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 ПРАКТИЧЕСКОЙ РАБОТЕ №2**

**дисциплины «Программирование на Python»**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Выполнила:  Ковжого Елизавета Андреевна  2 курс, группа ИВТ-б-о-24-1,  09.03.01 «Информатика и вычислительная техника», направленность (профиль) «Программное обеспечение средств вычислительной техники и автоматизированных систем», очная форма обучения  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | | Руководитель практики:  Воронкин Р. А., доцент департамента цифровых, робототехнических систем и электроники института перспективной инженерии.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  |  | |  |

Отчет защищен с оценкой \_\_\_\_\_\_\_\_\_Дата защиты\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ставрополь, 2025 г.

Тема: оcновы языка Python.

Цель: исследовать процесс установки и базовые возможности языка Python версии 3.х.

Порядок выполнения работы:

Адрес созданного репозитория: https://github.com/LissKovzogo/Python\_LAB\_2.git

1. Последовательно установили на устройство Python, Anaconda и PyCharm, после чего проверили работоспособность.

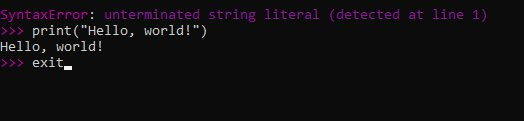


Рис.1 — Проверка работоспособности Python

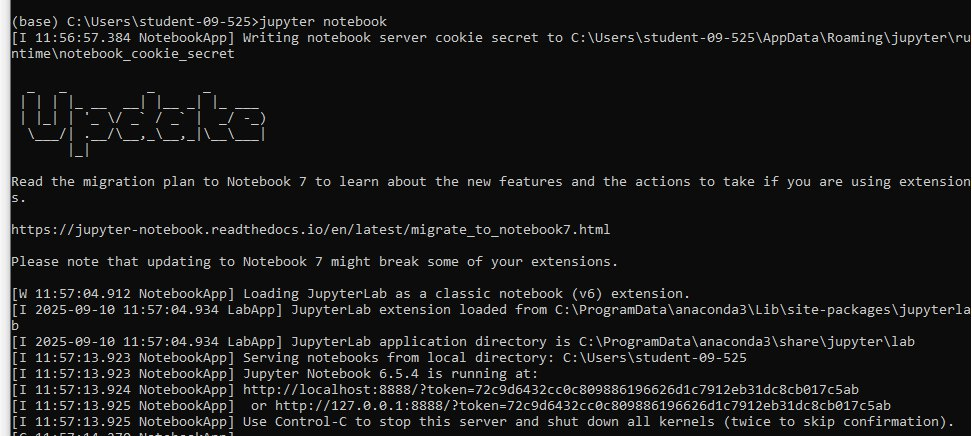


Рис. 2 — Проверка работоспособности Anaconda

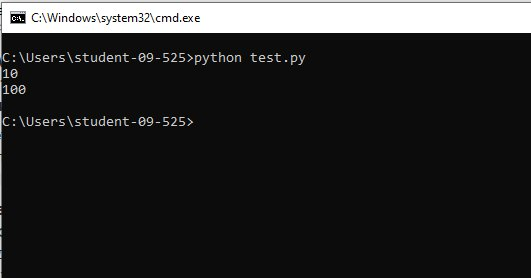


Рис. 3 —Проверка работоспособности PyCharm

2. Создали новый репозиторий и клонировали их.

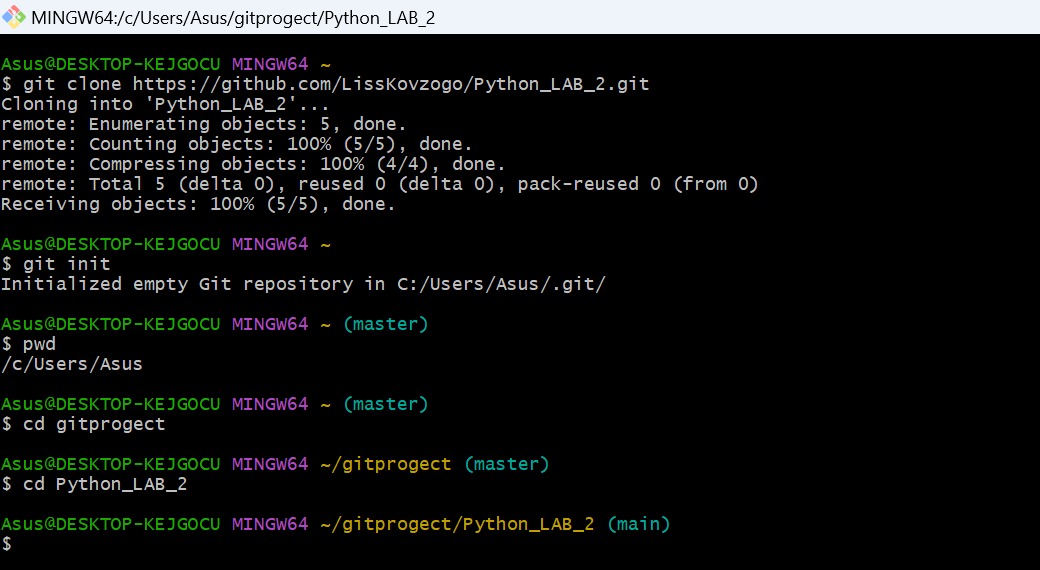


Рис. 4 — Клонирование репозитория

3. В соответствии с заданиями написали код и сохранили в файлы user.py, arithmetic.py,numbers.py,individul.py,individual2.py.

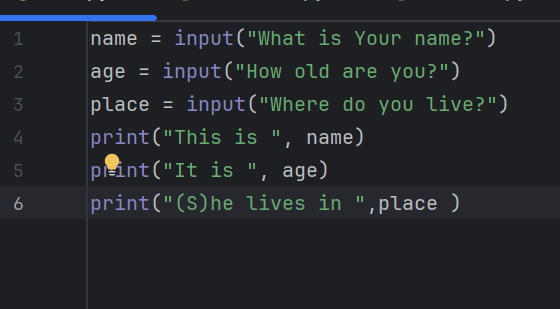


Рис. 5 — user.py

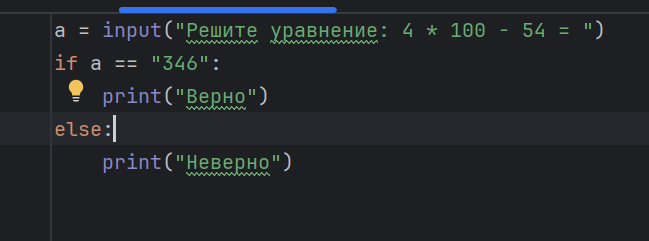


Рис. 6 — arithmetic.py

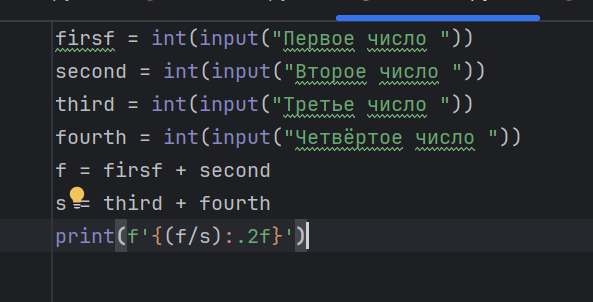


Рис. 7 — numbers.py

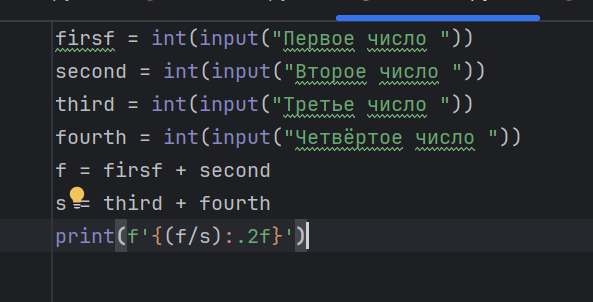


Рис. 8 — individul.py

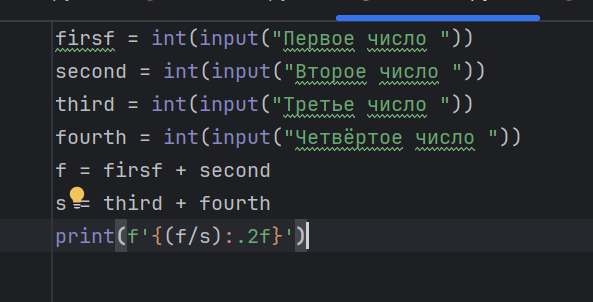


Рис. 9 — individual2.py

4. Произвели коммит всех созданных файлов и выполнили слияние веток.

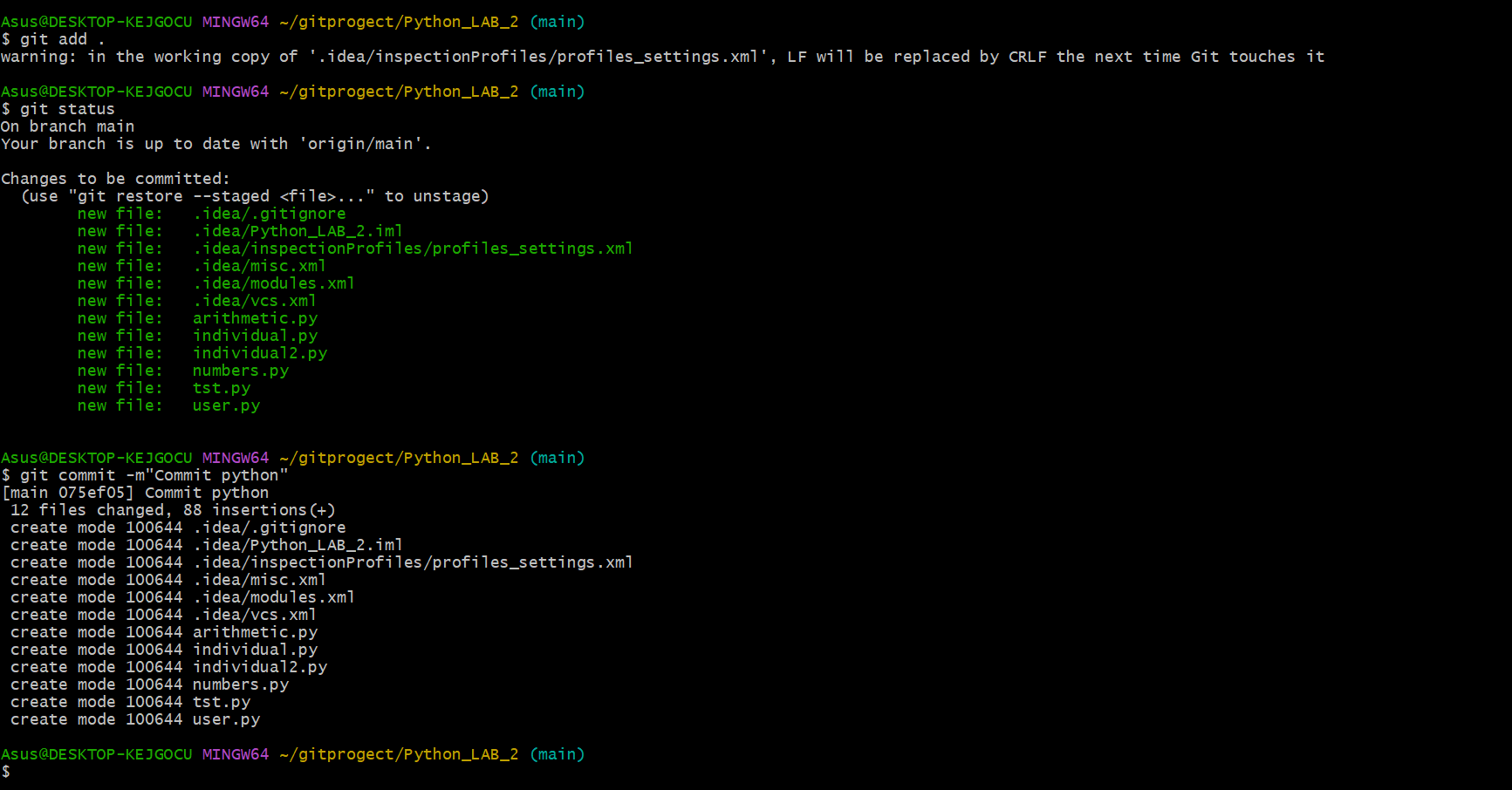


Рис. 10 — Сохранение обновлений репозитория

5. Отправили изменения в удалённый репозиторий.

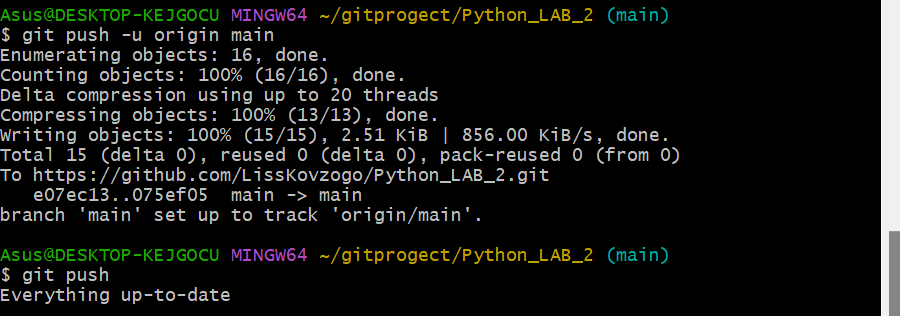


Рис. 11 — Отправление на сервер GitHub

Контрольные вопросы:

1. Основные этапы установки Python

Windows:

* Скачать установщик с python.org
* Запустить установщик, отметить "Add Python to PATH"
* Выбрать опцию "Install Now" или "Customize installation"
* Завершить установку
* Проверить в командной строке: python --version

Linux:

* Для Ubuntu/Debian: sudo apt update && sudo apt install python3 python3-pip
* Для CentOS/RHEL: sudo yum install python3
* Для Fedora: sudo dnf install python3
* Проверить: python3 --version

2. Отличие Anaconda от официального Python

Anaconda:

* Включает Python + 1500+ популярных научных пакетов
* Собственный менеджер пакетов conda
* Упрощенное управление виртуальными окружениями
* Готовое окружение для data science

Официальный Python:

* Только интерпретатор Python
* Менеджер пакетов pip
* Требует ручной установки дополнительных библиотек

3. Проверка работоспособности Anaconda

bash

conda --version

conda list

python --version

jupyter notebook

4. Настройка интерпретатора в PyCharm

File → Settings → Project → Python Interpreter

Settings → Add → Выбрать нужный интерпретатор

Можно выбрать: System Python, Virtualenv, Conda, Docker

5. Запуск программы в PyCharm

Правой кнопкой по файлу → Run 'filename'

6. Режимы работы Python

Интерактивный режим:

Команды выполняются сразу после ввода

Запуск: python или python3

Подходит для тестирования и экспериментов

Пакетный режим:

Выполнение программы из файла

Запуск: python script.py

Для полноценных приложений

7. Динамическая типизация

Python определяет типы переменных во время выполнения, а не компиляции:

python

x = 5 *# int*

x = "hello" *# str - без ошибки*

8. Основные типы данных

Числовые: int, float, complex, bool

Последовательности: str, list, tuple, range

Множества: set, frozenset

Словари: dict

Бинарные: bytes, bytearray, memoryview

NoneType: None

9. Создание объектов и переменных

Создание объекта:

python

x = 5 *# Создается объект int со значением 5*

10. Список ключевых слов

python

import keyword

print(keyword.kwlist)

11. Функции id() и type()

id() - возвращает уникальный идентификатор объекта

type() - возвращает тип объекта

12. Изменяемые vs неизменяемые типы

Неизменяемые (immutable):

int, float, str, tuple, frozenset

Нельзя изменить после создания

Изменяемые (mutable):

list, dict, set, bytearray

Можно изменять содержимое

13. Операции деления

/ - обычное деление (возвращает float)

// - целочисленное деление (возвращает int)

% - остаток от деления

14. Комплексные числа

python

z = 3 + 4j

print(z.real) *# 3.0*

print(z.imag) *# 4.0*

print(z.conjugate()) *# (3-4j)*

15. Модули math и cmath

math - математические функции для вещественных чисел:

python

import math

math.sqrt(16) *# 4.0*

math.sin(math.pi/2) *# 1.0*

cmath - для комплексных чисел:

python

import cmath

cmath.sqrt(-1) *# 1j*

16. Параметры sep и end в print()

sep - разделитель между аргументами (по умолчанию пробел)

end - что печатать в конце (по умолчанию '\n')

python

print(1, 2, 3, sep='-', end='!') *# 1-2-3!*

17. Форматирование строк

Метод format():

python

"{} {}".format("Hello", "World")

"{1} {0}".format("World", "Hello")

f-строки (Python 3.6+):

python

name = "Alice"

age = 25

f"Name: {name}, Age: {age}"

Другие способы:

%-форматирование: "Name: %s" % name

Конкатенация: "Hello " + name

## 18. Ввод числовых переменных

python

# Целое число

x = int(input("Введите целое число: "))

# Вещественное число

y = float(input("Введите вещественное число: "))

# С обработкой ошибок

try:

num = int(input("Введите число: "))

except ValueError:

print("Это не число!")