

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 №4
дисциплины
«Основы кроссплатформенного программирования»
Вариант 12

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.

Лабораторная работа 1.3 «Основы языка Python»

Цель работы: исследование процесса установки и базовых возможностей языка Python 3.x.

Ссылка на GitHub <https://github.com/Zloybanan4ik/lab4.git>

Ход работы:

1. Создан общедоступный репозиторий на GitHub

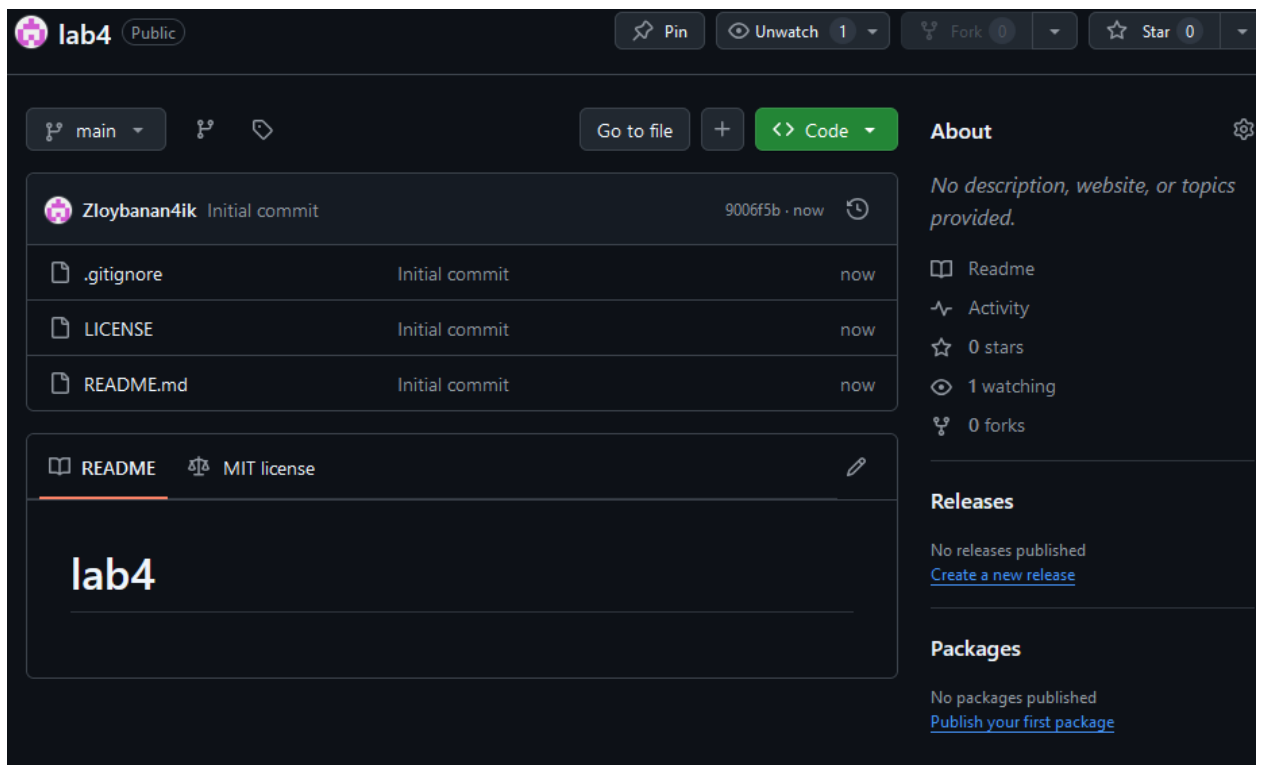


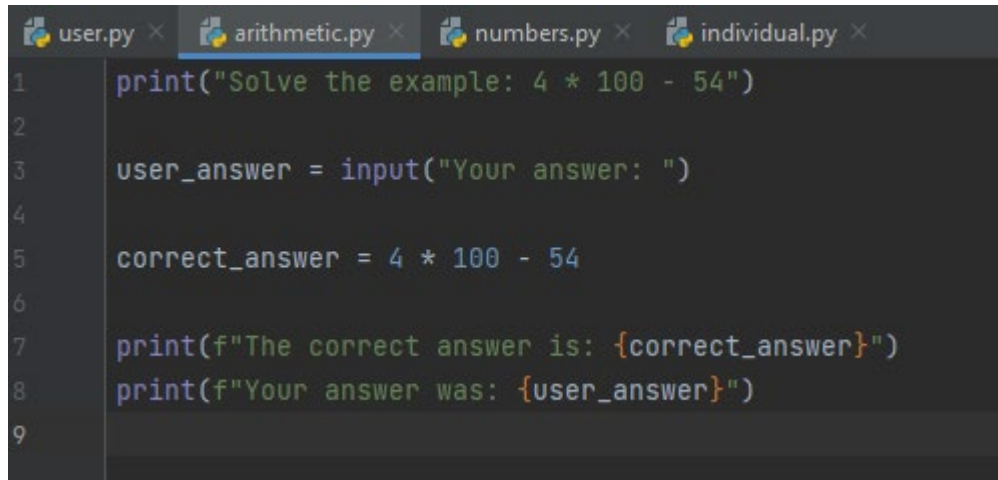
Рисунок 1. Общедоступный репозиторий

2. Выполнил задание «user.py»:

```
user.py x arithmetic.py x numbers.py x individual.py x
1 name = input("What is your name? ")
2 age = input("How old are you? ")
3 location = input("Where do you live? ")
4
5 print(f"This is {name}")
6 print(f"It is {age}")
7 print(f"(S)he live in {location}")
8
```

Рисунок 2. Программа задания «user»

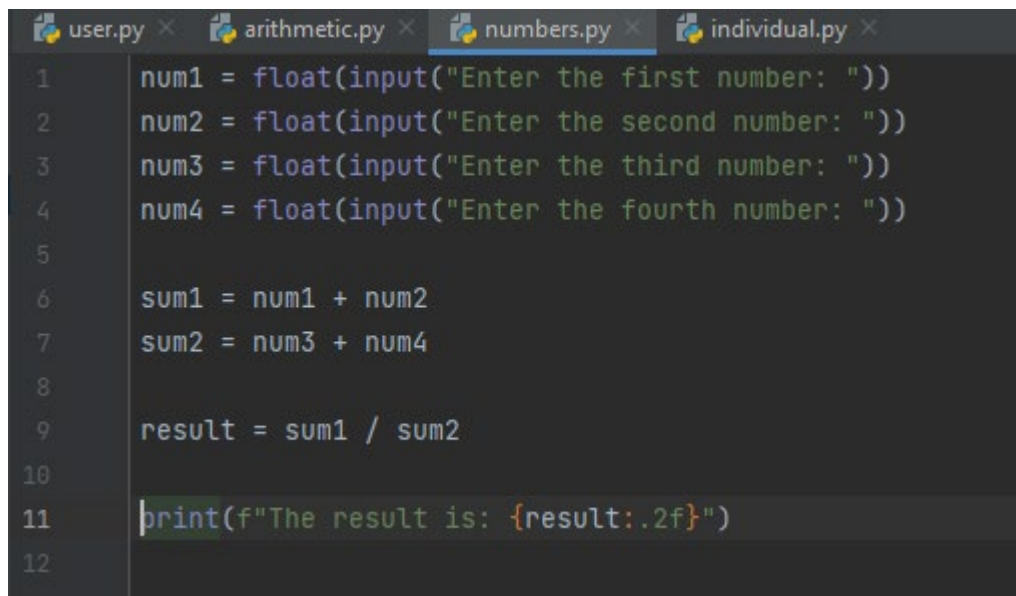
3. Выполнил задание «arithmetic»:



```
1 print("Solve the example: 4 * 100 - 54")
2
3 user_answer = input("Your answer: ")
4
5 correct_answer = 4 * 100 - 54
6
7 print(f"The correct answer is: {correct_answer}")
8 print(f"Your answer was: {user_answer}")
9
```

Рисунок 3. Программа задания «arithmetic»

4. Выполнил задание «numbers»



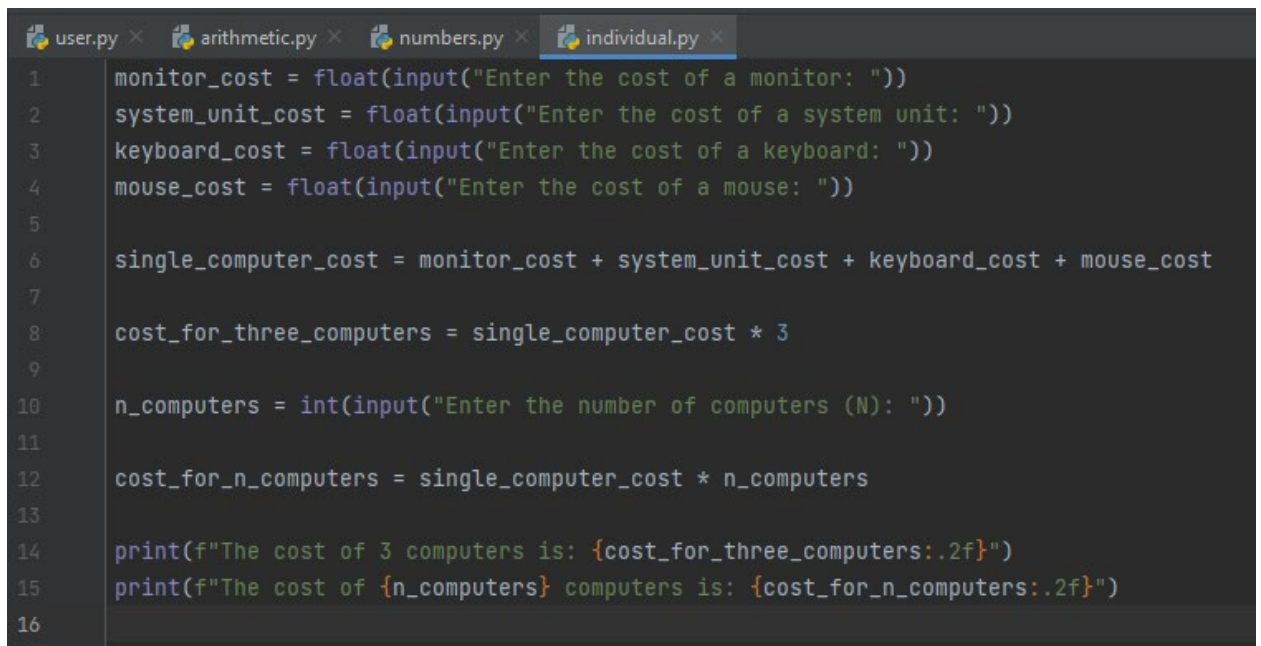
```
1 num1 = float(input("Enter the first number: "))
2 num2 = float(input("Enter the second number: "))
3 num3 = float(input("Enter the third number: "))
4 num4 = float(input("Enter the fourth number: "))
5
6 sum1 = num1 + num2
7 sum2 = num3 + num4
8
9 result = sum1 / sum2
10
11 print(f"The result is: {result:.2f}")
12
```

Рисунок 4. Программа задания «numbers»

5. Выполнил индивидуальное задание:

12. Известна стоимость монитора, системного блока, клавиатуры и мыши. Сколько будут стоить 3 компьютера из этих элементов? N компьютеров?

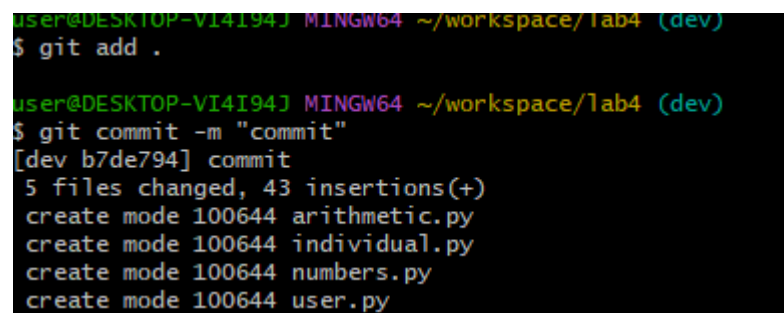
Рисунок 5. Индивидуальное задание в соответствии с вариантом



```
1 monitor_cost = float(input("Enter the cost of a monitor: "))
2 system_unit_cost = float(input("Enter the cost of a system unit: "))
3 keyboard_cost = float(input("Enter the cost of a keyboard: "))
4 mouse_cost = float(input("Enter the cost of a mouse: "))
5
6 single_computer_cost = monitor_cost + system_unit_cost + keyboard_cost + mouse_cost
7
8 cost_for_three_computers = single_computer_cost * 3
9
10 n_computers = int(input("Enter the number of computers (N): "))
11
12 cost_for_n_computers = single_computer_cost * n_computers
13
14 print(f"The cost of 3 computers is: {cost_for_three_computers:.2f}")
15 print(f"The cost of {n_computers} computers is: {cost_for_n_computers:.2f}")
16
```

Рисунок 6. Программа для индивидуального задания

6. Закоммитил изменения

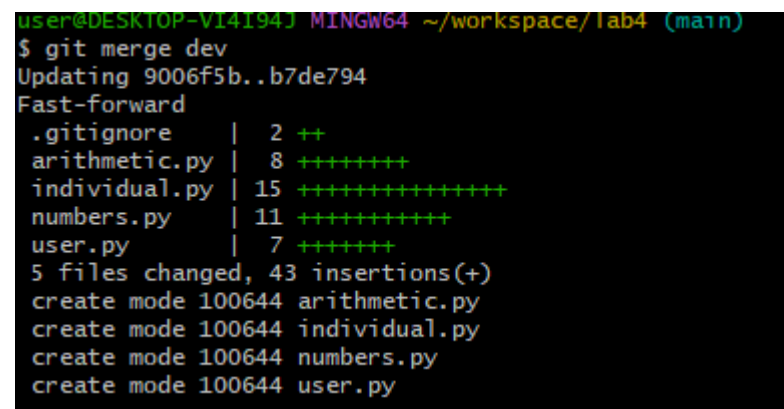


```
user@DESKTOP-VI4I94J MINGW64 ~/workspace/lab4 (dev)
$ git add .

user@DESKTOP-VI4I94J MINGW64 ~/workspace/lab4 (dev)
$ git commit -m "commit"
[dev b7de794] commit
5 files changed, 43 insertions(+)
create mode 100644 arithmetic.py
create mode 100644 individual.py
create mode 100644 numbers.py
create mode 100644 user.py
```

Рисунок 7. Коммит созданных файлов

7. Выполнил слияние ветки разработки с основной веткой



```
user@DESKTOP-VI4I94J MINGW64 ~/workspace/lab4 (main)
$ git merge dev
Updating 9006f5b..b7de794
Fast-forward
 .gitignore      | 2 ++
 arithmetic.py   | 8 ++++++++
 individual.py   | 15 ++++++++
 numbers.py      | 11 ++++++++
 user.py         | 7 ++++++
 5 files changed, 43 insertions(+)
 create mode 100644 arithmetic.py
 create mode 100644 individual.py
 create mode 100644 numbers.py
 create mode 100644 user.py
```

Рисунок 8. Слияние веток

8. Запустил изменения на удаленный сервер:

```
user@DESKTOP-VI4I94J MINGW64 ~/workspace/lab4 (main)
$ git push
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 12 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 1.09 KiB | 1.09 MiB/s, done.
Total 7 (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/lab4.git
9006f5b..b7de794 main -> main
```

Рисунок 9. Пуш на удаленный сервер

Ответы на контрольные вопросы:

1. Основные этапы установки Python в Windows и Linux:

Windows: Скачайте установочный файл с официального сайта Python, запустите его и следуйте инструкциям, не забыв отметить опцию "Add Python to PATH".

Linux: Используйте пакетный менеджер вашей дистрибуции, например, `sudo apt install python3` для Ubuntu.

2. Различие пакета Anaconda от пакета Python:

Anaconda включает в себя множество библиотек и инструментов для научных вычислений и анализа данных, таких как Jupyter Notebook и Conda.

Официальный Python поставляется только с основными библиотеками и инструментами.

3. Проверка работоспособности пакета Anaconda:

Запустите `conda list` в командной строке, чтобы увидеть установленные пакеты.

4. Задание интерпретатора Python в IDE PyCharm:

Перейдите в *File > Settings > Project: <ваш проект> > Python Interpreter* и выберите необходимый интерпретатор.

5. Запуск программы с помощью IDE PyCharm:

Нажмите кнопку *Run* или используйте сочетание клавиш *Shift + F10*.

6. Суть интерактивного и пакетного режимов работы Python:

Интерактивный режим: Прямой ввод и выполнение команд в интерактивной оболочке.

Пакетный режим: Выполнение команд из сохраненного файла скрипта.

7. Почему Python называется языком динамической типизации:

Переменные в Python могут изменять свой тип во время выполнения программы, и тип данных определяется автоматически.

8. Основные типы данных в Python:

int, float, str, list, tuple, dict, set, bool.

9. Создание объектов в памяти и объявление переменных:

Объекты создаются при присваивании значений переменным, а память под них выделяется автоматически.

10. Получение списка ключевых слов в Python:

Выполните команду `import keyword; print(keyword.kwlist)`.

11. Назначение функций `id()` и `type()`:

`id()` возвращает уникальный идентификатор объекта.

`type()` возвращает тип объекта.

12. Изменяемые и неизменяемые типы в Python:

Изменяемые: list, dict, set.

Неизменяемые: int, float, str, tuple.

13. Отличие операций деления и целочисленного деления:

`/` выполняет обычное деление.

`//` выполняет целочисленное деление.

14. Средства Python для работы с комплексными числами:

Встроенные функции и модуль `cmath`.

15. Назначение и функции библиотеки `math`:

Библиотека `math` предоставляет математические функции, такие как `sqrt`, `sin`, `cos`.

Модуль `cmath` аналогичен `math`, но для комплексных чисел.

16. Назначение параметров `sep` и `end` в функции `print()`:

`sep` задает разделитель между аргументами.

`end` задает окончание строки.

17. Назначение метода `format()` и другие средства форматирования строк:

Метод `format()` используется для форматирования строк.

Альтернативы: f-строки, оператор `%`.

18. Ввод значения целочисленной и вещественной переменной с консоли:

```
int_value = int(input("Введите целое число: "))
```

```
float_value = float(input("Введите вещественное число: "))
```

Вывод: в ходе выполнения лабораторной работы был исследован процесс установки и базовые конструкции и возможности языка Python.