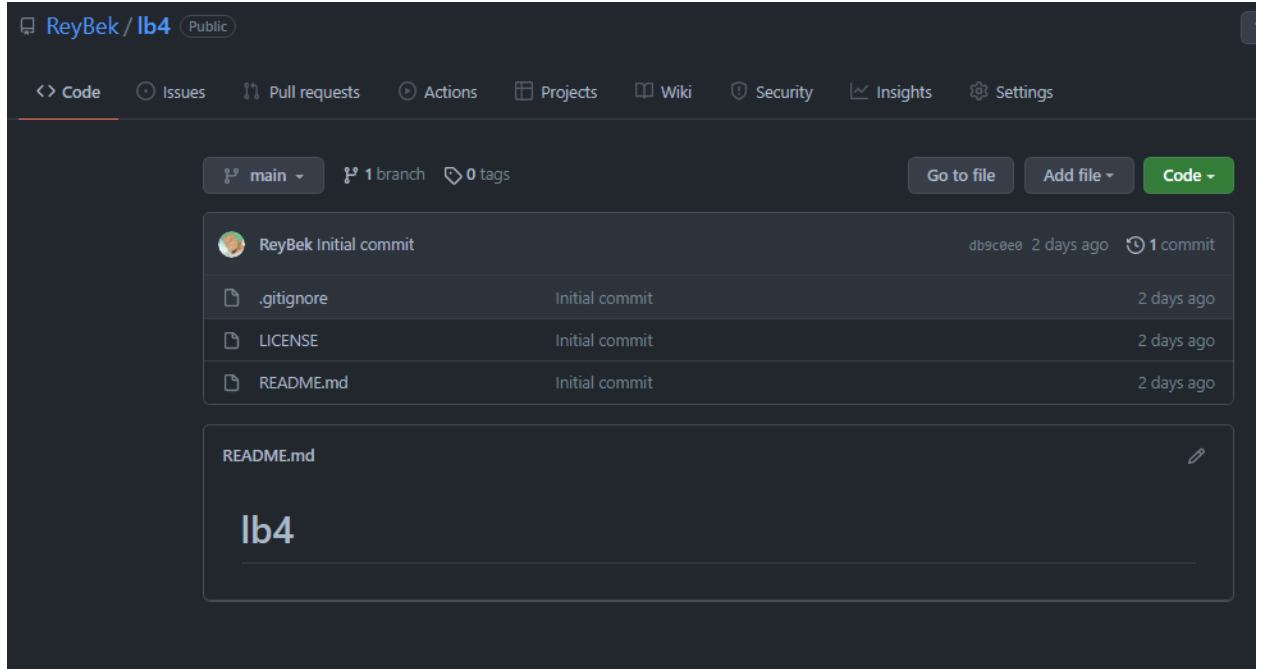


Отчет №4 по дисциплине
“Основы программной инженерии”
Группа: ПИЖ-б-о-20-1, Бекишев М. М.

ВЫПОЛНЕНИЕ

1) Создание репозитория:



2) Работа программы user.py

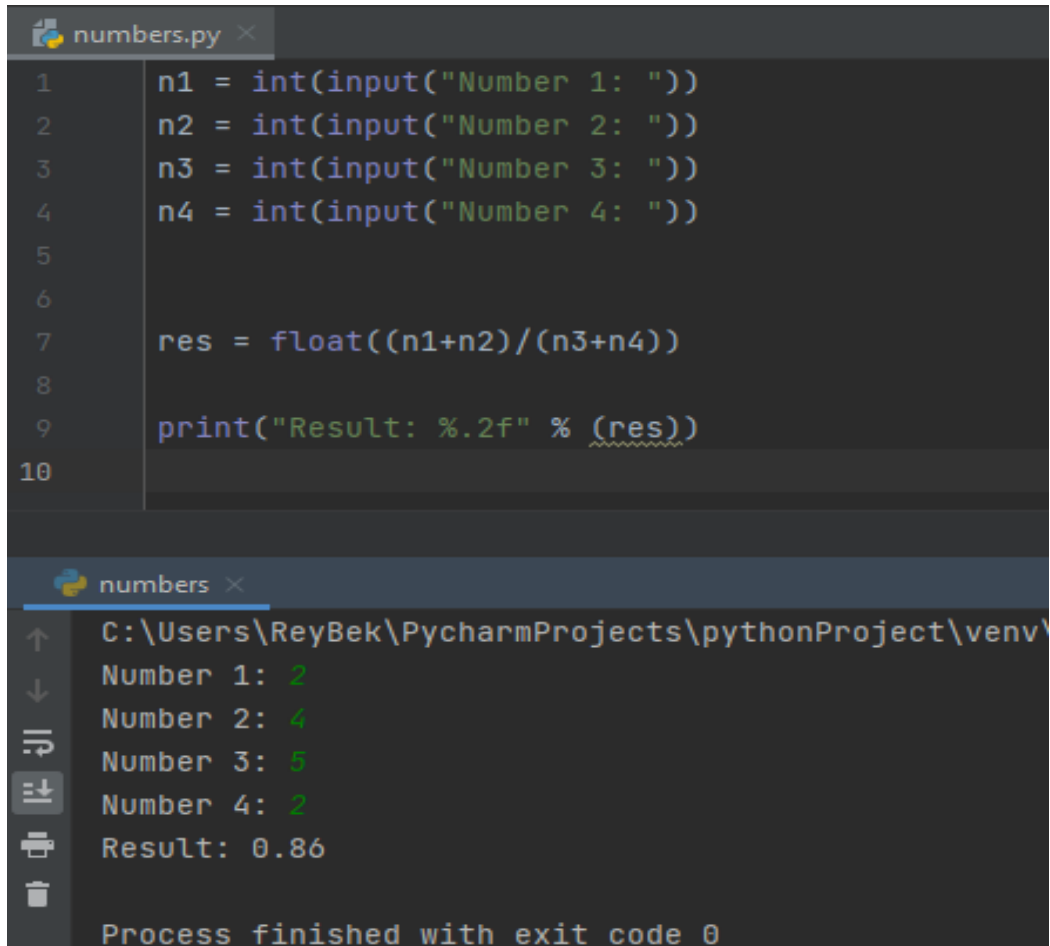
```
user.py
1 name = input("Your name is: ")
2 age = int(input("Your age: "))
3 livingPlace = input("Where you live: ")
4 print("This is %s\nIt is %d\n(S)he live in %s" % (name, age, livingPlace))

user
C:\Users\ReyBek\PycharmProjects\pythonProject\venv\Scripts\python.exe C:/Users/ReyBek/PycharmProjects/pythonProject/user.py
Your name is: Musaid
Your age: 18
Where you live: stv
This is Musaid
It is 18
(S)he live in stv

Process finished with exit code 0
|
```

Отчет №4 по дисциплине
“Основы программной инженерии”
Группа: ПИЖ-б-о-20-1, Бекишев М. М.

3) Работа программы numbers.py



The screenshot displays a Python IDE with two panels. The top panel shows the source code for a file named `numbers.py`. The code consists of ten lines: four lines for inputting integers `n1`, `n2`, `n3`, and `n4`; one line for calculating the result `res` as a float of the average of `(n1+n2)` and `(n3+n4)`; and one line for printing the result with two decimal places. The bottom panel shows the execution output for a file named `numbers`. It displays the prompts and inputs for each number, followed by the calculated result, and a message indicating the process finished successfully.

```
1 n1 = int(input("Number 1: "))
2 n2 = int(input("Number 2: "))
3 n3 = int(input("Number 3: "))
4 n4 = int(input("Number 4: "))
5
6
7 res = float((n1+n2)/(n3+n4))
8
9 print("Result: %.2f" % (res))
10
```

numbers

C:\Users\ReyBek\PycharmProjects\pythonProject\venv\

Number 1: 2

Number 2: 4

Number 3: 5

Number 4: 2

Result: 0.86

Process finished with exit code 0

Отчет №4 по дисциплине
“Основы программной инженерии”
Группа: ПИЖ-б-о-20-1, Бекишев М. М.

4) Работа программы arithmetic.py

```
arithmetic.py
1 rightAns = int(4*100-54)
2 print("Can YOU solve this:\n4 * 100 - 54")
3 userAns = int(input("Write your answer there: "))
4 if userAns == rightAns:
5     print("Your answer is absolutely right! Right answer is %d" % (rightAns))
6 else:
7     print("You was wrong :(\nThe right answer is %d\nYour answer is %d" % (rightAns, userAns))
```

arithmetic

C:\Users\ReyBek\PycharmProjects\pythonProject\venv\Scripts\python.exe C:/Users/ReyBek/PycharmProjects/pythonProject/arithmetic.py

Can YOU solve this:
4 * 100 - 54
Write your answer there: 6
You was wrong :(
The right answer is 346
Your answer is 6

Process finished with exit code 0

5) Работа программы individual.py

```
individual.py
1 from math import sqrt
2 a = float(input('a='))
3 b = float(input('b='))
4 d = abs(a)
5 c = abs(b)
6 print("Среднее арифметическое: ")
7 print((d + c)/2)
8 print("Среднее геометрическое ")
9 print(sqrt(d*c))
```

individual

C:\Users\ReyBek\PycharmProjects\pythonProject\venv\Scripts\python.exe C:/Users/ReyBek/PycharmProjects/pythonProject/individual.py

a= -4
b= -4
Среднее арифметическое:
4.0
Среднее геометрическое
4.0

Process finished with exit code 0

ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Скачать дистрибутив; Запустить exe файл и следовать установщику (для Windows) / Python уже или входит в состав дистрибутива (для Linux)
2. Anaconda более удобная чем пакет python. Лучше подходит для изучения языка
3. Для проверки работоспособности anaconda надо запустить командный процессор с поддержкой виртуальных окружений anaconda. В появившейся строке надо ввести команду jupyter notebook
4. При создании нового проекта будет предложено выбрать необходимый интерпретатор
5. Запустить код можно через меню правой кнопки мыши >> run или через значок зеленого треугольник сверху справа интерфейса
6. В интерактивном режиме код можно вводить и выполнять строками, а в пакетном режиме файлы будут интерпретированы построчно целиком.
7. Так как тип переменных определяется при выполнении программы
8. Неопределенные, логические, числа, списки, строки, бинарные списки, множества, словари
9. Для того, чтобы объявить и сразу инициализировать переменную необходимо написать её имя, потом поставить знак равенства и значение, с которым эта переменная будет создана. значение 5 в рамках языка Python по сути своей является объектом. Каждый объект имеет три атрибута – это идентификатор, значение и тип. Идентификатор – это уникальный признак объекта, позволяющий отличать объекты друг от друга, а значение – непосредственно информация, хранящаяся в памяти, которой управляет интерпретатор. При инициализации переменной создается целочисленный объект 5, данный объект имеет некоторый идентификатор, значение: 5, и тип: целое число; посредством оператора “=” создается ссылка между переменной b и целочисленным объектом 5 10. нужно подключить модуль keyword и воспользоваться командой keyword.kwlist.
11. id() – Чтобы посмотреть на какой объект ссылается переменная type() – для определения типа переменной
12. Это такие типы данных, которые можно или нельзя изменять в процессе выполнения программы

Отчет №4 по дисциплине
“Основы программной инженерии”
Группа: ПИЖ-б-о-20-1, Бекишев М. М.

13. Деление – точное деление; целочисленное деление – получение целой части (деление без остатка)

14. Комплексные числа в python можно создавать(`complex()`), складывать, вычитать, умножать, делить и возводить в степень. Можно извлечь действительную и мнимую части (`.real` и `.imag`). Получить комплексно сопряжённое число (`conjugate()`)

15. `Math` дает доступ к большому количеству математических функций. `Math.ceil()` - Возвращает ближайшее целое число большее, чем `x`. `Math.fabs()` - Возвращает абсолютное значение числа. `Math.factorial()` - Вычисляет факториал `x`. `Math.floor()` - Возвращает ближайшее целое число меньшее, чем `x`. `Math.exp()` - Вычисляет e^{**x} . `Math.log2()` - Логарифм по основанию 2. `Math.log10()` - Логарифм по основанию 10. `Math.log()` - По умолчанию вычисляет логарифм по основанию `e`, дополнительно можно указать основание логарифма. `Math.pow()` - Вычисляет значение `x` в степени `y`. `Math.sqrt()` - Корень квадратный от `x`. `Math.cos()` - Косинус от `x`. `Math.sin()` - Синус от `x`. `Math.tan()` - Тангенс от `x`. `Math.acos()` - Арккосинус от `x`. `Math.asin()` - Арксинус от `x`. `Math.atan()` - Арктангенс от `x`. `Math.pi` - Число π . `Math.e` - Число e . Модуль `cmath` – предоставляет функции для работы с комплексными числами.

16. через параметр `sep` можно указать отличный от пробела разделитель строк; Параметр `end` позволяет указывать, что делать, после вывода строки

17. Форматирование может выполняться с помощью строкового метода `format`, с помощью оператора `%`

18. через команду `input()`