Лабораторная работа №10

Практическая работа с Git и GitHub, часть 2.

Автор:

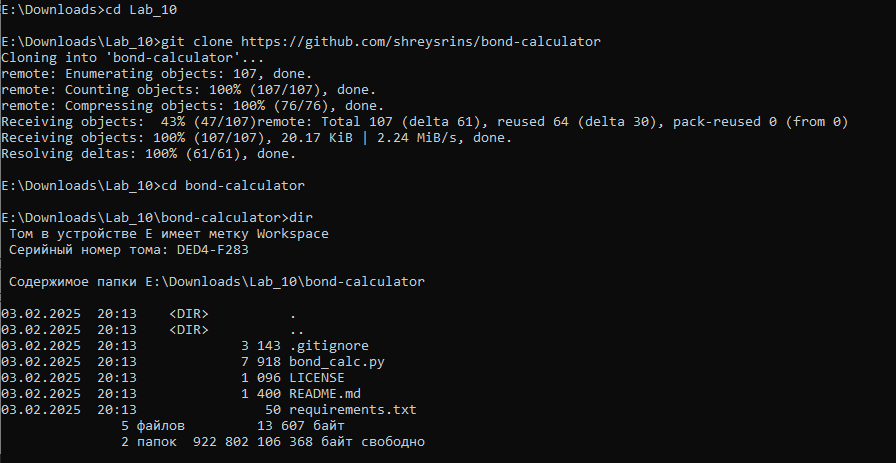
Голиков Михаил Вячеславович, БВТ2402

**Цель лабораторной работы**

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

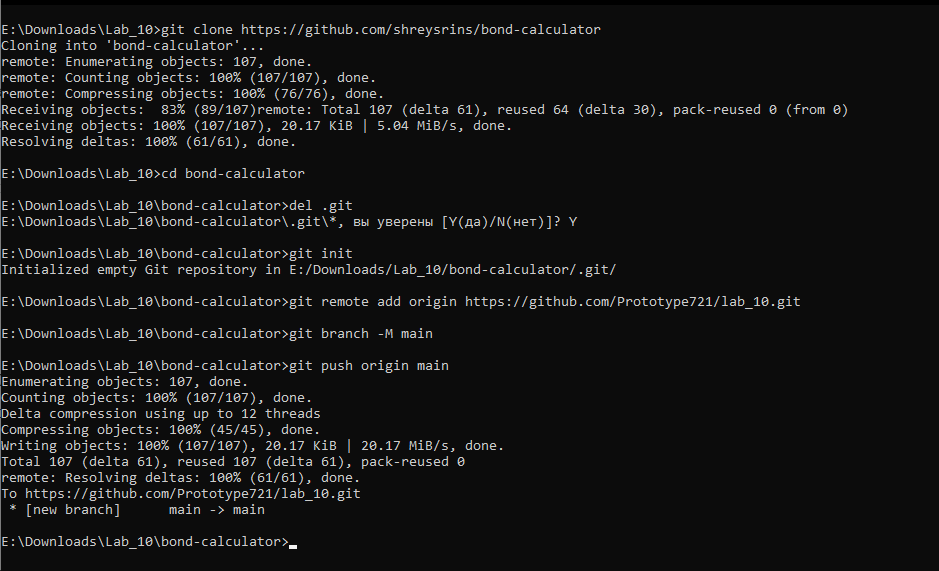
**Ход выполнения лабораторной работы**

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.



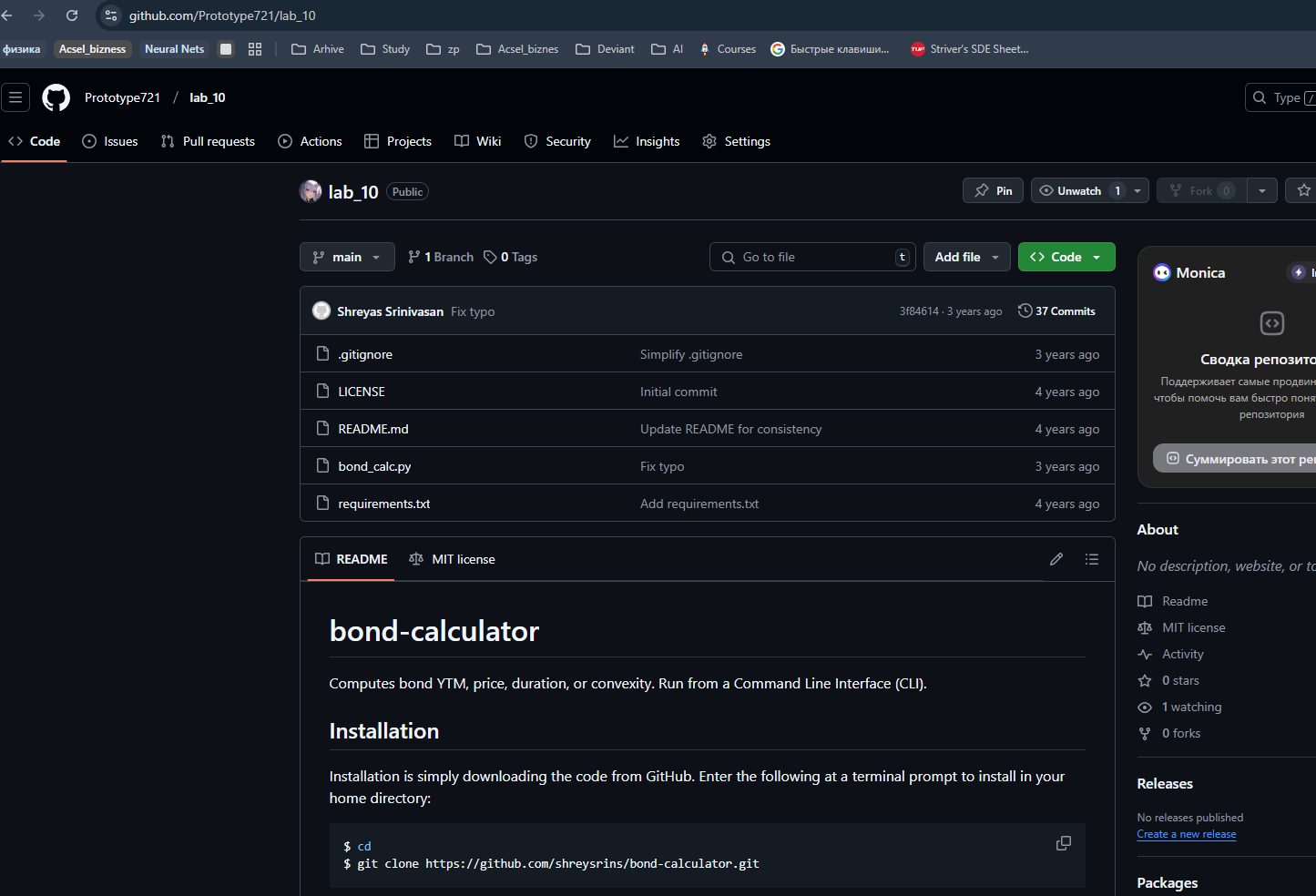
Элемент 1 ― Задание 1

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.



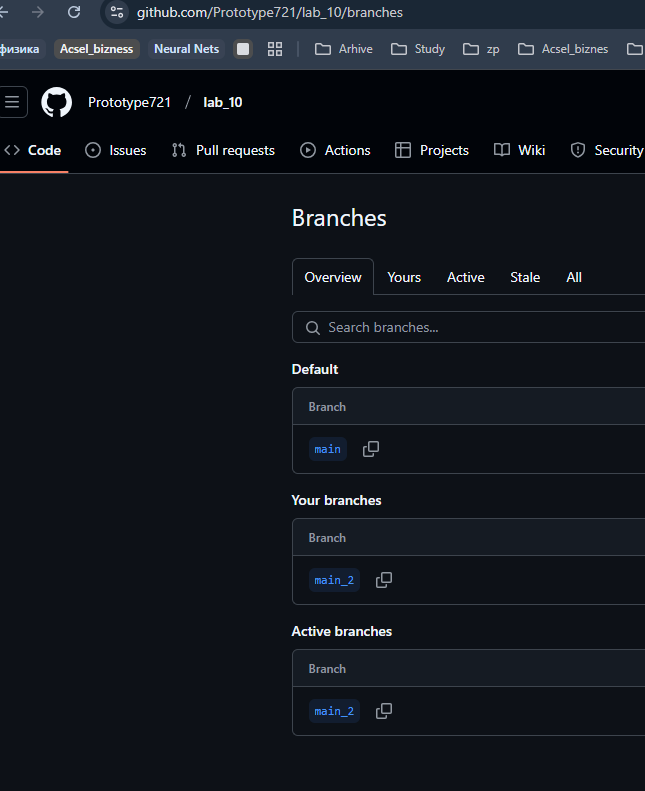
Элемент 2 ― Задания 2 и 3 в терминале

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

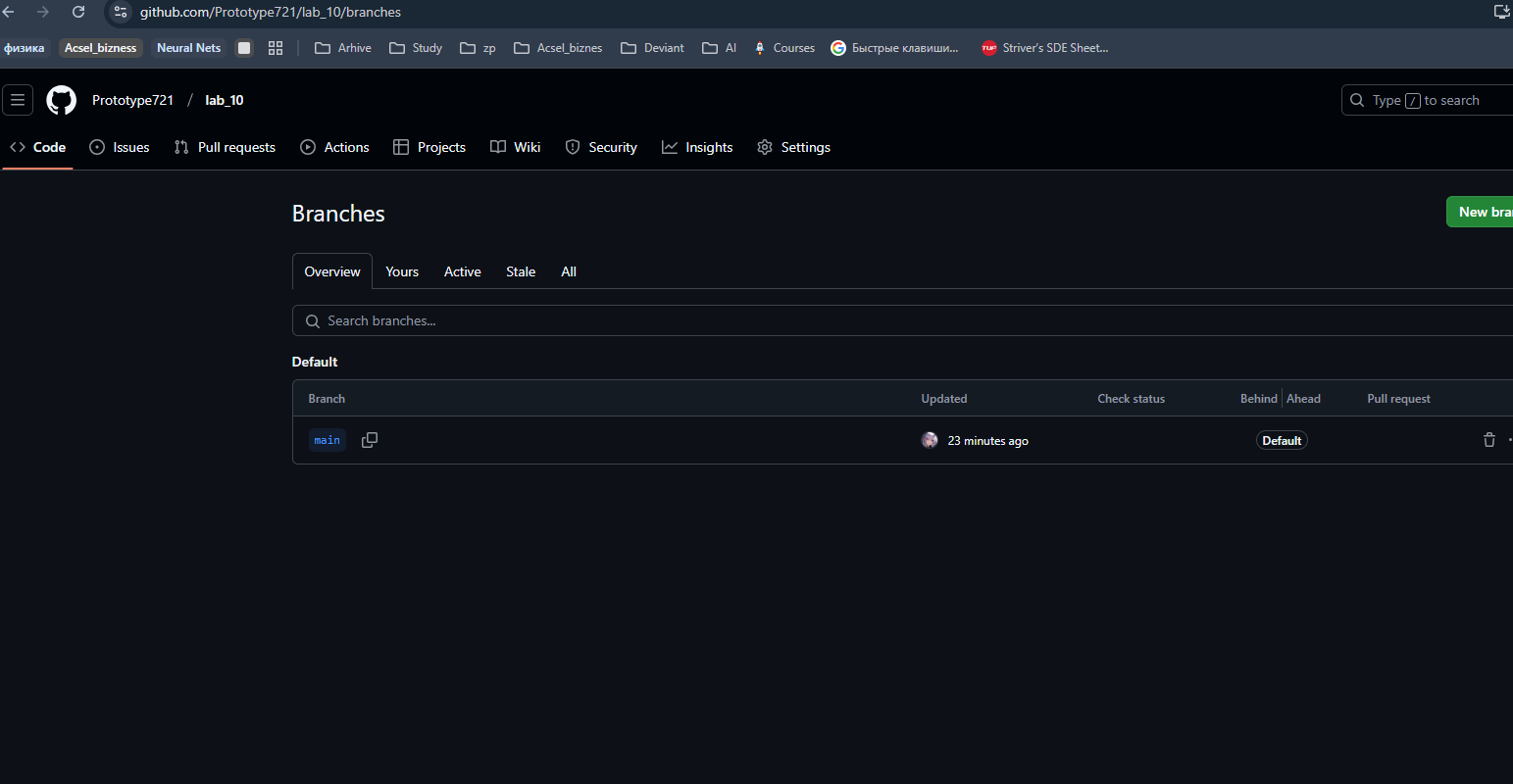


Элемент 3 ― Задания 2 и 3 на Github – проверка

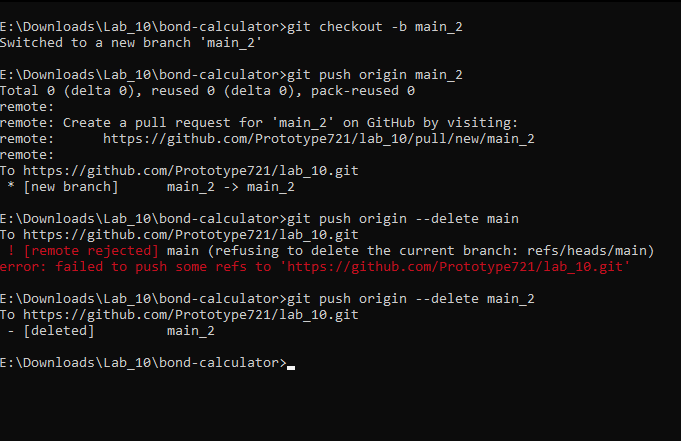
Задание 4: создадим новую ветку main\_2 в нашем локальном репозитории с помощью команды git checkout -b main\_2. Отправим эту ветку в удалённый репозиторий, после чего удалим его удалённо (элементы 4 и 5).



Элемент 4 ― Задание 4 на Github – создание ветки

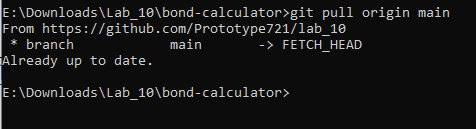


Элемент 5 ― Задание 4 на Github – удаление ветки

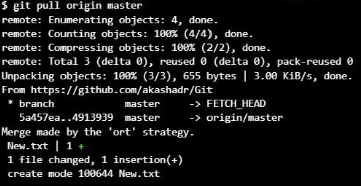


Элемент 6 ― Задание 4 в терминале

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.



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



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 безопастный, 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.