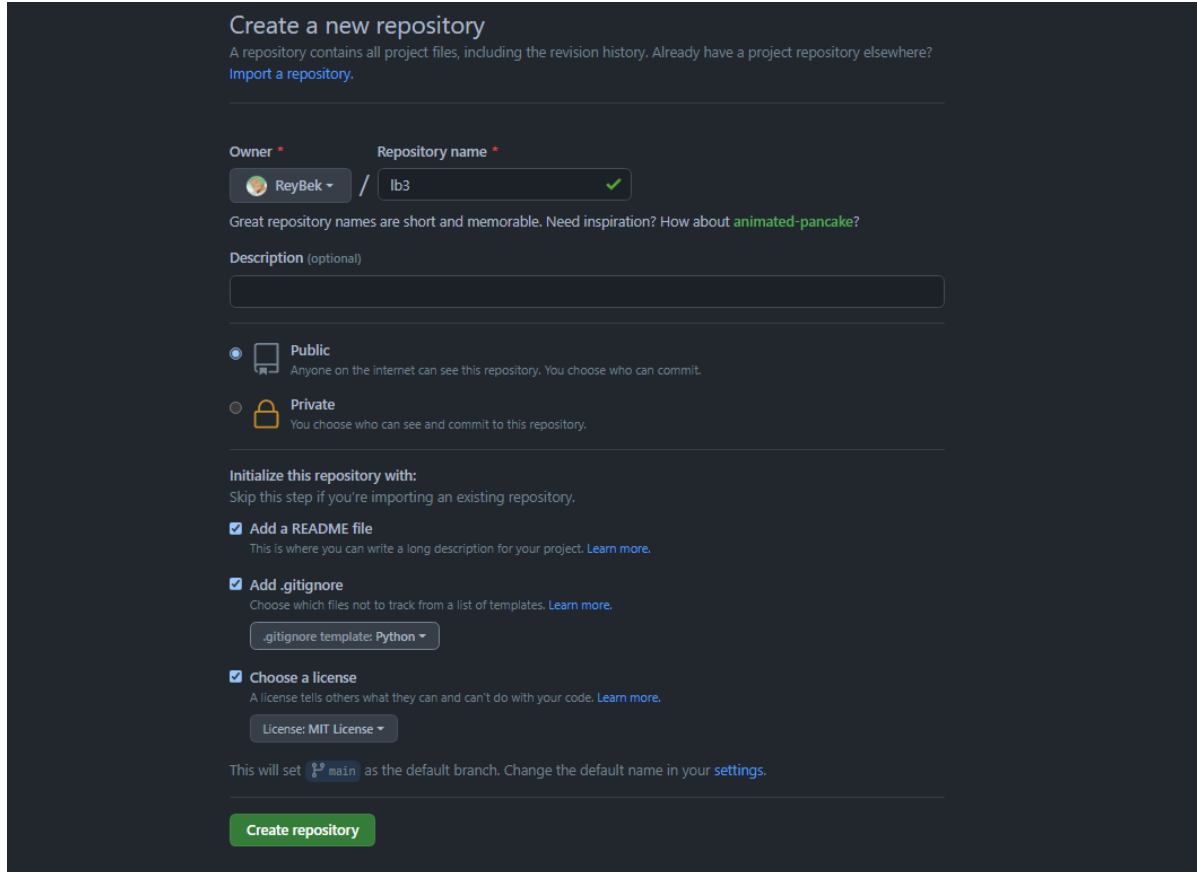


Отчет №3 по дисциплине
“Основы программной инженерии”
Группа: ПИЖ-б-о-20-1, Бекишев М. М.

ВЫПОЛНЕНИЕ



1) Создание репозитория:



Create a new repository


A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)


Owner ^{*} Repository name ^{*}

 ReyBek / lb3 

Great repository names are short and memorable. Need inspiration? How about [animated-pancake?](#)

Description (optional)

☒  **Public**
Anyone on the Internet can see this repository. You choose who can commit.


☐  **Private**
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

☒ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

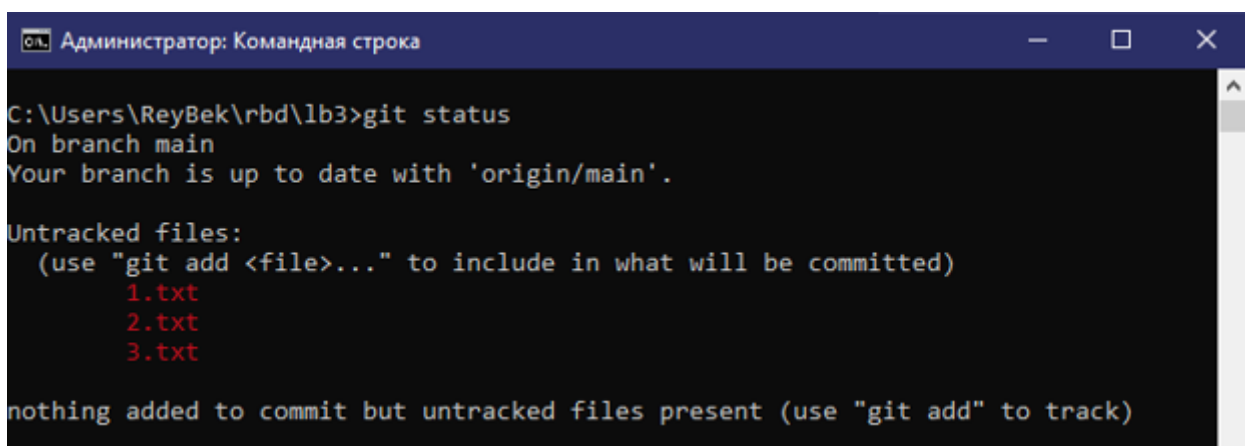
☒ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)

☒ **Choose a license**
A license tells others what they can and can't do with your code. [Learn more.](#)

This will set  **main** as the default branch. Change the default name in your [settings](#).

[Create repository](#)

2) Добавление 1.txt, 2.txt, 3.txt.



```
Администратор: Командная строка

C:\Users\ReyBek\rbd\lb3>git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        1.txt
        2.txt
        3.txt

nothing added to commit but untracked files present (use "git add" to track)
```

Отчет №3 по дисциплине
“Основы программной инженерии”
Группа: ПИЖ-б-о-20-1, Бекишев М. М.

3) Комит 1.txt

```
C:\Users\ReyBek\rbd\lb3>git add 1.txt

C:\Users\ReyBek\rbd\lb3>git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   1.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        2.txt
        3.txt

C:\Users\ReyBek\rbd\lb3>git commit -m "add 1.txt"
[main da9d9df] add 1.txt
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 1.txt
```

4) Перезапись коммита с добавлением 2.txt и 3.txt

```
Администратор: Командная строка

C:\Users\ReyBek\rbd\lb3>git add 2.txt

C:\Users\ReyBek\rbd\lb3>git add 3.txt

C:\Users\ReyBek\rbd\lb3>git commit --amend -m "add 1.txt"
[main 4057013] add 1.txt
Date: Mon Jan 31 00:01:49 2022 +0300
3 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 1.txt
create mode 100644 2.txt
create mode 100644 3.txt

C:\Users\ReyBek\rbd\lb3>git push --force
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 347 bytes | 347.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/ReyBek/lb3.git
9b54754..4057013  main -> main
```

5) Создание ветки

Отчет №3 по дисциплине
“Основы программной инженерии”
Группа: ПИЖ-б-о-20-1, Бекишев М. М.

```
Администратор: Командная строка

C:\Users\ReyBek\rbd\lb3>git branch my_first_branch

C:\Users\ReyBek\rbd\lb3>git status
On branch main
Your branch and 'origin/main' have diverged,
and have 1 and 1 different commits each, respectively.
(use "git pull" to merge the remote branch into yours)

nothing to commit, working tree clean
```

6) Создание файла в ветке

```
Администратор: Командная строка

checkout

C:\Users\ReyBek\rbd\lb3>git checkout my_first_branch
Switched to branch 'my_first_branch'

C:\Users\ReyBek\rbd\lb3>git add .

C:\Users\ReyBek\rbd\lb3>git commit -m "add in_bracnh"
[my_first_branch fda0308] add in_bracnh
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 in_branch.txt
```

7) Создание новой ветки и переключение на нее

```
Администратор: Командная строка

C:\Users\ReyBek\rbd\lb3>git branch new_branch

C:\Users\ReyBek\rbd\lb3>git checkout new_branch
Switched to branch 'new_branch'
```

8) Коммит изменённого файла 1.txt

```
Администратор: Командная строка

C:\Users\ReyBek\rbd\lb3>git add 1.txt

C:\Users\ReyBek\rbd\lb3>git commit -m "new row in the 1.txt file"
[new_branch 6da6742] new row in the 1.txt file
1 file changed, 1 insertion(+)
```

Отчет №3 по дисциплине
“Основы программной инженерии”
Группа: ПИЖ-б-о-20-1, Бекишев М. М.

9) Переключение на ветку main и слияние с my_first_branch и new_branch.

```
C:\Users\ReyBek\rbd\lb3>git merge my_first_branch
Already up to date.

C:\Users\ReyBek\rbd\lb3>git merge new_branch
Updating fda0308..6da6742
Fast-forward
 1.txt | 1 +
 1 file changed, 1 insertion(+)
```

10) Удаление веток my_first_branch и new_branch

```
Администратор: Командная строка

C:\Users\ReyBek\rbd\lb3>git branch -d my_first_branch
Deleted branch my_first_branch (was fda0308).

C:\Users\ReyBek\rbd\lb3>git branch -d new_branch
Deleted branch new_branch (was 6da6742).

C:\Users\ReyBek\rbd\lb3>git push
Everything up-to-date
```

11) Создать ветки branch_1 и branch_2.

```
Администратор: Командная строка

C:\Users\ReyBek\rbd\lb3>git branch branch_1

C:\Users\ReyBek\rbd\lb3>git branch branch_2

C:\Users\ReyBek\rbd\lb3>
```

12) Изменение и коммит файлов на ветке branch_1

```
Администратор: Командная строка

C:\Users\ReyBek\rbd\lb3>git branch branch_1

C:\Users\ReyBek\rbd\lb3>git branch branch_2

C:\Users\ReyBek\rbd\lb3>git checkout branch_1
Switched to branch 'branch_1'

C:\Users\ReyBek\rbd\lb3>git add 1.txt

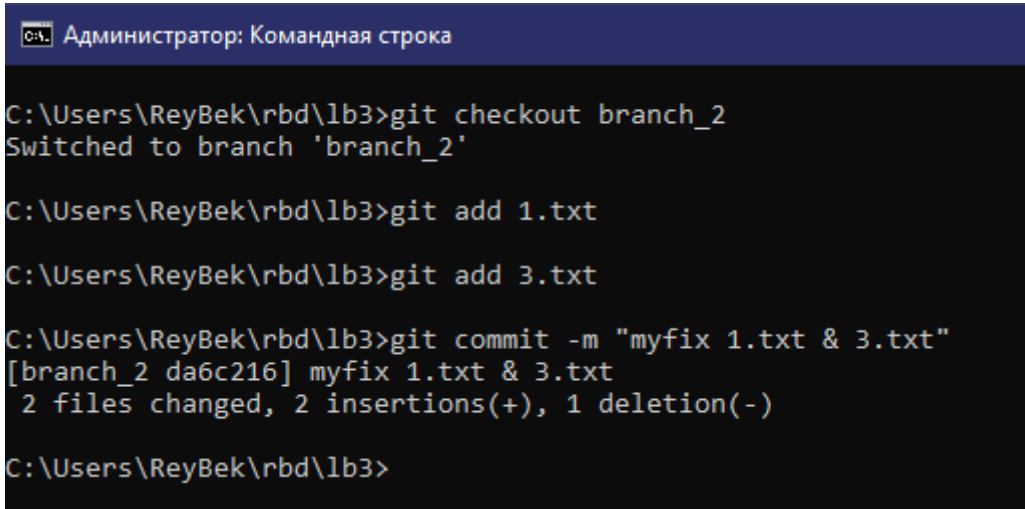
C:\Users\ReyBek\rbd\lb3>git add 3.txt

C:\Users\ReyBek\rbd\lb3>git commit -m "fix 1.txt & 3.txt"
[branch_1 1b5b25d] fix 1.txt & 3.txt
 2 files changed, 2 insertions(+), 1 deletion(-)

C:\Users\ReyBek\rbd\lb3>
```

Отчет №3 по дисциплине
“Основы программной инженерии”
Группа: ПИЖ-б-о-20-1, Бекишев М. М.

13) Изменение и коммит фалов на ветке branch_2



```
Администратор: Командная строка

C:\Users\ReyBek\rbd\lb3>git checkout branch_2
Switched to branch 'branch_2'

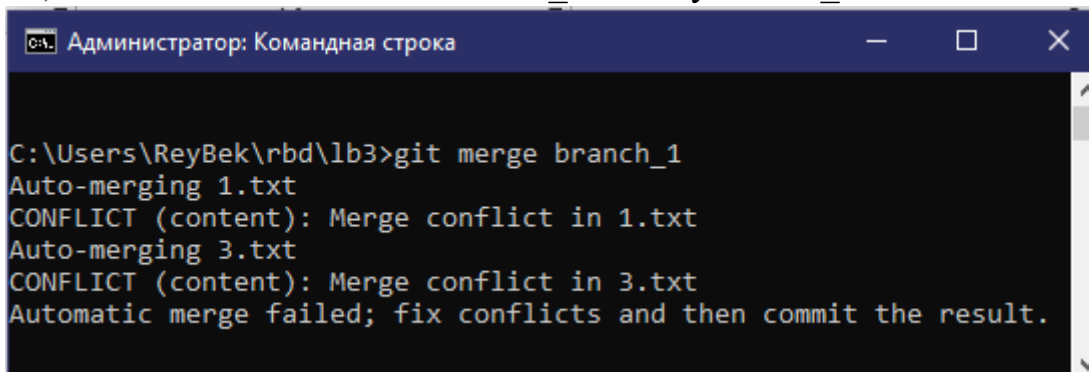
C:\Users\ReyBek\rbd\lb3>git add 1.txt

C:\Users\ReyBek\rbd\lb3>git add 3.txt

C:\Users\ReyBek\rbd\lb3>git commit -m "myfix 1.txt & 3.txt"
[branch_2 da6c216] myfix 1.txt & 3.txt
 2 files changed, 2 insertions(+), 1 deletion(-)

C:\Users\ReyBek\rbd\lb3>
```

14) Слить изменения ветки branch_2 в ветку branch_1

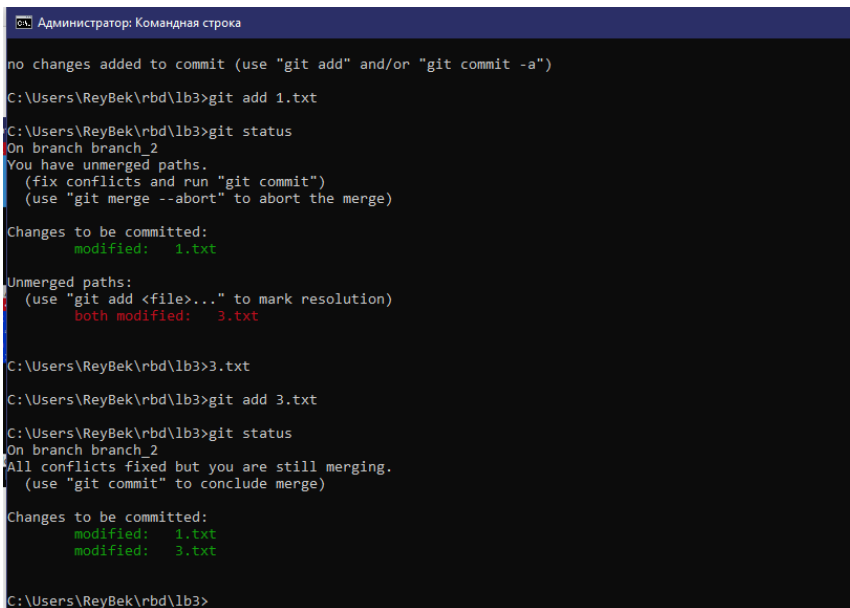
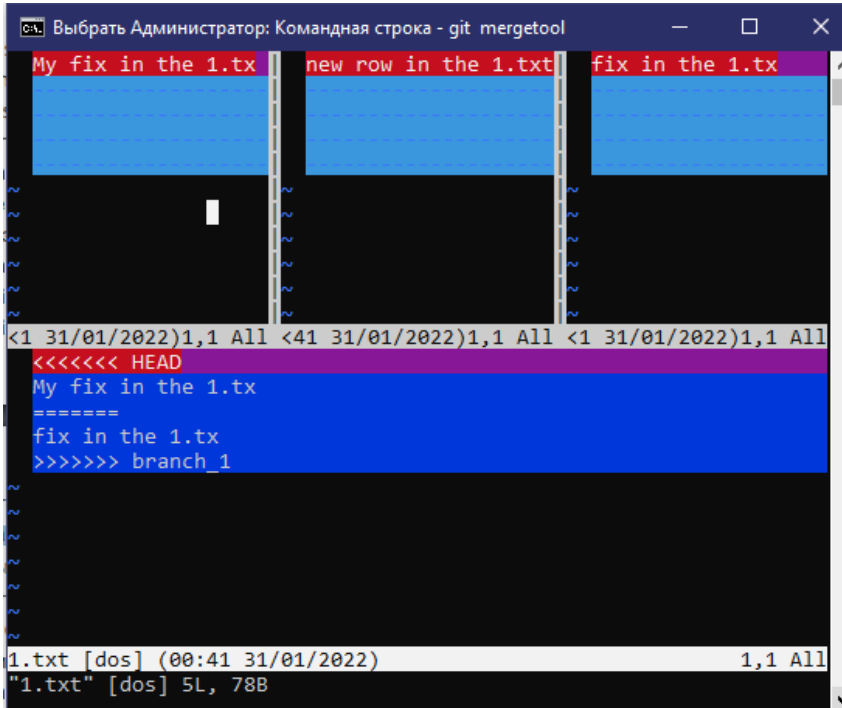


```
Администратор: Командная строка

C:\Users\ReyBek\rbd\lb3>git merge branch_1
Auto-merging 1.txt
CONFLICT (content): Merge conflict in 1.txt
Auto-merging 3.txt
CONFLICT (content): Merge conflict in 3.txt
Automatic merge failed; fix conflicts and then commit the result.
```

15) Решение конфликта слияния

Отчет №3 по дисциплине
“Основы программной инженерии”
Группа: ПИЖ-б-о-20-1, Бекишев М. М.



16) Отправка branch_1 на GitHub

```
C:\Users\ReyBek\rbd\lb3>git push --set-upstream origin branch_2
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (10/10), 795 bytes | 795.00 KiB/s, done.
Total 10 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
remote:
remote: Create a pull request for 'branch_2' on GitHub by visiting:
remote:   https://github.com/ReyBek/lb3/pull/new/branch_2
remote:
To https://github.com/ReyBek/lb3.git
 * [new branch]      branch_2 -> branch_2
Branch 'branch_2' set up to track remote branch 'branch_2' from 'origin'
```

ОТВЕТЫ НА КОТРОЛЬНЫЕ ВОПРОСЫ

1. Ветка – это простой перемещаемый указатель на один из коммитов
2. HEAD – это указатель, задача которого ссылаться на определенный коммит в репозитории. Определенный коммит тут, это или родитель следующего коммита, или коммит, относительно которого будет создана рабочая копия во время операции checkout.
3. Ветку можно создать командой `git branch` (создание ветки, но не переключение), `git checkout -b` или через github.
4. Узнать текущую ветку можно переключившись на неё командой `git checkout`.
5. Переключиться на ветку можно командой `git checkout`. Если добавить параметр `-b`, то ветка не только переключится, но и создастся.
6. Удаленная ветка – это ссылка (указатель) в удаленном репозитории, которая включает ветки, теги и т.д.
7. Ветки отслеживания – это локальные ветки, которые связаны и удаленной веткой
8. Ветку отслеживания можно создать через команду `–track`
9. Изменения из локальной ветки в удаленную можно отправить по команде `git push`
10. `Git fetch` получает все изменения, которых у вас еще нет, но не будет изменять состояние рабочей директории. Команда `git fetch` – это грубо говоря порядок команд `git fetch` и `git merge`
11. Удалить удаленную ветку можно через команду `git push origin –delete`. Удалить локальную ветку можно через команду `git branch -d`
12. Типы веток `git flow`: функциональная, магистральная, ветка разработки, главная ветка, ветки выпуска, ветки исправления. В `Git-flow` используется больше веток, каждая из которых существует дольше, а коммиты обычно крупнее. Разработчики создают функциональную ветку и откладывают ее слияние с главной магистральной веткой до завершения работы над функцией. `Git-flow` не слишком удобно применять в процессах CI/CD.
13. В SM ветки представлены в виде линий разного цвета. По ним удобно смотреть, когда ветки были созданы, слиты, изменены и т.д. Для переключения веток есть отдельное окно.

Отчет №3 по дисциплине
“Основы программной инженерии”
Группа: ПИЖ-б-о-20-1, Бекишев М. М.