

**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное
учреждение высшего образования

«Московский технический университет связи и информатики»

Факультет «Информационные технологии»

Кафедра «Искусственный интеллект и машинное обучение»

Лабораторная работа №10
Практическая работа с Git и GitHub, часть 2.

Автор:

Голиков Михаил Вячеславович, БВТ2402

Цель лабораторной работы

Познакомить студентов с основными операциями работы с удаленными репозиториями в Git, включая клонирование, добавление удаленных репозитория, отправку изменений и работу с ветками на удаленном репозитории.

Ход выполнения лабораторной работы

Задание 1: скопируем случайный удалённый репозиторий без авторских прав (<https://github.com/shreysrins/bond-calculator> в моём случае) на наше устройство, используя git clone. Убедимся в успешности выполнения программы.

```
E:\Downloads>cd Lab_10

E:\Downloads\Lab_10>git clone https://github.com/shreysrins/bond-calculator
Cloning into 'bond-calculator'...
remote: Enumerating objects: 107, done.
remote: Counting objects: 100% (107/107), done.
remote: Compressing objects: 100% (76/76), done.
Receiving objects: 43% (47/107)remote: Total 107 (delta 61), reused 64 (delta 30), pack-reused 0 (from 0)
Receiving objects: 100% (107/107), 20.17 KiB | 2.24 MiB/s, done.
Resolving deltas: 100% (61/61), done.

E:\Downloads\Lab_10>cd bond-calculator

E:\Downloads\Lab_10\bond-calculator>dir
Том в устройстве E имеет метку Workspace
Серийный номер тома: DED4-F283

Содержимое папки E:\Downloads\Lab_10\bond-calculator

03.02.2025  20:13    <DIR>          .
03.02.2025  20:13    <DIR>          ..
03.02.2025  20:13             3 143 .gitignore
03.02.2025  20:13             7 918 bond_calc.py
03.02.2025  20:13             1 096 LICENSE
03.02.2025  20:13             1 400 README.md
03.02.2025  20:13             50 requirements.txt
               5 файлов             13 607 байт
               2 папок   922 802 106 368 байт свободно
```

Элемент 1 — Задание 1

Задания 2 и 3: добавим новый удалённый репозиторий для нашего проекта. Для этого сперва удалим файл .git и перезапишем его с нужным репозиторием,

используя соответствующие команды. После отправим наши локальные файлы на удалённый репозиторий.

```
E:\Downloads\Lab_10>git clone https://github.com/shreysrins/bond-calculator
Cloning into 'bond-calculator'...
remote: Enumerating objects: 107, done.
remote: Counting objects: 100% (107/107), done.
remote: Compressing objects: 100% (76/76), done.
Receiving objects: 83% (89/107)remote: Total 107 (delta 61), reused 64 (delta 30), pack-reused 0 (from 0)
Receiving objects: 100% (107/107), 20.17 KiB | 5.04 MiB/s, done.
Resolving deltas: 100% (61/61), done.

E:\Downloads\Lab_10>cd bond-calculator

E:\Downloads\Lab_10\bond-calculator>del .git
E:\Downloads\Lab_10\bond-calculator\git\*, вы уверены [Y(да)/N(нет)]? Y

E:\Downloads\Lab_10\bond-calculator>git init
Initialized empty Git repository in E:/Downloads/Lab_10/bond-calculator/.git/

E:\Downloads\Lab_10\bond-calculator>git remote add origin https://github.com/Prototype721/lab_10.git

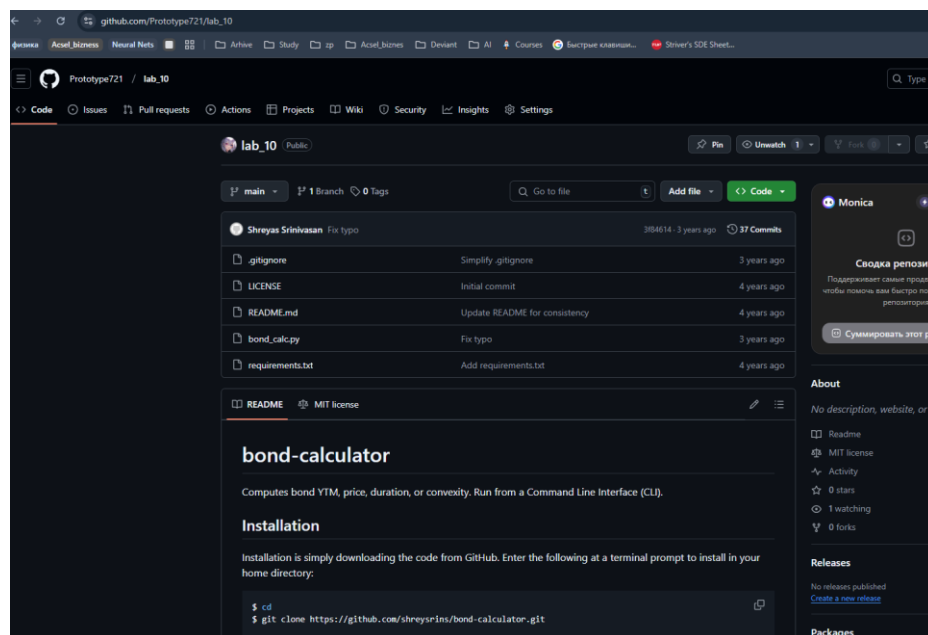
E:\Downloads\Lab_10\bond-calculator>git branch -M main

E:\Downloads\Lab_10\bond-calculator>git push origin main
Enumerating objects: 107, done.
Counting objects: 100% (107/107), done.
Delta compression using up to 12 threads
Compressing objects: 100% (45/45), done.
Writing objects: 100% (107/107), 20.17 KiB | 20.17 MiB/s, done.
Total 107 (delta 61), reused 107 (delta 61), pack-reused 0
remote: Resolving deltas: 100% (61/61), done.
To https://github.com/Prototype721/lab_10.git
 * [new branch]      main -> main

E:\Downloads\Lab_10\bond-calculator>
```

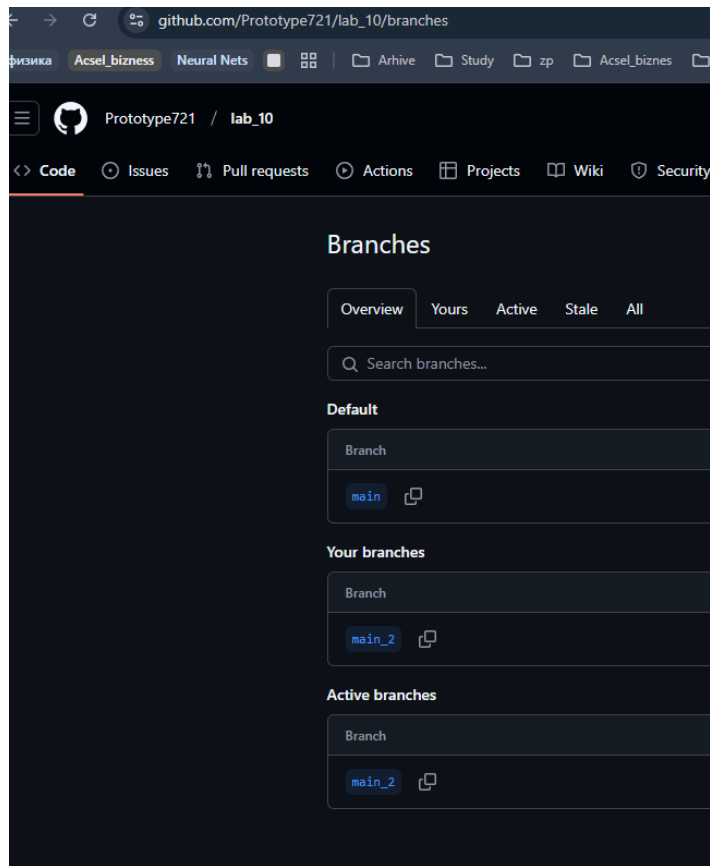
Элемент 2 — Задания 2 и 3 в терминале

Как видно на элементе 3, мы успешно скопировали наши локальные файлы на удалённый репозиторий.

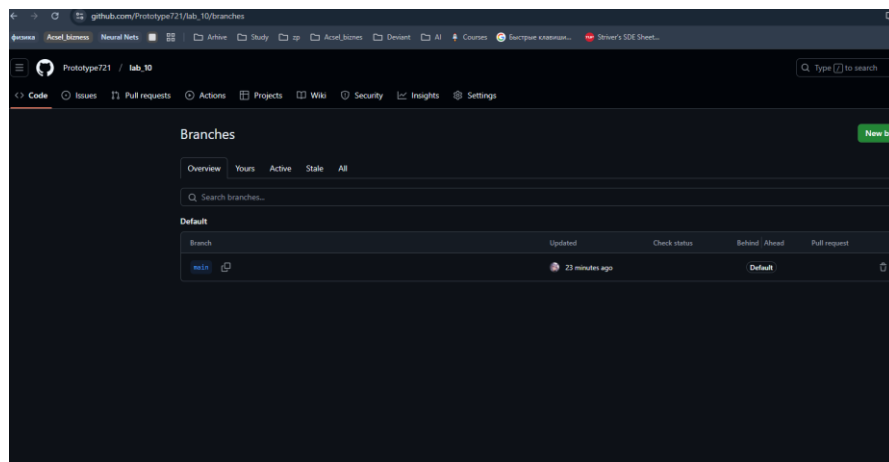


Элемент 3 — Задания 2 и 3 на Github – проверка

Задание 4: создадим новую ветку `main_2` в нашем локальном репозитории с помощью команды `git checkout -b main_2`. Отправим эту ветку в удалённый репозиторий, после чего удалим его удалённо (элементы 4 и 5).



Элемент 4 — Задание 4 на Github – создание ветки



Элемент 5 — Задание 4 на Github – удаление ветки

```

E:\Downloads\Lab_10\bond-calculator>git checkout -b main_2
Switched to a new branch 'main_2'

E:\Downloads\Lab_10\bond-calculator>git push origin main_2
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'main_2' on GitHub by visiting:
remote:   https://github.com/Prototype721/lab_10/pull/new/main_2
remote:
To https://github.com/Prototype721/lab_10.git
 * [new branch]      main_2 -> main_2

E:\Downloads\Lab_10\bond-calculator>git push origin --delete main
To https://github.com/Prototype721/lab_10.git
! [remote rejected] main (refusing to delete the current branch: refs/heads/main)
error: failed to push some refs to 'https://github.com/Prototype721/lab_10.git'

E:\Downloads\Lab_10\bond-calculator>git push origin --delete main_2
To https://github.com/Prototype721/lab_10.git
- [deleted]          main_2

E:\Downloads\Lab_10\bond-calculator>_

```

Элемент 6 — Задание 4 в терминале

Задание 5: получим изменения из удалённого репозитория на локальный с помощью команды `git pull origin main`. Так как у нас не было изменений, то `git` написал “Already up to date” (элемент 7). Если файлы на локальном и глобальном репозитории разнятся, то результат можно увидеть на элементе 8.

```

E:\Downloads\Lab_10\bond-calculator>git pull origin main
From https://github.com/Prototype721/lab_10
 * branch            main            -> FETCH_HEAD
Already up to date.

E:\Downloads\Lab_10\bond-calculator>

```

Элемент 7 — Задание 5 в терминале (обновления не было)

```

$ git pull origin master
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 655 bytes | 3.00 KiB/s, done.
From https://github.com/akashadr/Git
 * branch            master          -> FETCH_HEAD
  5a457ea..4913939  master          -> origin/master
Merge made by the 'ort' strategy.
New.txt | 1 +
1 file changed, 1 insertion(+)
create mode 100644 New.txt

```

Элемент 8 — Задание 5 в терминале (с обновлением)

Ниже представлены все команды git, связанные с темой лабораторной работы:

`git clone https://github.com/youre_hub_link` — клонирует git репозиторий на ваш ПК

`git init` — создание пустого git репозитория на вашем ПК

`git remote add origin https://github.com/youre_hub_link` — добавляет удаленный репозитория в локальный.

`git push origin https://github.com/youre_hub_link` — отправка изменений из локального репозитория на удаленный.

`git status` — выводит необходимую информацию о текущем состоянии индексации

`git commit -m "Text"` — закрепляет текущее состояние индексации в отдельный комит, добавляет к нему уникальный `commit_id`

`git pull` — обновление персонального репозитория на вашем ПК с общего репозитория

`git push` — отправляет текущее состояние на GitHub (можно отправить отдельную ветку, написав её название после push)

`git checkout` — показывает текущую ветку

`git chechout your_branch / git switch your_branch` – смена ветки

`git checkout -b new_branch_name` — создание новой ветки (локально)

`git branch -d (-D) (your_branch)` — удаление ветки (обычное и принудительное), если не писать `your_branch`, то удалит тут ветку, в которой вы находитесь

`git merge` – объединяет ветки

`git push origin --delete branch_name` — удаление ветки из удаленного репозитория

`git pull origin branch_name` — получение изменений из удаленного репозитория на локальный.

`git cherry-pick` – пытается вставить отдельный комит из другой ветки

`git rebase` — вставляет все комиты из другой ветки в вашу ветку

`git reset / revert` – отмена действия, возврат к предыдущему комиту (revert более безопасный, так как не удаляет комит, а создаёт новый комит возврата)

Элемент 9 — Git запросы

Заключение

Была изучена доскональная работа с удаленными репозиториями GitHub через систему контроля версий Git. Все задачи выполнены, результаты соответствуют ожидаемым.