

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт перспективной инженерии
Департамент цифровых, робототехнических систем и электроники

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №2
дисциплины
«Основы кроссплатформенного программирования»

Выполнил:
Ляш Денис Александрович
2 курс, группа ИТС-б-о-23-1,
11.03.02 «Инфокоммуникационные
технологии и системы связи»,
направленность (профиль)
«Инфокоммуникационные системы и
сети»,
очная форма обучения

(подпись)

Проверил:
Доцент департамента цифровых,
робототехнических систем и
электроники Воронкин Р.А

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2024 г.

Тема: Исследование возможностей Git для работы с локальными репозиториями

Цель: исследовать базовые возможности системы контроля версий Git для работы с локальными репозиториями.

Порядок выполнения работы:

1. Были изучены основные теоретические сведения.
2. Был создан общедоступный репозиторий:

Create a new repository

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

Required fields are marked with an asterisk ().*

Owner * Zloybanan4ik / Repository name * lab2

✔ lab2 is available.

Great repository names are short and memorable. Need inspiration? How about **ideal-telegram** ?

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:

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

Add .gitignore
.gitignore template: Python

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license
License: MIT License

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

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

i You are creating a public repository in your personal account.

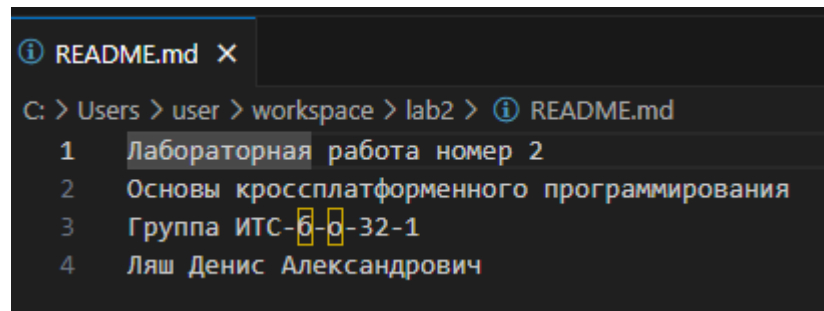
Рисунок 1. Создание общедоступного репозитория

3. Был скопирован репозиторий на компьютер:

```
user@DESKTOP-VI4I94J MINGW64 ~/workspace
$ git clone https://github.com/Zloybanan4ik/lab2.git
Cloning into 'lab2'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (5/5), done.
```

Рисунок 2. Клонирование репозитория

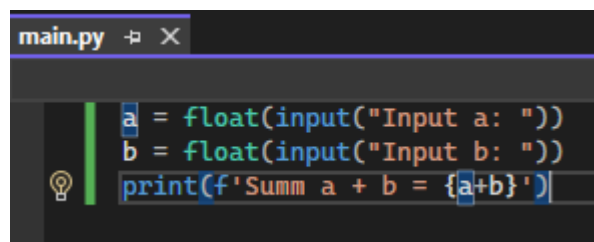
4. Была добавлена в файл README.md дополнительная информация:



```
① README.md X
C: > Users > user > workspace > lab2 > ① README.md
1 Лабораторная работа номер 2
2 Основы кроссплатформенного программирования
3 Группа ИТС-6-0-32-1
4 Ляш Денис Александрович
```

Рисунок 3. Информация в файле README

5. Была написана программа на Python и зафиксированы изменения:



```
main.py ↗ X
a = float(input("Input a: "))
b = float(input("Input b: "))
print(f'Summ a + b = {a+b}')
```

Рисунок 4. Программа, написанная на Python

```
user@DESKTOP-VI4I94J MINGW64 ~/workspace/lab2 (main)
$ git add *

user@DESKTOP-VI4I94J MINGW64 ~/workspace/lab2 (main)
$ 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)
        modified:   README.md

user@DESKTOP-VI4I94J MINGW64 ~/workspace/lab2 (main)
$ git commit -m "Изменения файла README"
[main a2ff5df] Изменения файла README
1 file changed, 4 insertions(+), 1 deletion(-)

user@DESKTOP-VI4I94J MINGW64 ~/workspace/lab2 (main)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 440 bytes | 440.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Zloybanan4ik/lab2.git
   6eaa207..a2ff5df  main -> main
```

Рисунок 5. Фиксирование изменений программы

6. Были добавили теги:

```

user@DESKTOP-VI4I94J MINGW64 ~/workspace/lab2 (main)
$ 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)
        .vs/
        main.py

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

user@DESKTOP-VI4I94J MINGW64 ~/workspace/lab2 (main)
$ git add *

user@DESKTOP-VI4I94J MINGW64 ~/workspace/lab2 (main)
$ git commit -m "Базовое сложение"
[main da5e66f] Базовое сложение
1 file changed, 3 insertions(+)
create mode 100644 main.py

user@DESKTOP-VI4I94J MINGW64 ~/workspace/lab2 (main)
$ git tag -a v1.0 -m "Сложение"

user@DESKTOP-VI4I94J MINGW64 ~/workspace/lab2 (main)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 361 bytes | 361.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Zloybanan4ik/lab2.git
   a2ff5df..da5e66f  main -> main

user@DESKTOP-VI4I94J MINGW64 ~/workspace/lab2 (main)
$ git push origin v1.0
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 181 bytes | 181.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Zloybanan4ik/lab2.git
 * [new tag]           v1.0 -> v1.0

```

Рисунок 6. Добавление тега

7. Были созданы не менее 7 коммитов:

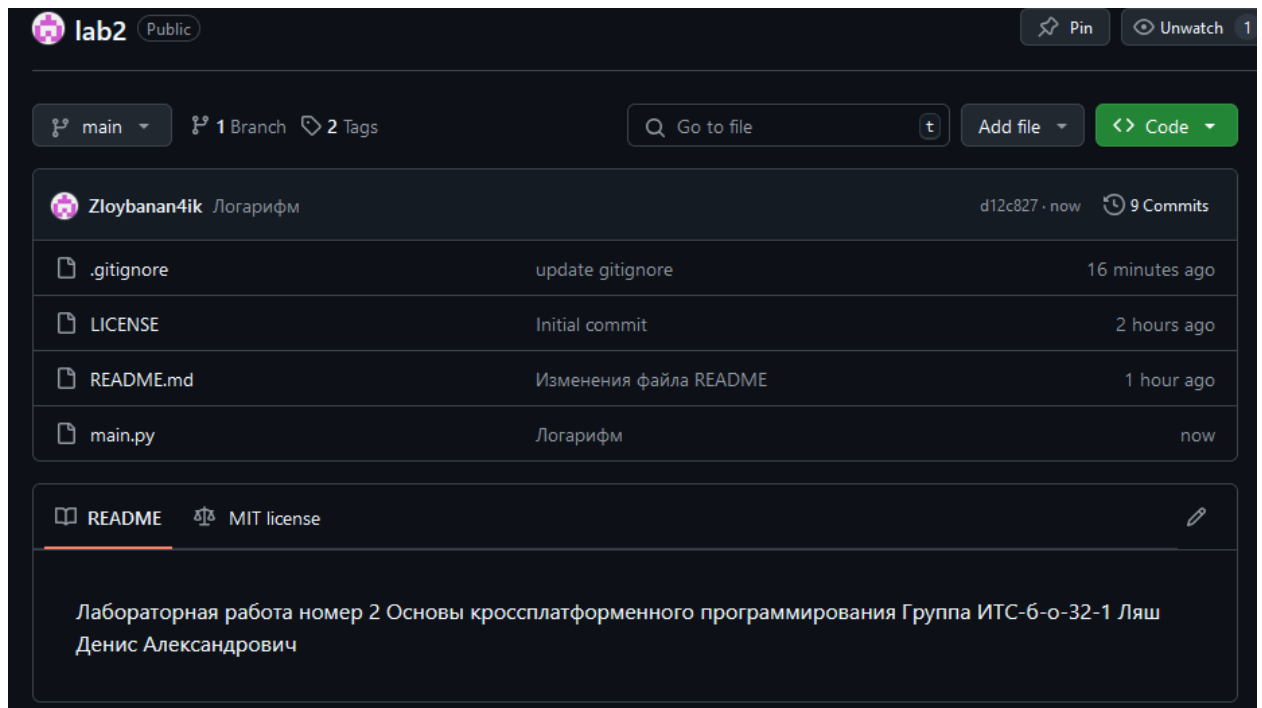


Рисунок 7. Созданы 9 коммитов

8. Была просмотрена история коммитов с помощью команды git log

```
commit d12c8279760effdccec9b3310967bdfa5c9644a5 (HEAD -> main, origin/main, origin/HEAD)
Author: Zloybanan4ik <denik55557777809@gmail.com>
Date:   Fri Dec 27 02:57:33 2024 +0300

    Логарифм

commit 032a41d7c06167f50f805115aba205c8ed9be896
Author: Zloybanan4ik <denik55557777809@gmail.com>
Date:   Fri Dec 27 02:56:34 2024 +0300

    Модуль

commit cdae12b13477a1e47ce2969260890b79cb1035fd
Author: Zloybanan4ik <denik55557777809@gmail.com>
Date:   Fri Dec 27 02:55:47 2024 +0300

    Добавлено извлечение корня

commit ed84e9a79c4004f2bdc13125180f0a2c14d1df6c (tag: v2.0)
Author: Zloybanan4ik <denik55557777809@gmail.com>
Date:   Fri Dec 27 02:48:42 2024 +0300

    Добавлено возведение в степень

commit 126e39ced9ee2a5e330ac68c59e736aa004f3f81
Author: Zloybanan4ik <denik55557777809@gmail.com>
Date:   Fri Dec 27 02:41:46 2024 +0300

    update gitignore

commit 1cc0c495e36d7d76ed92da7270c7492d596a2f6c (tag: v1.1)
Author: Zloybanan4ik <denik55557777809@gmail.com>
Date:   Fri Dec 27 02:00:22 2024 +0300

    Добавлена операция вычитания с возможностью выбора операции

commit da5e66f9bd271510cbaafe38486018d5e2d2398e (tag: v1.0)
Author: Zloybanan4ik <denik55557777809@gmail.com>
Date:   Fri Dec 27 01:51:58 2024 +0300

    Базовое сложение

commit a2ff5dfc80f5bbc0fec8e321c4e6232ef5cbeb00
Author: Zloybanan4ik <denik55557777809@gmail.com>
Date:   Fri Dec 27 01:23:01 2024 +0300

    Изменения файла README

commit 6eaa207e6aa437efbb8cd5d151c84436d4c47155
Author: Zloybanan4ik <denik55557777809@gmail.com>
Date:   Fri Dec 27 00:57:11 2024 +0300

    Initial commit
```

Рисунок 8. Просмотр истории коммитов

9. Просмотр последнего коммита с помощью команды `git show HEAD`:

```
commit d12c8279760effdccec9b3310967bdfa5c9644a5 (HEAD -> main, origin/main, origin/HEAD)
Author: Zloybanan4ik <denik55557777809@gmail.com>
Date:   Fri Dec 27 02:57:33 2024 +0300

    Логарифм

diff --git a/main.py b/main.py
index 4f8d976..8da1a3b 100644
--- a/main.py
+++ b/main.py
@@ -1,5 +1,7 @@
+import math
+
 a = float(input("Input a: "))
-s = input('Input "+", "-", "*", "/", "^", "sqrt", or "abs": ')
+s = input('Input "+", "-", "*", "/", "^", "sqrt", "abs", or "log": ')

 if s == '-':
     b = float(input("Input b: "))
@@ -23,5 +25,10 @@ elif s == 'sqrt':
     print(math.sqrt(a))
 elif s == 'abs':
     :
```

Рисунок 9. Последний коммит

10. Просмотр предпоследнего коммита:


```

user@DESKTOP-VI4I94J MINGW64 ~/workspace/lab2 (main)
$ git show HEAD~1
commit 032a41d7c06167f50f805115aba205c8ed9be896
Author: Zloybanan4ik <denik55557777809@gmail.com>
Date:   Fri Dec 27 02:56:34 2024 +0300

    Модуль

diff --git a/main.py b/main.py
index a0c0def..4f8d976 100644
--- a/main.py
+++ b/main.py
@@ -1,7 +1,5 @@
-import math
-
 a = float(input("Input a: "))
-s = input('Input "+", "-", "*", "/", "^", or "sqrt": ' )
+s = input('Input "+", "-", "*", "/", "^", "sqrt", or "abs": ' )

 if s == '-':
     b = float(input("Input b: "))
@@ -23,5 +21,7 @@ elif s == '^':
     print(a**b)
 elif s == 'sqrt':
     print(math.sqrt(a))
:...skipping...
commit 032a41d7c06167f50f805115aba205c8ed9be896
Author: Zloybanan4ik <denik55557777809@gmail.com>
Date:   Fri Dec 27 02:56:34 2024 +0300

    Модуль

diff --git a/main.py b/main.py
index a0c0def..4f8d976 100644
--- a/main.py
+++ b/main.py
@@ -1,7 +1,5 @@
-import math
-
 a = float(input("Input a: "))
-s = input('Input "+", "-", "*", "/", "^", or "sqrt": ' )
+s = input('Input "+", "-", "*", "/", "^", "sqrt", or "abs": ' )

```

Рисунок 10. Предпоследний коммит

11. Просмотр коммита через указанный хэш:

```

user@DESKTOP-VI4I94J MINGW64 ~/workspace/lab2 (main)
$ git show ed84e9a79c4004f2bdc13125180f0a2c14d1df6c
commit ed84e9a79c4004f2bdc13125180f0a2c14d1df6c (tag: v2.0)
Author: Zloybanan4ik <denik55557777809@gmail.com>
Date:   Fri Dec 27 02:48:42 2024 +0300

    Добавлено возведение в степень

diff --git a/main.py b/main.py
index 3176126..289567b 100644
--- a/main.py
+++ b/main.py
@@ -1,6 +1,6 @@
 a = float(input("Input a: "))
 b = float(input("Input b: "))
-s = input('Input "+", "-", "*", or "/": ')
+s = input('Input "+", "-", "*", "/", or "^": ')

 if s == '-':
     print(a-b)
@@ -13,5 +13,7 @@ elif s == '/':
     print(a/b)
 else:
     print("Division by zero is not allowed!")
+elif s == '^':
+...skipping...
commit ed84e9a79c4004f2bdc13125180f0a2c14d1df6c (tag: v2.0)
Author: Zloybanan4ik <denik55557777809@gmail.com>
Date:   Fri Dec 27 02:48:42 2024 +0300

    Добавлено возведение в степень

diff --git a/main.py b/main.py
index 3176126..289567b 100644
--- a/main.py
+++ b/main.py
@@ -1,6 +1,6 @@
 a = float(input("Input a: "))
 b = float(input("Input b: "))
-s = input('Input "+", "-", "*", or "/": ')
+s = input('Input "+", "-", "*", "/", or "^": ')

```

Рисунок 11. Коммит с указанным хэшем

12. Освоил возможность отката к заданной версии. Удалил весь код из одного из файлов программы репозитория и сохранил этот файл.
13. Удалил все несохраненные изменения в файле командой `git checkout`:

```

PS C:\Users\user\workspace\lab2> git checkout -- main.py
PS C:\Users\user\workspace\lab2>

```

Рисунок 12. Использование команды `git checkout`

14. Откатил состояние хранилища к предыдущей версии:

```

PS C:\Users\user\workspace\lab2> git add *
PS C:\Users\user\workspace\lab2> git status
On branch main

Changes to be committed:
  modified:   main.py

PS C:\Users\user\workspace\lab2> git commit -m "Удаление кода"
[main df090a9] Удаление кода
 1 file changed, 34 deletions(-)
PS C:\Users\user\workspace\lab2> git reset --hard HEAD~1
HEAD is now at d12c827 Логарифм
PS C:\Users\user\workspace\lab2>

```

Рисунок 13. Откат изменений

Вывод: в результате использования команды `git reset --hard` отменяется последние коммиты и сбрасывается рабочая копия до определенного состояния.

Ответы на контрольные вопросы:

1. Как выполнить историю коммитов в Git? Какие существуют дополнительные опции для просмотра истории коммитов?

Для просмотра истории коммитов в Git используется команда `git log`. Она отображает список коммитов в хронологическом порядке, начиная с самого последнего.

При помощи аргумента `-p` можно посмотреть разницу внесенную в каждый коммит.

При помощи параметра `-n`, где `n` это число, можно посмотреть историю последних `n`-коммитов.

2. Как ограничить вывод при просмотре истории коммитов?

Для ограничения вывода при просмотре истории коммитов в Git можно использовать несколько опций:

- Ограничение количества коммитов; `-(n)`;
- Фильтрация по времени;

С помощью `-since`, `-after` показывает те коммиты, которые сделаны после указанной даты

С помощью `-until`, `-before` показывает только те коммиты, которые были сделаны до указанной даты.

- Фильтрация по автору;
`--author`
- Фильтрация по сообщению коммита;
`--committer`
- Ограничение вывода по формату.
`--pretty=format'''`

3. Как внести изменения в уже сделанный коммит?

- `git commit --amend`;
- `git reset <имя_коммита>`.

4. Как отменить индексацию файла в Git?

`git checkout`. Эта команда отменит все изменения, внесенные в файлы, вернув их к состоянию в последнем коммите.

`git reset HEAD`. Эта команда удалит файл из индекса, но оставит его в рабочей области.

5. Как отменить изменения в файле?

В Git есть несколько способов отменить изменения в файле:

- `git checkout -- <имя файла>`;
- `git reset HEAD <имя файла>`;
- `git restore <имя файла>`;
- `git revert <имя файла>`;
- `git stash`

6. Что такое удаленный репозиторий Git?

Удаленный репозиторий Git – это хранилище кода, доступное с разных компьютеров через сеть.

7. Как выполнить просмотр удаленных репозиториях данного локального репозитория?

Для просмотра удаленных репозиториях, связанных с вашим локальным репозиторием Git, используйте команду `git remote`. Эта команда выведет

список имен, удаленных репозиториях, которые вы добавили к своему локальному репозиторию.

8. Как добавить удаленный репозиторий для данного локального репозитория?

Чтобы добавить удаленный репозиторий к вашему локальному репозиторию Git, используйте команду `git remote add`: `git remote add <имя_удаленного_репозитория> <URL_удаленного_репозитория>`

9. Как выполнить отправку/получение изменений с удаленного репозитория?

Отправка изменений (push): `git push <имя_удаленного_репозитория><ветвь>`: Эта команда отправляет ваши локальные изменения в указанную ветвь на удаленный репозиторий.

Получение изменений (pull): `git pull <имя_удаленного_репозитория><ветвь>`: Эта команда получает изменения из указанной ветви на удаленном репозитории и объединяет их с вашей локальной веткой

10. Как выполнить просмотр удаленного репозитория?

- С помощью команды `git log`;
- С помощью команды `git fetch`;
- С помощью команды `git remote show`.

11. Каково назначение тэгов Git?

Тэги Git – это метки, которые позволяют пометить определенные коммиты в репозитории. Они служат для идентификации важных моментов в истории проекта, таких как релизы, версии или отправные точки для ветвления.

12. Как осуществляется работа с тэгами Git?

- Создание тега;
- Просмотр тегов;
- Перемещение тега;
- Удаление тега;

- Отправка тегов на удаленный репозиторий;
- Просмотр информации о теге;
- Переключение на коммит, помеченный тегом.

13. Самостоятельно изучите назначение флага `--prune` в командах `git fetch` и `git push`. Каково назначение этого флага?

Флаг `--prune` в командах `git fetch` и `git push` используется для удаления удаленных веток, которые уже не существуют на удаленном сервере.

Вывод: в ходе работы исследовал базовые возможности системы контроля версий Git для работы с локальными репозиториями.

Ссылка на GitHub: <https://github.com/Zloybanan4ik/lab2.git>