Северо-Кавказский федеральный университет

Институт математики и информационных технологий

ОТЧЕТ

о выполнении лабораторной работы №4

по дисциплине

«Основы Программной Инженерии»

Выполнил:

Маняхин Тимур Александрович

студент 2 курса, ПИЖ-б-о-20-1 группы
бакалавриата «Программная инженерия»

очной формы обучения

Ставрополь, 2021

СКРИНШОТЫ РЕЗУЛЬТАТОВ РАБОТЫ ПРОГРАММ

```
nameuser = input("What is your name?")
ageuser = input("How old are you?")
countryuser = input("Where are you live?")
print("This is '{0}'".format(nameuser))
print("It is '{0}'".format(ageuser))
print("(S)he live in '{0}'".format(countryuser))
```

Рисунок 1.1 – программа user

```
What is your name? tim
How old are you? 18
Where are you live? Rus
This is ' tim'
It is ' 18'
(S)he live in ' Rus'
```

Рисунок 1.2 – результат работы программы user

```
answear1 = int(input("solve the example 4 * 100 - 54 and enter the answer"))
print("your answear {0}, right {1}".format(answear1, 4 * 100 - 54))
```

Рисунок 1.3 – программа arithmetic

```
solve the example 4 * 100 - 54 and enter the answer 243
your answear 243, right 346
```

Рисунок 1.4 – результат работы программы arithmetic

```
a = int(input("enter the first  number: "))
b = int(input("enter the second number: "))
c = int(input("enter the third  number: "))
d = int(input("enter the fourth number: "))
e = a + b
f = c + d
g = e / f
print(f'{g:.2f}')
```

Рисунок 1.5 – программа numbers

```
enter the first  number: 23
enter the second number: 5
enter the third  number: 2
enter the fourth number: 13
1.87
```

Рисунок 1.6 – результат работы программы numbers

```
v1 = int(input("enter speed 1"))
v2 = int(input("enter speed 2"))
s = int(input("enter distance"))
print(s / (v1 + v2))
```

Рисунок 1.7 – программа individual

```
enter speed 1 60
enter speed 2 40
enter distance 120
1.2
```

Рисунок 1.8 – результат работы программы individual

```
angle = float(input("enter angle"))
minutes = int(2*angle) % 60
hours = int(angle/30)
print("minutes {0}, hours {1}".format(minutes, hours))
```

Рисунок 1.9 – программа increasedcomplexity

```
enter angle 70
minutes 20, hours 2
```

Рисунок 1.10 – результат работы программы increasedcomplexity

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
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.sit() - 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 пи.
    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()