**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:



Рисунок 1.1 – Проверка установленной версии git

Процесс клонирования репозитория представлен на рисунке 1.2.

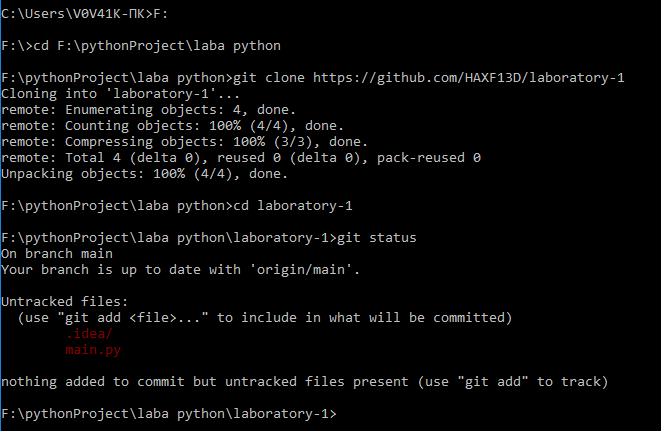


Рисунок 1.2 – Клонирование репозитория

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

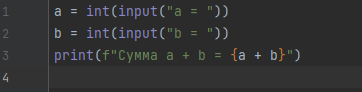


Рисунок 1.3 – Первый вариант кода программы

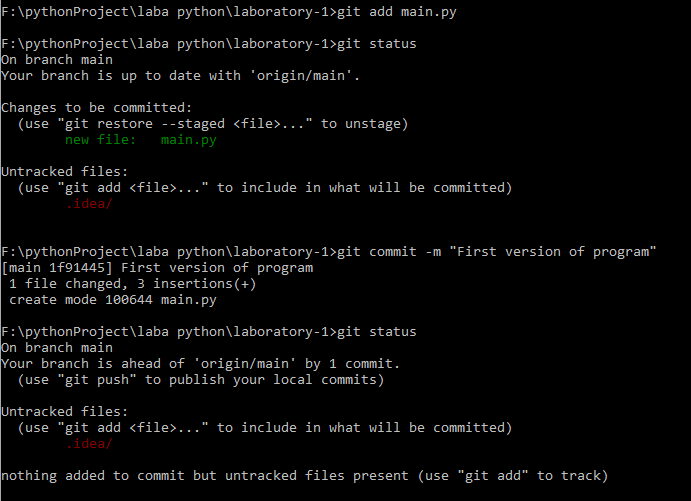


Рисунок 1.4 – Первый коммит

Программа после внесения изменений представлена на рисунке 1.5.

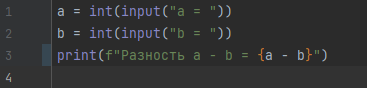


Рисунок 1.5 – Второй вариант кода программы

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

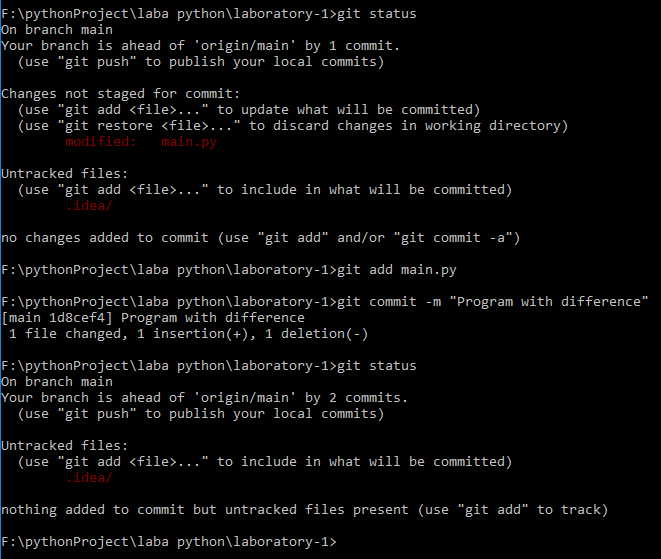


Рисунок 1.6 – Второй коммит

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

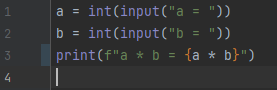


Рисунок 1.7 – Третий вариант кода программы

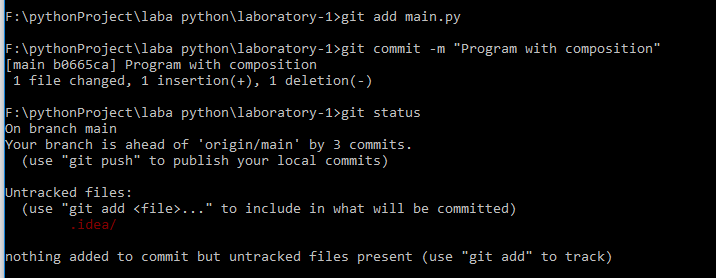


Рисунок 1.8 – Третий коммит

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

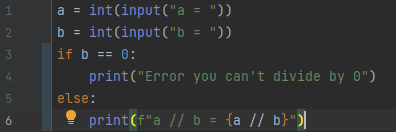


Рисунок 1.9 – Четвертый вариант кода программы

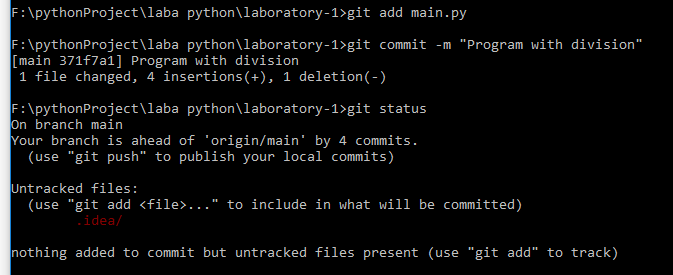


Рисунок 1.10 – Четвертый коммит

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

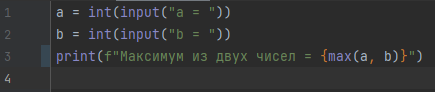


Рисунок 1.11 – Пятый вариант кода программы

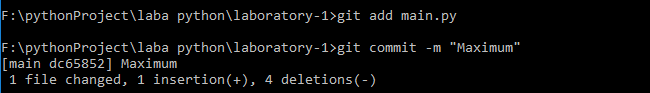


Рисунок 1.12 – Пятый коммит

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

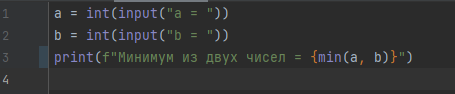


Рисунок 1.13 – Шестой вариант кода программы

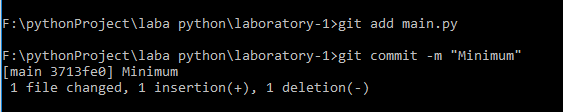


Рисунок 1.14 – Шестой коммит

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

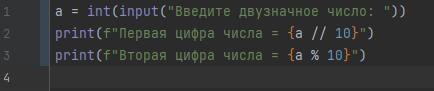


Рисунок 1.15 – Седьмой вариант кода программы

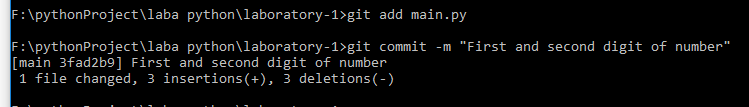


Рисунок 1.16 – Седьмой коммит

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

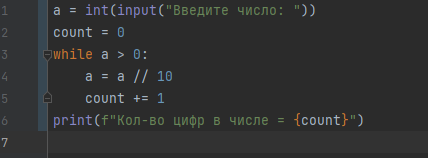


Рисунок 1.17 – Восьмой вариант кода программы

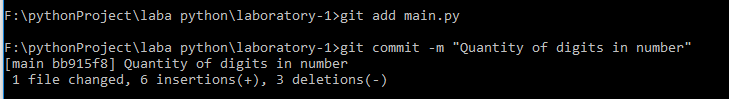


Рисунок 1.18 – Восьмой коммит

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

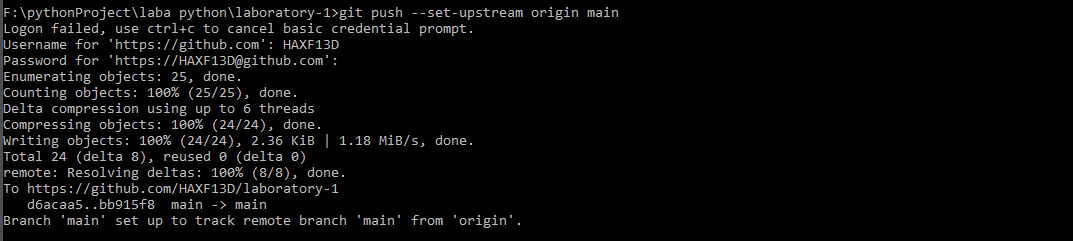


Рисунок 1.19 – Распространение изменений в исходный репозиторий на GitHub



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 внисения всех необходимых изменений их можно отправить на удаленный репозиторий.
10. Для проверки нажать win+r ввести cmd, в открывшемся окней ввести git version. После этого ввести следующие команды:

git config --global user.name <YOUR\_NAME>

git config --global user.email <EMAIL>

1. 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.

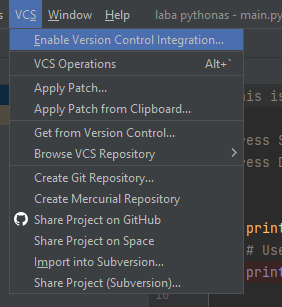
1. Apache License 2.0, GNU General Public License v3.0, MIT License, BSD 2-Clause "Simplified" License, BSD 3-Clause "New" or "Revised" License, Boost Software License 1.0, Creative Commons Zero v1.0 Universal, Eclipse Public License 2.0, GNU Affero General Public License v3.0, GNU General Public License v2.0, GNU Lesser General Public License v2.1, Mozilla Public License 2.0, The Unlicense.
2. Командой git clone, это нужно для копирования репозитория на локальный компьютер.
3. Вводом команды git status.
4. 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 измененяи нуждающиееся в комите. После выполнения команды git commit будет отображено, что текущая ветвь находится на опредленное количество комитов дальше исходной. После отправки изменений будет отображено, что “конец” удаленной ветки находится там, же где и результат последних коммитов.
5. 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

1. GitLab, Bitbucket, Beanstalk. К примеру пользователь GitLab может создать сколько угодно приватных 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.
2. 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

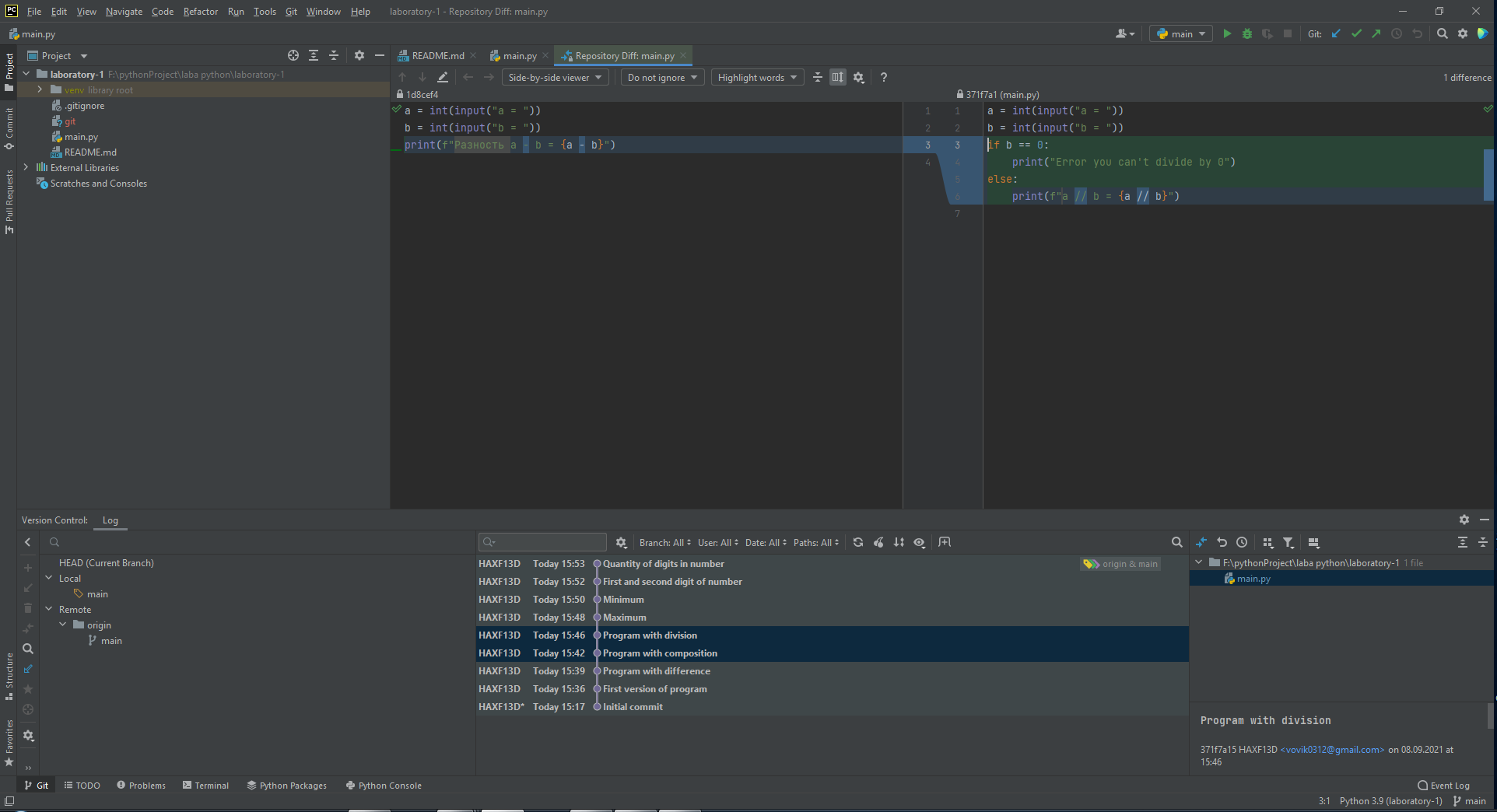


Рисунок 1.23 – Контроль версий с помощью VCS встроенной в PyCharm