# МИНИCTEPCTBO НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ

**ФЕДЕРАЦИИ**

# Федеральное государственное автономное образовательное учреждение высшего образования

**«СЕВЕРОКАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

# Кафедра инфокоммуникаций Институт цифрового развития

**ОТЧЁТ**

# по лабораторной работе №2.1

Дисциплина: «Основы кроссплатформенного программирования» Тема: «Основы языка Python»

|  |
| --- |
| Выполнил: студент 1 курса |
| группы ИВТ-б-о-21-1 |
| Богдашов Артём Владимирович |

Ставрополь 2022

Выполнение работы:

1. Создал репозиторий GitHub с лицензией MIT, добавил .gitignore python, клонировал репозиторий на ПК и организовал репозиторий согласно модели ветвления git-flow:

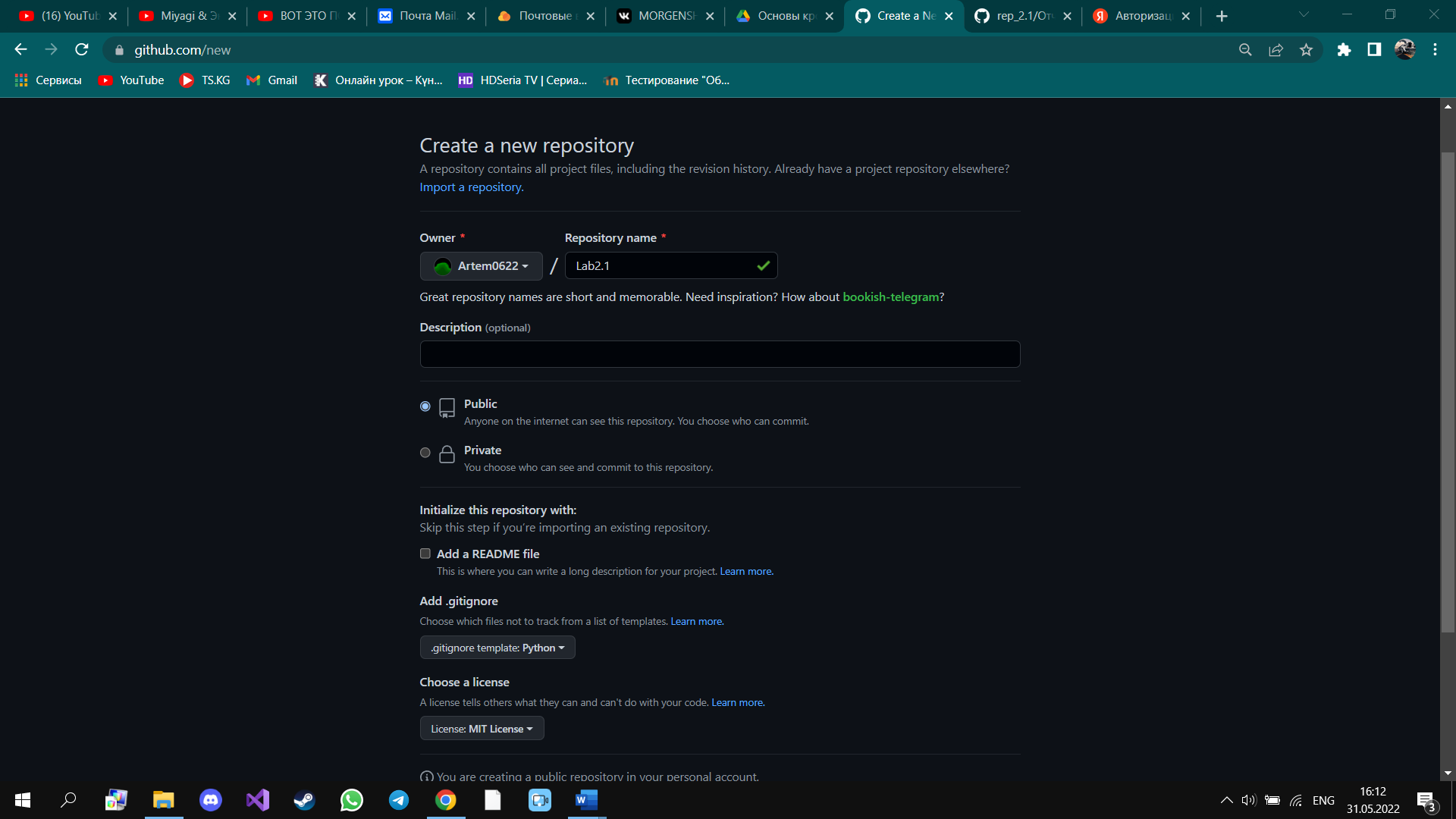


Рисунок 1.1 Создание репозитория

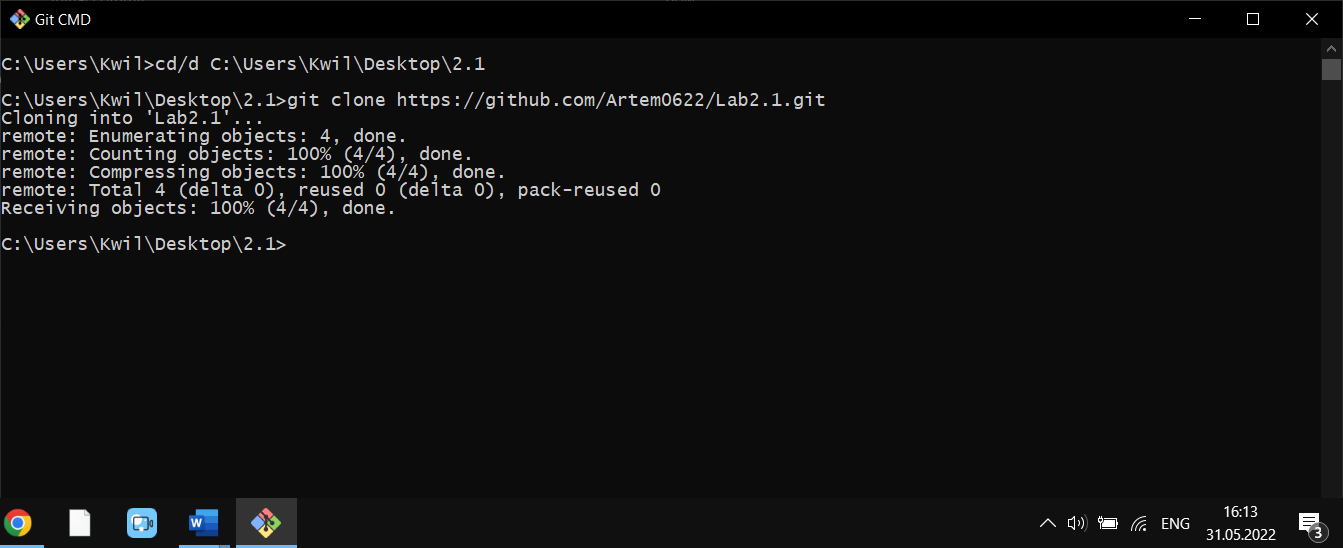


Рисунок 1.2 Клонирование репозитория

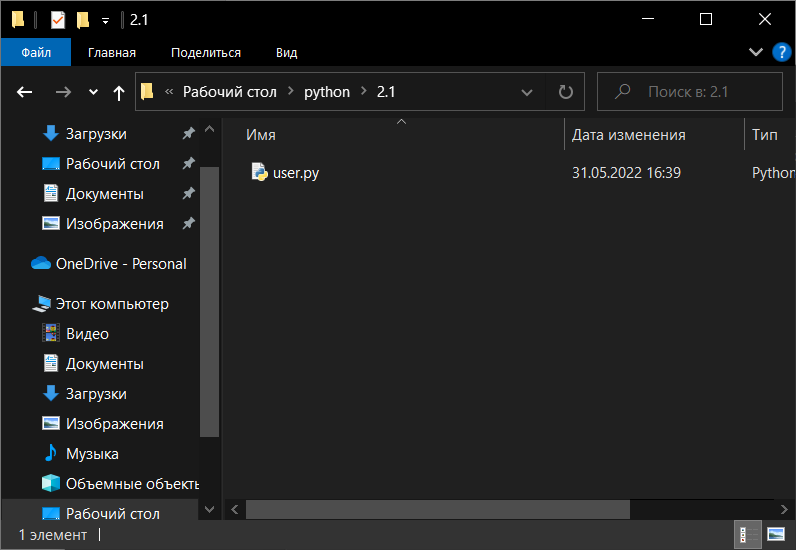


Рисунок 1.3 Организация репозитория согласно модели ветвления git-flow

1. Написал программу user.py, которая запрашивала бы у пользователя имя, возраст и место жительства, после этого выводила бы 3 строки

"This is `имя`" "It is `возраст`"

"(S)he live in `место\_жительства`"



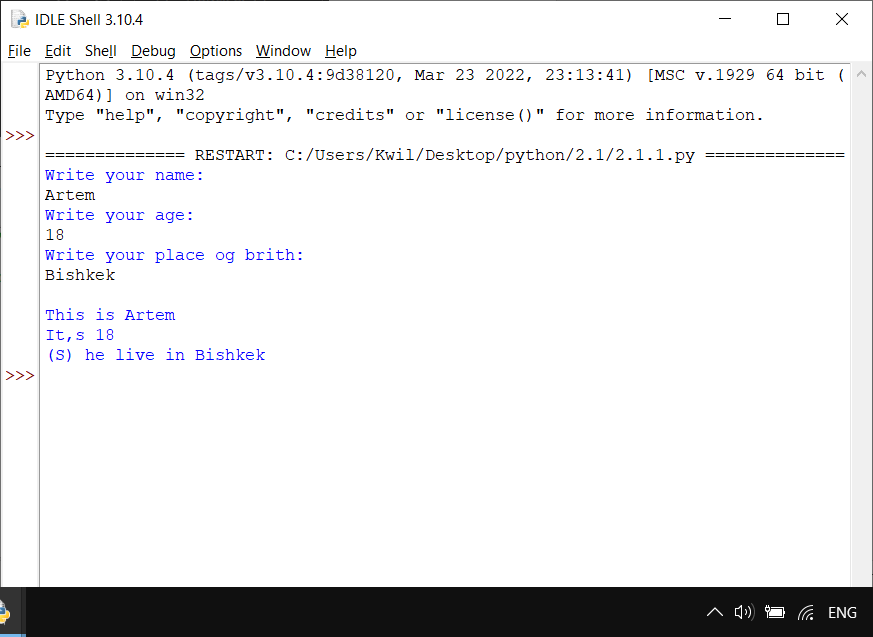
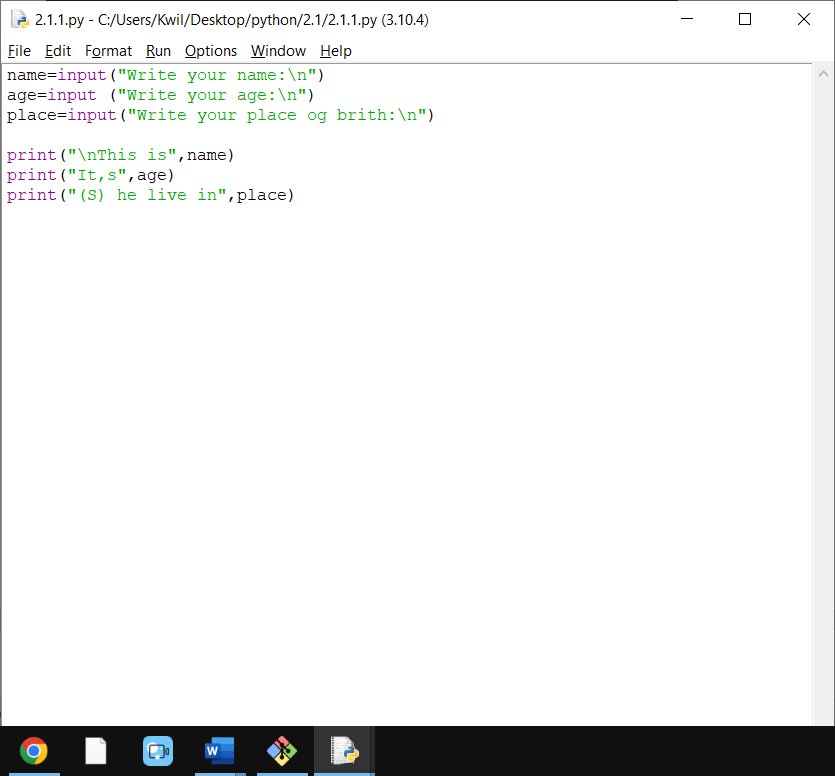


Рисунок 1.1 Программа user

1. . Написал программу (файл arithmetic.py), которая предлагала бы пользователю решить пример 4 \* 100 - 54. Потом выводила бы на экран правильный ответ и ответ пользователя.

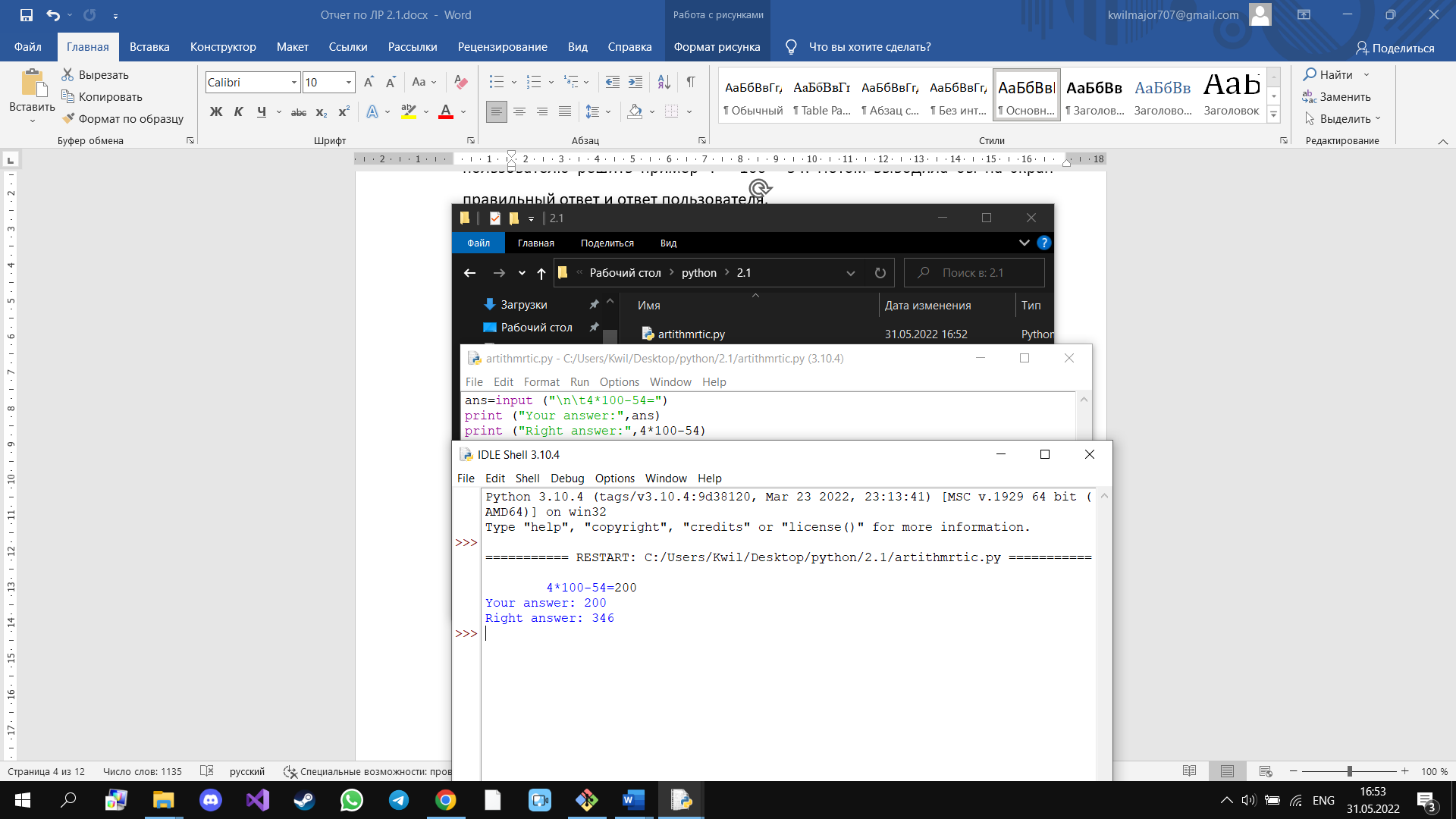


Рисунок 4.1 Программа arithmetic.py

1. Написал программу numbers.py, которая запрашивает у пользователя 4 числа, отдельно складывает первые два и вторые два, затем делит первую сумму на вторую, после выводит рез-т на экран с точностью до сотен.

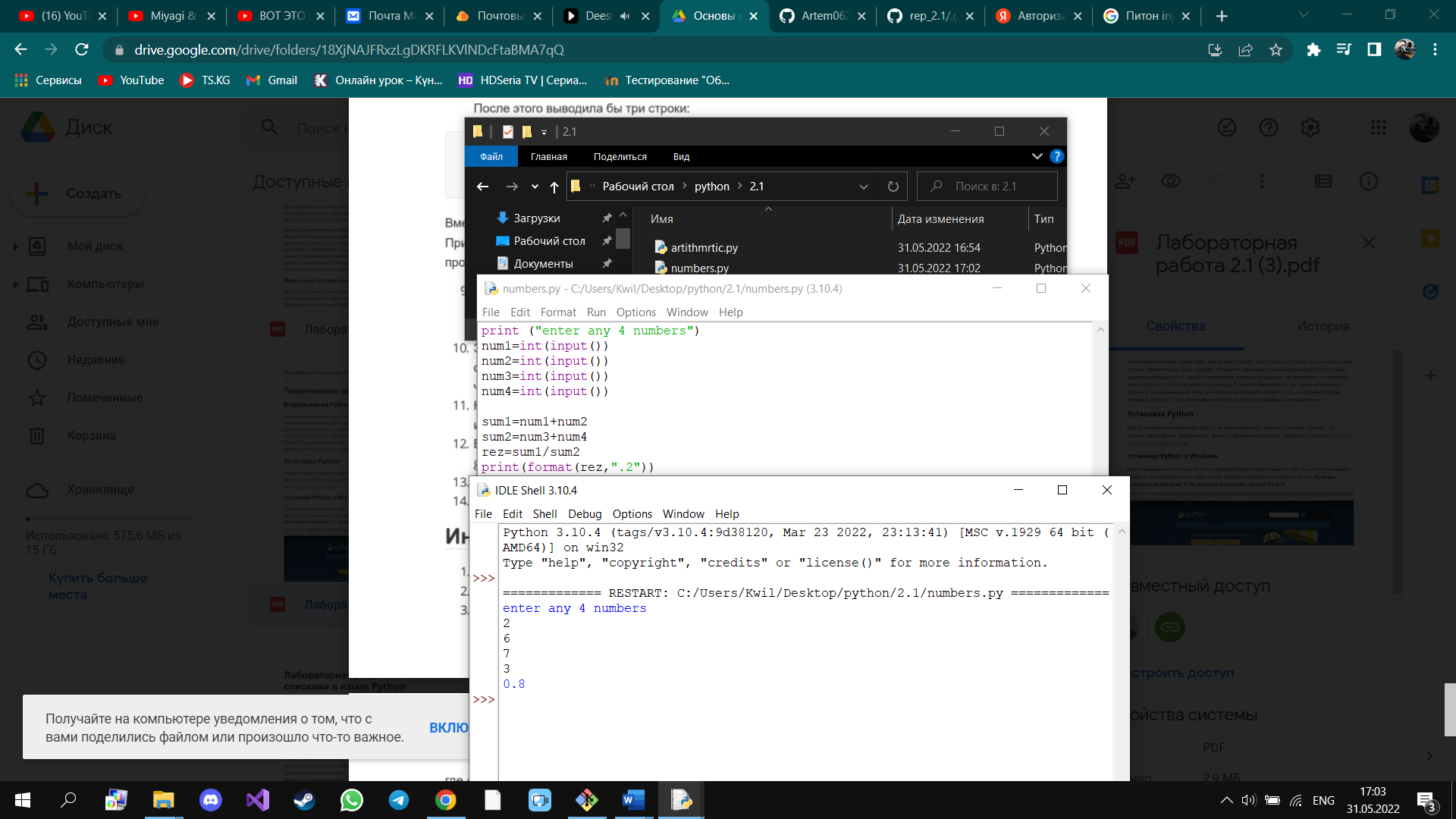


Рисунок 5.1 Программа numbers.py

1. Написал программу для индивидуального задания:

Даны координаты на плоскости двух точек. Найти расстояние между этими точками.

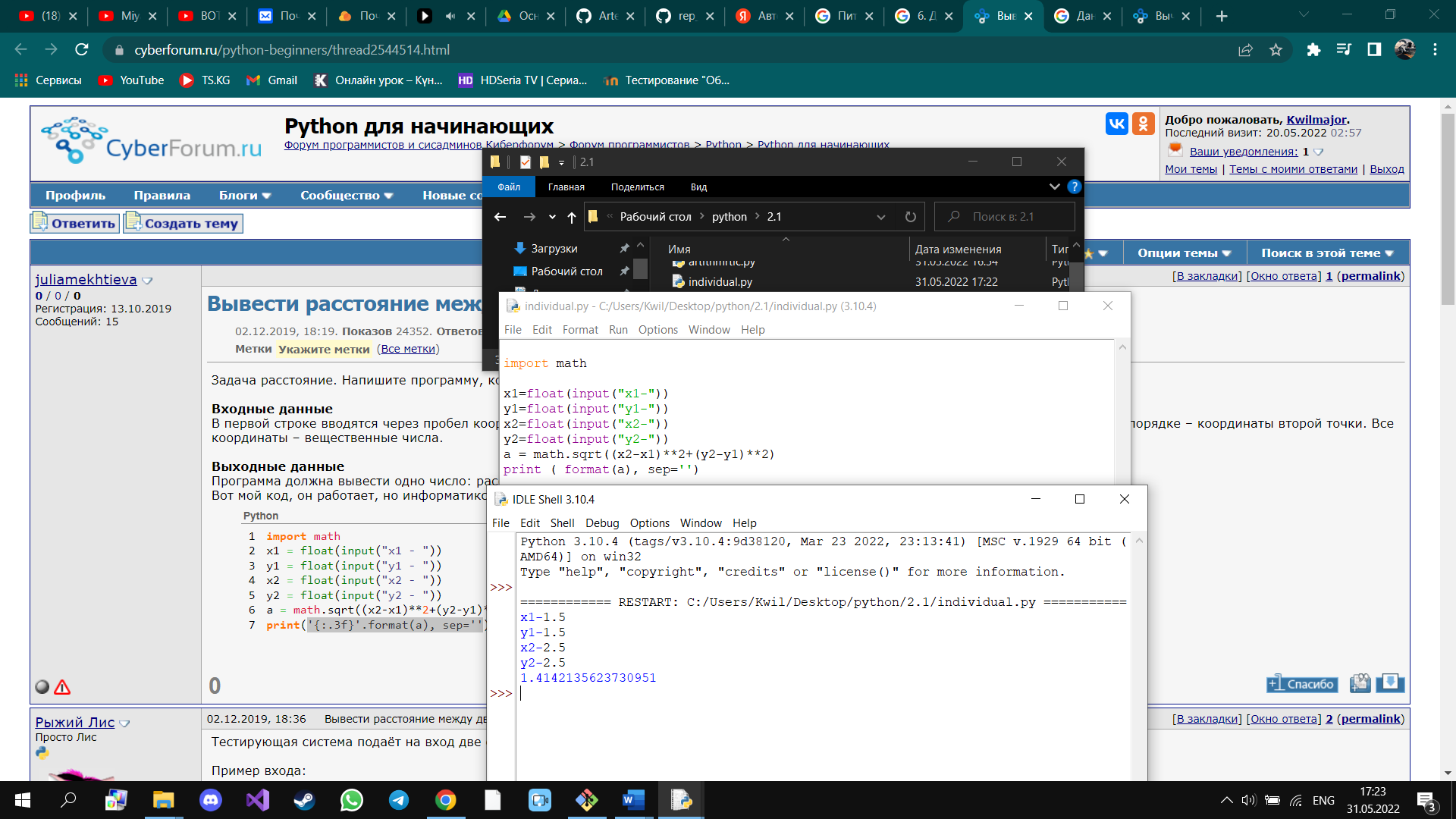


Рисунок 5.1 Программа индивидуального задания

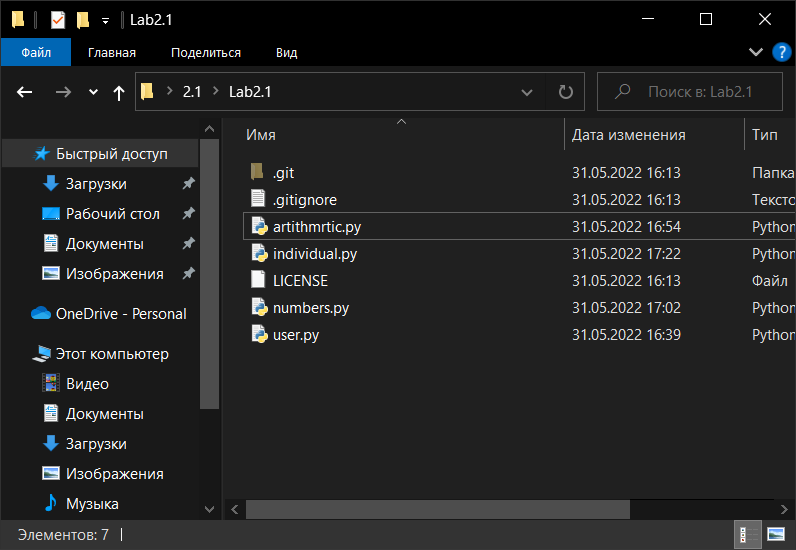


Рисунок5.2 Программы в репозитории

1. Сделал коммит изменений в ветку разработки, выполнил ее слияние с веткой main и отправил сделанные изменения на уд. репозиторий.

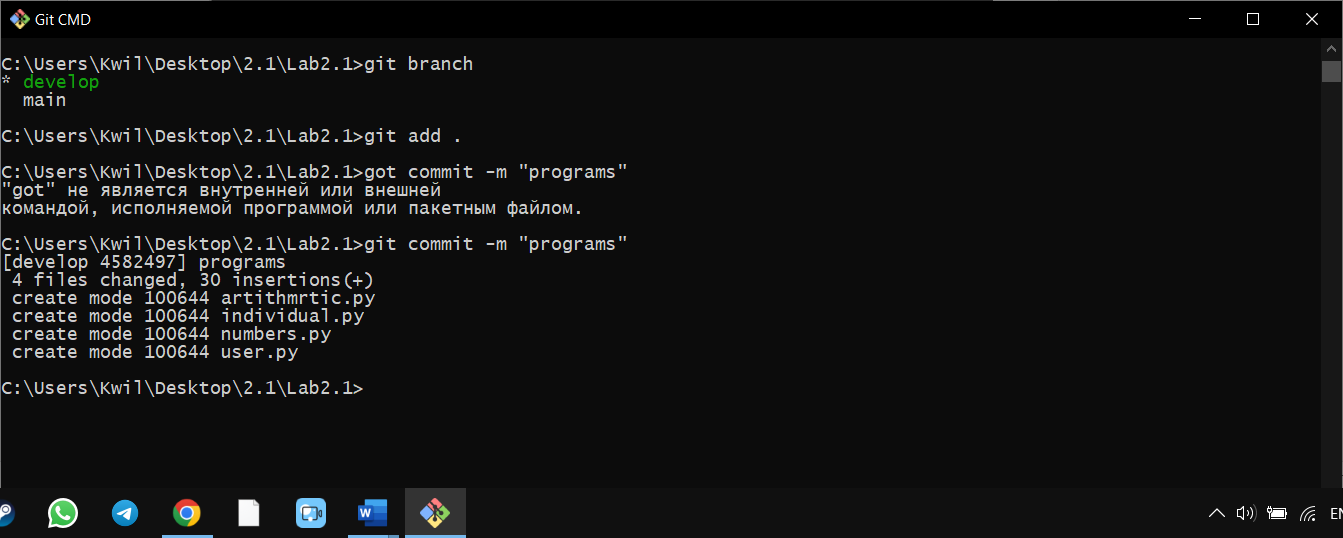


Рисунок 6.1 Коммит изменений в ветку develop

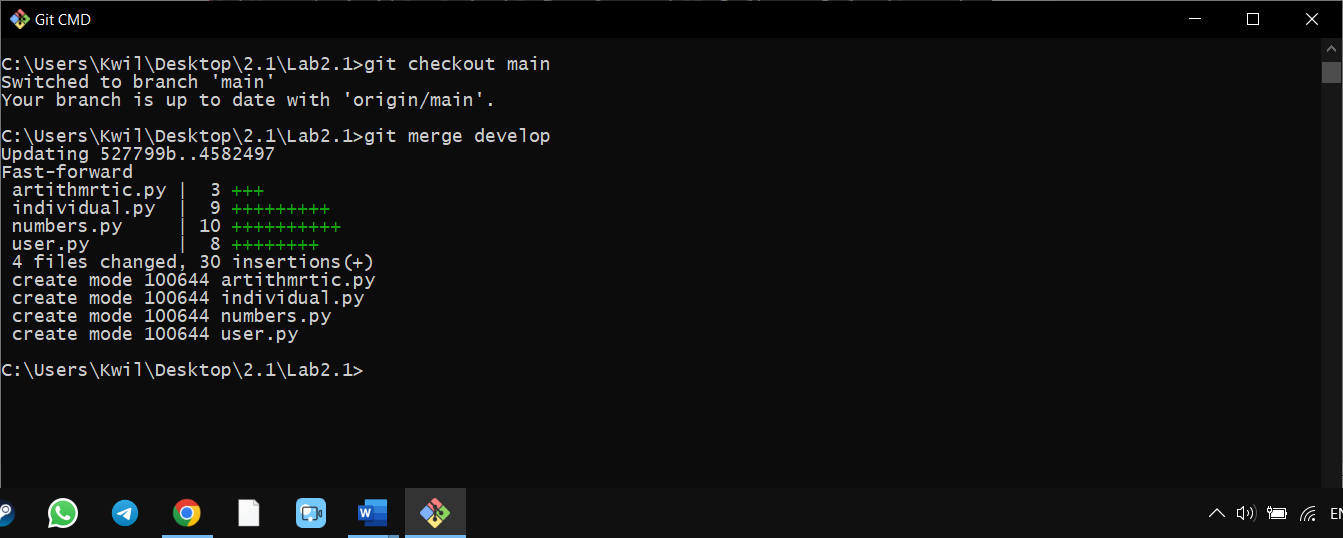


Рисунок 6.2 Слияние ветки develop с веткой main

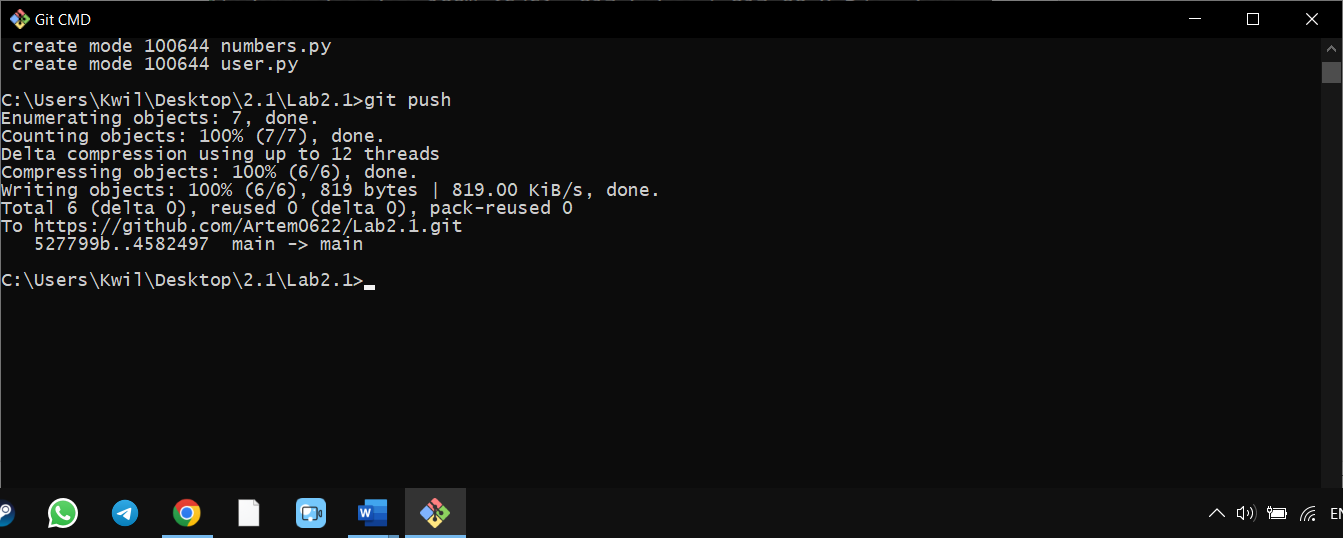


Рисунок 6.3 push коммитов на уд. репозиторий

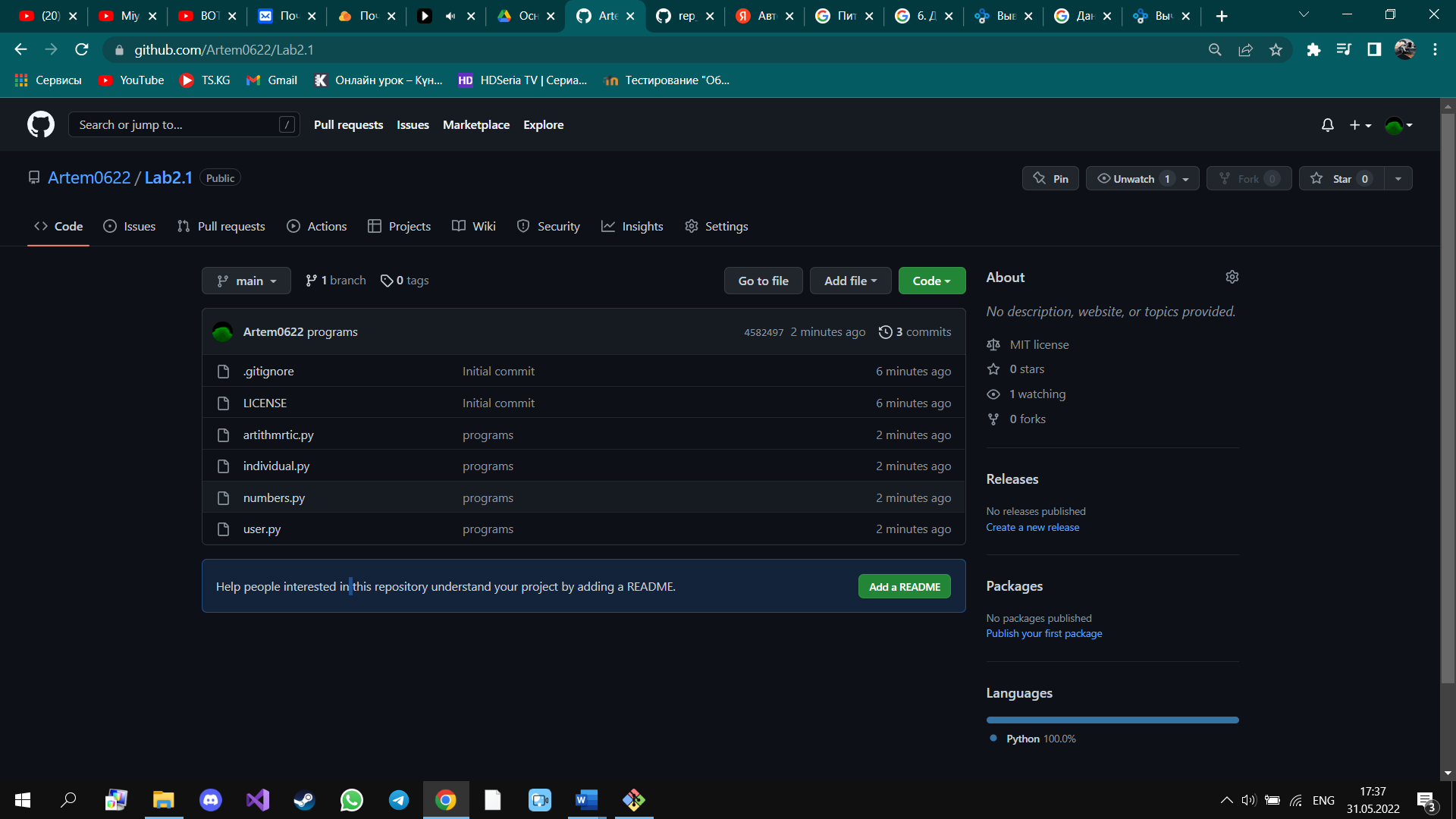


Рисунок 6.4 Изменения на уд. сервере

# Опишите основные этапы установки Python в Windows и Linux.

Linux: Чаще всего интерпретатор Python уже в входит в состав дистрибутива.

Windows: Осн. этапы установки Python на Windows:

1. Скачать дистрибутив с официального сайта;
2. Запустить скачанный установочный файл;
3. Выбрать способ установки;
4. Отметить необходимые опции установки;
5. Выбрать место устновки;
6. Готово.

# В чем отличие пакета Anaconda от пакета Python, скачиваемого с официального сайта?

Пакет Anaconda содержит версии языка Python 2 и 3, набор наиболее часто используемых библиотек и удобную среду разработки и исполнения, запускаемую в браузере, а также на Anaconda удобнее запускать примеры.

# Как осуществить проверку работоспособности пакета Anaconda?

Для выполнения проверки работоспособности Anaconda необходимо вначале запустить командный процессор с поддержкой виртуальных окружений Anaconda. В появившейся командной строке необходимо ввести > jupyter notebook, в результате чего отобразиться процесс загрузки веб-среды Jupyter Notebook, после чего запустится веб-сервер и среда разработки в браузере. Создать ноутбук для разработки, для этого нажать на кнопку New и в появившемся списке выбрать Python. В результате будет создана новая страница в браузере с ноутбуком. Ввести в первой ячейке команду print(“Hello, World!”) и нажать Alt+Enter на компьютере. Ниже ячейки должна появиться соответствующая надпись.

# Как задать используемый интерпретатор языка Python в IDE PyCharm?

Указать путь до интерпретатора в настройках IDE, для этого:

1. Нажмите на шестеренку в верхнем правом углу, выберите "Add..".
2. Далее выберите "System Interpreter";
3. Нажмите на 3 точки "..." справа от поля в выбором интерпретатора;
4. Укажите путь до интерпретатора.

# Как осуществить запуск программы с помощью IDE PyCharm?

Сочетанием клавиш Shift+F10.

# В чем суть интерактивного и пакетного режимов работы Python?

Интерактивный.

Python можно использовать как калькулятор для различных вычислений, а если дополнительно подключить необходимые математические библиотеки, то по своим возможностям он становится практически равным таким пакетам как Matlab, Octave и т.п.

Проектный.

В этом режиме сначала записывается вся программа, а потом эта программа выполняется полностью.

# Почему язык программирования Python называется языком динамической типизации?

Т. к. в ЯП Python проверка типа происходит во время выполнения, а не компиляции.

# Какие существуют основные типы в языке программирования Python?

Типы в ЯП Python:

1. None
2. Логические переменные
3. Числа
4. Списки
5. Строки
6. Бинарные списки
7. Множества
8. Словари

# Как создаются объекты в памяти? Каково их устройство? В чем заключается процесс объявления новых переменных и работа операции присваивания?

Для того, чтобы объявить и сразу инициализировать переменную необходимо написать её имя, потом поставить знак равенства и значение, с которым эта переменная будет создана.

При инициализации переменной, на уровне интерпретатора, создается целочисленный объект, который имеет некоторый идентификатор, значение и тип. Посредством оператора “=” создается ссылка между переменной и объектом.

# Как получить список ключевых слов в Python?

Список ключевых слов можно получить непосредственно в программе, для этого нужно подключить модуль keyword и воспользоваться командой keyword.kwlist.

# Каково назначение функций id() и type()?

Функция id() предназначена для получения значения идентичности объекта.

С помощью функции type() можно получить тип конкретного объекта.

# Что такое изменяемые и неизменяемые типы в Python.

К неизменяемым (immutable) типам относятся: целые числа (int), числа с плавающей точкойn(float), комплексные числа (complex), логические переменные (bool), кортежи (tuple), строки (str) и неизменяемые множества (frozen set).

К изменяемым (mutable) типам относятся: списки (list), множества (set), словари (dict).

# Чем отличаются операции деления и целочисленного деления?

При целочисленном делении отбрасывается дробная часть от деления чисел, при операции деления дробная часть не отбрасывается.

# Какие имеются средства в языке Python для работы с комплексными числами?

Для создания комплексного числа можно использовать функцию complex(a, b), в которую, в

качестве первого аргумента, передается действительная часть, в качестве второго – мнимая.

Либо записать число в виде a + bj. Комплексные числа можно складывать, вычитать, умножать, делить и возводить в степень. У комплексного числа можно извлечь действительную(x.real) и мнимую части(x.imag).

Для получения комплексносопряженного число необходимо использовать метод conjugate().

# Каково назначение и основные функции библиотеки (модуля) math? По аналогии с модулем math изучите самостоятельно назначение и основные функции модуля cmath.

Для выполнения математических операций необходим модуль math. Осн. операции библиотеки math:

math.ceil(x) - возвращает ближайшее целое число большее, чем x. math.fabs(x) - возвращает абсолютное значение числа. math.factorial(x) - вычисляет факториал x.

math.floor(x) - возвращает ближайшее целое число меньшее, чем x. math.exp(x) - вычисляет e\*\*x.

math.log2(x) - логарифм по основанию 2. math.log10(x) - логарифм по основанию 10.

math.log(x[, base]) - по умолчанию вычисляет логарифм по основанию e, дополнительно можно указать основание логарифма.

math.pow(x, y) - вычисляет значение x в степени y. math.sqrt(x) - корень квадратный от x.

math.cos(x) - косинус от x. math.sin(x) - синус от x. math.tan(x) - тангенс от x. math.acos(x) - арккосинус от x. math.asin(x) - арксинус от x. math.atan(x) - арктангенс от x. math.pi - число пи.

math.e - число е.

# Каково назначение именных параметров sep и end в функции print()?

Через параметр sep можно указать отличный от пробела разделитель строк.

Параметр end позволяет указывать, что делать, после вывода строки.

# Каково назначение метода format()? Какие еще существуют средства для форматирования строк в Python? Примечание: в дополнение к рассмотренным средствам изучите самостоятельно работу с f-строками в Python.

Форматирование может выполняться в так называемом старом стиле или с помощью строкового метода format.

Cимволы %s , %d , %f подставляются значения переменных. Буквы s, d, f обозначают типы данных – строку, целое число, вещественное число.

# Каким образом осуществить ввод с консоли значения целочисленной и вещественной переменной в языке Python?

Указать перед input тип данных: int(input()).