

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ**

**Отчет о лабораторной работе №1.1 по дисциплине основы программной
инженерии**

Выполнил:
Шальнев Владимир Сергеевич,
2 курс, группа ПИЖ-б-о-20-1,

Проверил:
Доцент кафедры
прикладной математики и
компьютерной безопасности,
Воронкин Р.А.

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2021 г.

ВЫПОЛНЕНИЕ:

```
C:\Users\V0V41K-ПК>git version  
git version 2.24.1.windows.2
```

Рисунок 1.1 – Проверка установленной версии git

Процесс клонирования репозитория представлен на рисунке 1.2.

```
C:\Users\V0V41K-ПК>F:  
F:\>cd F:\pythonProject\laba python  
F:\pythonProject\laba python>git clone https://github.com/HAXF13D/laboratory-1  
Cloning into 'laboratory-1'...  
remote: Enumerating objects: 4, done.  
remote: Counting objects: 100% (4/4), done.  
remote: Compressing objects: 100% (3/3), done.  
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0  
Unpacking objects: 100% (4/4), done.  
F:\pythonProject\laba python>cd laboratory-1  
F:\pythonProject\laba python\laboratory-1>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)  
    .idea/  
    main.py  
  
nothing added to commit but untracked files present (use "git add" to track)  
F:\pythonProject\laba python\laboratory-1>
```

Рисунок 1.2 – Клонирование репозитория

Код первой программы, добавление и коммит представлены на рисунках 1.3 и 1.4 соответственно.

```
1 a = int(input("a = "))  
2 b = int(input("b = "))  
3 print(f"Сумма a + b = {a + b}")  
4
```

Рисунок 1.3 – Первый вариант кода программы

```

F:\pythonProject\laba python\laboratory-1>git add main.py

F:\pythonProject\laba python\laboratory-1>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:   main.py

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

F:\pythonProject\laba python\laboratory-1>git commit -m "First version of program"
[main 1f91445] First version of program
 1 file changed, 3 insertions(+)
 create mode 100644 main.py

F:\pythonProject\laba python\laboratory-1>git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

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

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

```

Рисунок 1.4 – Первый коммит

Программа после внесения изменений представлена на рисунке 1.5.

```

1      a = int(input("a = "))
2      b = int(input("b = "))
3      print(f"Разность a - b = {a - b}")
4

```

Рисунок 1.5 – Второй вариант кода программы

Выполнение команд git add и git commit во второй раз представлено на рисунке 1.6.

```

F:\pythonProject\laba python\laboratory-1>git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   main.py

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

no changes added to commit (use "git add" and/or "git commit -a")

F:\pythonProject\laba python\laboratory-1>git add main.py

F:\pythonProject\laba python\laboratory-1>git commit -m "Program with difference"
[main 1d8cef4] Program with difference
 1 file changed, 1 insertion(+), 1 deletion(-)

F:\pythonProject\laba python\laboratory-1>git status
On branch main
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)

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

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

F:\pythonProject\laba python\laboratory-1>

```

Рисунок 1.6 – Второй коммит

Третий вариант программы и третий коммит представлены на рисунках 1.7 и 1.8.

```

1  a = int(input("a = "))
2  b = int(input("b = "))
3  print(f"a * b = {a * b}")
4  |

```

Рисунок 1.7 – Третий вариант кода программы

```

F:\pythonProject\laba python\laboratory-1>git add main.py

F:\pythonProject\laba python\laboratory-1>git commit -m "Program with composition"
[main b0665ca] Program with composition
 1 file changed, 1 insertion(+), 1 deletion(-)

F:\pythonProject\laba python\laboratory-1>git status
On branch main
Your branch is ahead of 'origin/main' by 3 commits.
  (use "git push" to publish your local commits)

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

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

```

Рисунок 1.8 – Третий коммит

Четвертый вариант программы и четвертый коммит представлены на рисунках 1.9 и 1.10.

```

1      a = int(input("a = "))
2      b = int(input("b = "))
3      if b == 0:
4          print("Error you can't divide by 0")
5      else:
6          print(f"a // b = {a // b}")

```

Рисунок 1.9 – Четвертый вариант кода программы

```

F:\pythonProject\laba python\laboratory-1>git add main.py

F:\pythonProject\laba python\laboratory-1>git commit -m "Program with division"
[main 371f7a1] Program with division
 1 file changed, 4 insertions(+), 1 deletion(-)

F:\pythonProject\laba python\laboratory-1>git status
On branch main
Your branch is ahead of 'origin/main' by 4 commits.
  (use "git push" to publish your local commits)

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

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

```

Рисунок 1.10 – Четвертый коммит

Пятый вариант программы и пятый коммит представлены на рисунках 1.11 и 1.12.

```

1   a = int(input("a = "))
2   b = int(input("b = "))
3   print(f"Максимум из двух чисел = {max(a, b)}")
4

```

Рисунок 1.11 – Пятый вариант кода программы

```

F:\pythonProject\laba python\laboratory-1>git add main.py
F:\pythonProject\laba python\laboratory-1>git commit -m "Maximum"
[main dc65852] Maximum
1 file changed, 1 insertion(+), 4 deletions(-)

```

Рисунок 1.12 – Пятый коммит

Шестой вариант программы и шестой коммит представлены на рисунках 1.13 и 1.14.

```

1   a = int(input("a = "))
2   b = int(input("b = "))
3   print(f"Минимум из двух чисел = {min(a, b)}")
4

```

Рисунок 1.13 – Шестой вариант кода программы

```

F:\pythonProject\laba python\laboratory-1>git add main.py
F:\pythonProject\laba python\laboratory-1>git commit -m "Minimum"
[main 3713fe0] Minimum
1 file changed, 1 insertion(+), 1 deletion(-)

```

Рисунок 1.14 – Шестой коммит

Седьмой вариант программы и седьмой коммит представлены на рисунках 1.15 и 1.16.

```

1   a = int(input("Введите двузначное число: "))
2   print(f"Первая цифра числа = {a // 10}")
3   print(f"Вторая цифра числа = {a % 10}")
4

```

Рисунок 1.15 – Седьмой вариант кода программы

```

F:\pythonProject\laba python\laboratory-1>git add main.py
F:\pythonProject\laba python\laboratory-1>git commit -m "First and second digit of number"
[main 3fad2b9] First and second digit of number
1 file changed, 3 insertions(+), 3 deletions(-)

```

Рисунок 1.16 – Седьмой коммит

Восьмой вариант программы и Восьмой коммит представлены на рисунках 1.17 и 1.18.

```
1 a = int(input("Введите число: "))
2 count = 0
3 while a > 0:
4     a = a // 10
5     count += 1
6 print(f"Кол-во цифр в числе = {count}")
7
```

Рисунок 1.17 – Восьмой вариант кода программы

```
F:\pythonProject\laba python\laboratory-1>git add main.py
F:\pythonProject\laba python\laboratory-1>git commit -m "Quantity of digits in number"
[main bb915f8] Quantity of digits in number
1 file changed, 6 insertions(+), 3 deletions(-)
```

Рисунок 1.18 – Восьмой коммит

Распространение изменений в исходный репозиторий на GitHub и их проверка представлены на рисунках 1.19 и 1.20.

```
F:\pythonProject\laba python\laboratory-1>git push --set-upstream origin main
Logon failed, use ctrl+c to cancel basic credential prompt.
Username for 'https://github.com': HAXF13D
Password for 'https://HAXF13D@github.com':
Enumerating objects: 25, done.
Counting objects: 100% (25/25), done.
Delta compression using up to 6 threads
Compressing objects: 100% (24/24), done.
Writing objects: 100% (24/24), 2.36 KiB | 1.18 MiB/s, done.
Total 24 (delta 8), reused 0 (delta 0)
remote: Resolving deltas: 100% (8/8), done.
To https://github.com/HAXF13D/laboratory-1
 d6acaa5..bb915f8  main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

Рисунок 1.19 – Распространение изменений в исходный репозиторий на GitHub

Commits on Sep 8, 2021		
Quantity of digits in number HAXF13D committed 21 minutes ago	bb915f8	<>
First and second digit of number HAXF13D committed 23 minutes ago	3fad2b9	<>
Minimum HAXF13D committed 25 minutes ago	3713fe0	<>
Maximum HAXF13D committed 26 minutes ago	dc65852	<>
Program with division HAXF13D committed 28 minutes ago	371f7a1	<>
Program with composition HAXF13D committed 32 minutes ago	b0665ca	<>
Program with difference HAXF13D committed 35 minutes ago	1d8cef4	<>
First version of program HAXF13D committed 38 minutes ago	1f91445	<>
Initial commit HAXF13D committed 1 hour ago	Verified d6acaa5	<>

1.20 – Проверка изменений

Ответы на вопросы:

1. Система контроля версий (СКВ) — это система, регистрирующая изменения в одном или нескольких файлах с тем, чтобы в дальнейшем была возможность вернуться к определённым старым версиям этих файлов.
2. Недостаток локальной — можно легко забыть, в какой директории вы находитесь, и случайно изменить не тот файл или скопировать не те файлы, которые вы хотели. Недостаток центральной — это единая точка отказа, представленная централизованным сервером. Если этот сервер выйдет из строя на час, то в течение этого времени никто не сможет использовать контроль версий для сохранения изменений, над которыми работает, а также никто не сможет обмениваться этими изменениями с другими разработчиками.
3. Git относится к распределённой системе контроля версий.
4. Основное отличие — особый подход к работе с данными. Хранения файлов в Git похоже на набор снимков файловой системы.
5. Целостность обеспечивается с помощью хеширования(sha-1).
6. Зафиксированное (committed), изменённое (modified) и подготовленное (staged).
7. Профиль - это ваша публичная страница на GitHub, как и в социальных сетях.
8. Два типа — публичный и приватный.
9. На компьютере содержится локальный репозиторий, в который вносятся изменения. Локальный репозиторий не требует интернета. После внесения всех необходимых изменений их можно отправить на удаленный репозиторий.
10. Для проверки нажать win+г ввести cmd, в открывшемся окне ввести git version. После этого ввести следующие команды:
git config --global user.name <YOUR_NAME>
git config --global user.email <EMAIL>
11. Этапы создания: 1) Нажать на кнопку «New»
2) Ввести название, описание по желанию.
3) Выбрать тип — публичный/приватный.
4) Добавить README, gitignore, выбрать тип лицензии.
5) Нажать на кнопку создания.
12. Apache License 2.0, GNU General Public License v3.0, MIT License, BSD 2-Clause "Simplified" License, BSD 3-Clause "New" or "Revised" License, Boost Software License 1.0, Creative Commons Zero v1.0 Universal, Eclipse Public License 2.0, GNU Affero General Public License v3.0, GNU General Public License v2.0, GNU Lesser General Public License v2.1, Mozilla Public License 2.0, The Unlicense.
13. Командой git clone, это нужно для копирования репозитория на локальный компьютер.

14. Вводом команды `git status`.
15. После добавления изменений в файл, при вводе команды `git status` данный файл будет подсвечен красным цветом, говоря о том, что изменения в нем не отображены в локальном репозитории. После выполнения команды `git add` название файла будет подсвечено зеленым, а текст в консоли будет говорить о том, что есть изменения нуждающиеся в комите. После выполнения команды `git commit` будет отображено, что текущая ветвь находится на определенное количество комитов дальше исходной. После отправки изменений будет отображено, что “конец” удаленной ветки находится там, же где и результат последних коммитов.
16. Первым делом мы должны на обоих компьютерах прописать `git clone`. После этого мы можем начинать работу. Если мы внесли изменения на 1 компьютере, то мы должны их отправить на сервер командой `git push`, а для того, что бы эти изменения были на другом компьютере мы должны прописать `git pull`, для того, что переместить файлы из удаленного репозитория в наш локальный. Последовательность действий – такая: 1) Сел за ПК.
2) Сделал `git pull`.
3) Поработал над проектом.
4) Сделал `git commit`, `git push`
17. GitLab, Bitbucket, Beanstalk. К примеру пользователь GitLab может создать сколько угодно частных репозиториев. Основное различие между GitHub и GitLab заключается в философии, которую представляет каждая платформа. GitHub имеет более высокую доступность и в большей степени ориентирован на производительность инфраструктуры, в то время как GitLab в большей степени ориентирован на предложение системы, основанной на функциях, с централизованной, интегрированной платформой для веб-разработчиков.
18. К примеру, в среду разработки PyCharm встроен VCS. Мы можем включить интеграцию СКВ, найти репозиторий, создать его, а также поделиться работой на гитхаб.

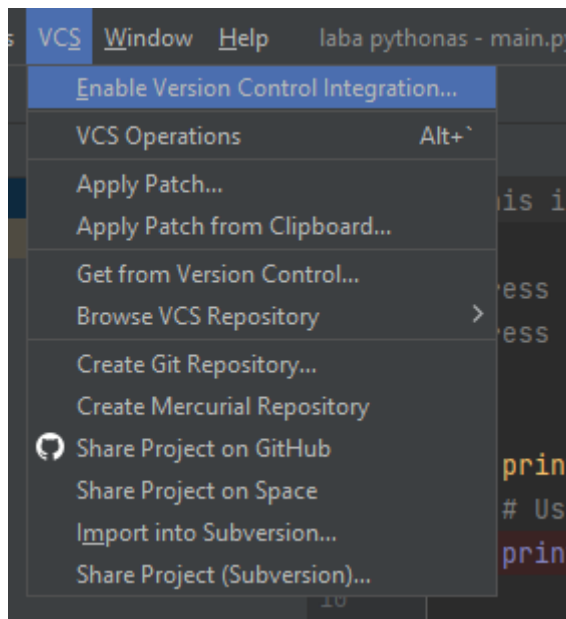


Рисунок 1.21 – Окно VCS в PyCharm.



Используя клавишу  можно выполнить commit, а используя клавишу  можно выполнить push.



Рисунок 1.22 – Основные клавиши работы с Git в Pycharm

С помощью встроенной системы контроля версий можно наблюдать различные изменения, как представлено на рисунке 1.23

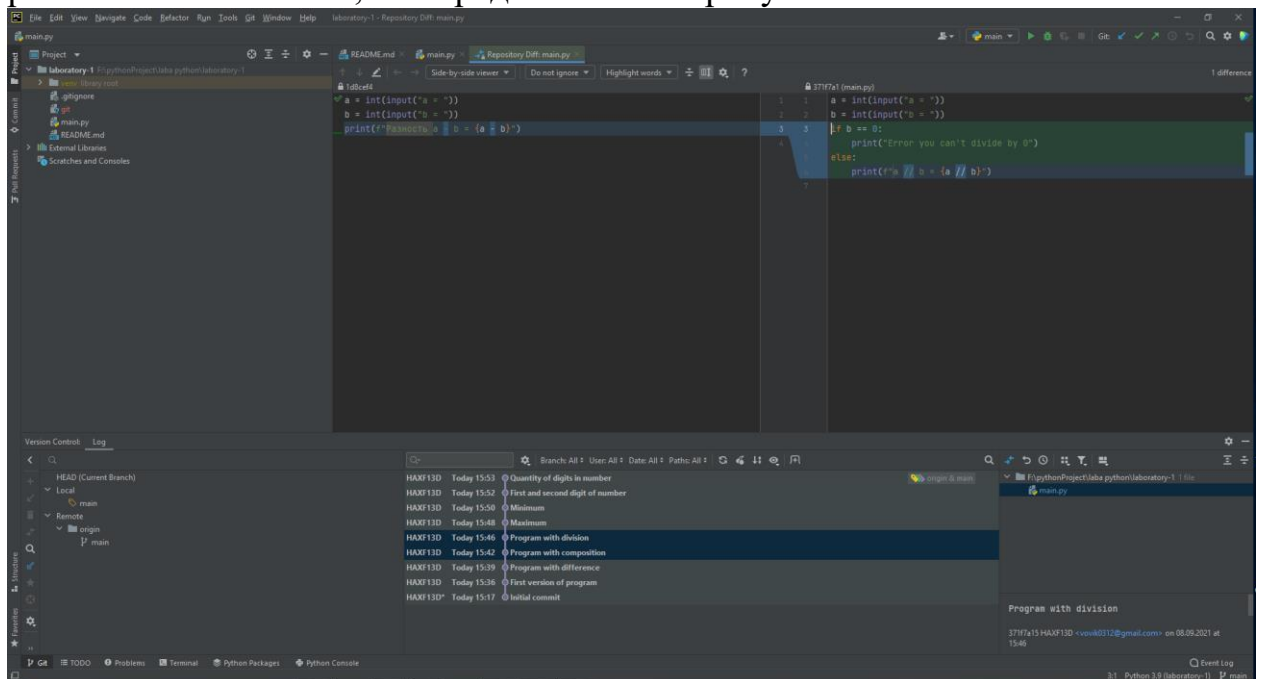


Рисунок 1.23 – Контроль версий с помощью VCS встроенной в PyCharm