

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №1.2
дисциплины «Основы кроссплатформенного программирования»

Выполнил:
Волошин Алексей
Вадимович
1 курс, группа ИТС-б-о-22-1,
11.03.02 «Инфокоммуникационные
технологии и системы связи»,
направленность (профиль)
«Инфокоммуникационные системы и
сети», очная форма обучения

(подпись)

Руководитель практики:
Воронкин Р.А., канд. техн. наук, доцент
кафедры инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

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

Ход работы:

Создал новый репозиторий на GitHub.

Выбрал в качестве языка программирования Python.

Задание 1.

8. Напишите программу (файл progа.py), которая запрашивала бы у пользователя:

его имя (например, "What is your name?")

возраст ("How old are you?")

место жительства ("Where are you live?")

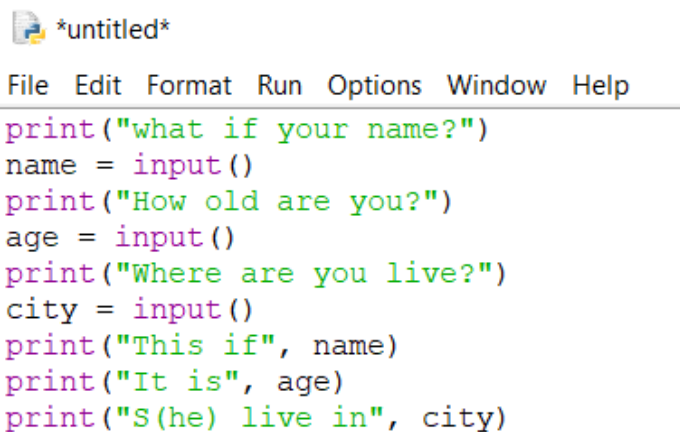
После этого выводила бы три строки:

"This is `имя`"

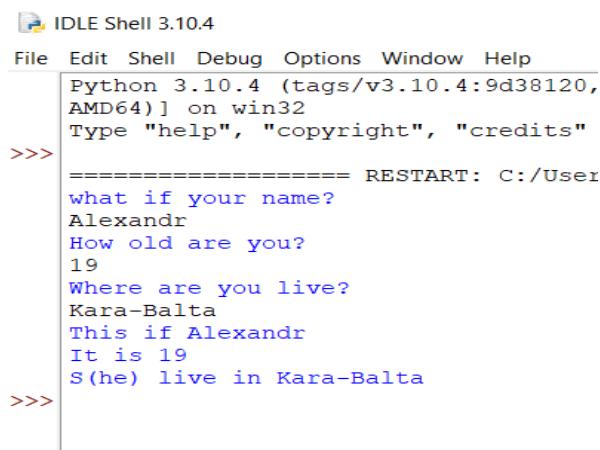
"It is `возраст`"

"(S)he live in `место_жительства`"

Сам код выглядит так:



```
*untitled*
File Edit Format Run Options Window Help
print("what if your name?")
name = input()
print("How old are you?")
age = input()
print("Where are you live?")
city = input()
print("This if", name)
print("It is", age)
print("S(he) live in", city)
```

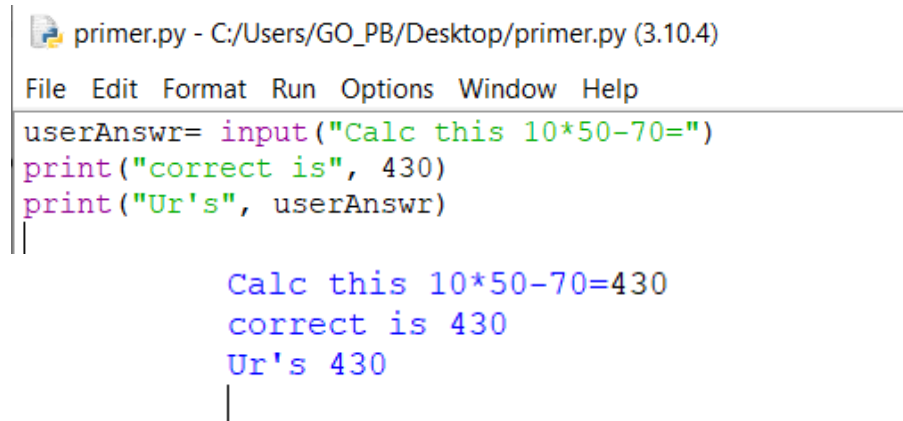


```
IDLE Shell 3.10.4
File Edit Shell Debug Options Window Help
Python 3.10.4 (tags/v3.10.4:9d38120,
AMD64) on win32
Type "help", "copyright", "credits"
>>>
===== RESTART: C:/User
what if your name?
Alexandr
How old are you?
19
Where are you live?
Kara-Balta
This if Alexandr
It is 19
S(he) live in Kara-Balta
>>>
```

Рисунок 1. Код программы progа.py и его результат.

Задание 2.

Напишу программу (файл primer.py), которая предлагала бы пользователю решить пример $10 * 50 - 70$. Потом выводила бы на экран правильный ответ и ответ пользователя.



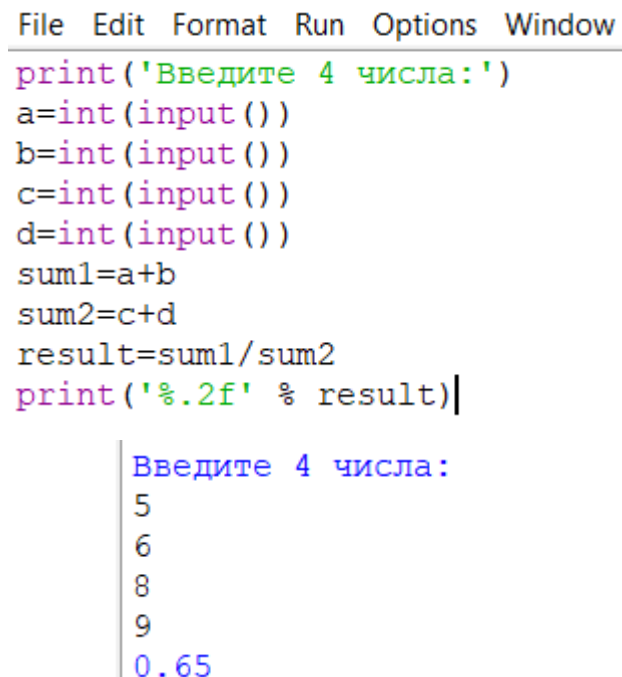
```
primer.py - C:/Users/GO_PB/Desktop/primer.py (3.10.4)
File Edit Format Run Options Window Help
userAnswr= input("Calc this 10*50-70=")
print("correct is", 430)
print("Ur's", userAnswr)

Calc this 10*50-70=430
correct is 430
Ur's 430
```

Рисунок 2. Код программы primer.py и результат

Задание 3.

Запросите у пользователя четыре числа (файл chisla.py). Отдельно сложите первые два и отдельно вторые два. Разделите первую сумму на вторую. Выведите результат на экран так, чтобы ответ содержал две цифры после запятой.



```
File Edit Format Run Options Window
print('Введите 4 числа:')
a=int(input())
b=int(input())
c=int(input())
d=int(input())
sum1=a+b
sum2=c+d
result=sum1/sum2
print('%.2f' % result)

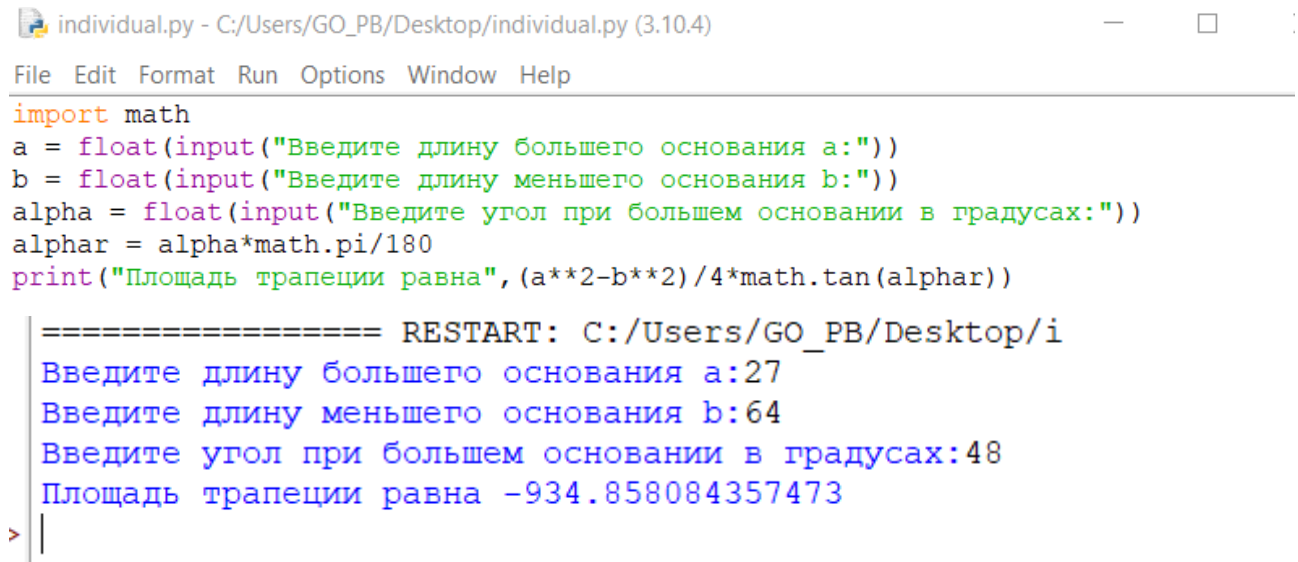
Введите 4 числа:
5
6
8
9
0.65
```

Рисунок 3. Код программы chisla.py и его результат

Задача 4.

Напишите программу (файл individual.py) для решения индивидуального задания.

8. Даны основания равнобедренной трапеции и угол при большем основании. Найти площадь трапеции.



The screenshot shows a text editor window titled 'individual.py - C:/Users/GO_PB/Desktop/individual.py (3.10.4)'. The editor contains a Python script that calculates the area of an isosceles trapezoid. The script prompts the user for the length of the larger base (a), the length of the smaller base (b), and the angle at the larger base (alpha) in degrees. It then calculates the area using the formula $\frac{(a^2 - b^2)}{4 \cdot \tan(\alpha)}$ and prints the result. Below the script, the execution output is shown, indicating a restart and the input values: a=27, b=64, alpha=48. The calculated area is -934.858084357473.

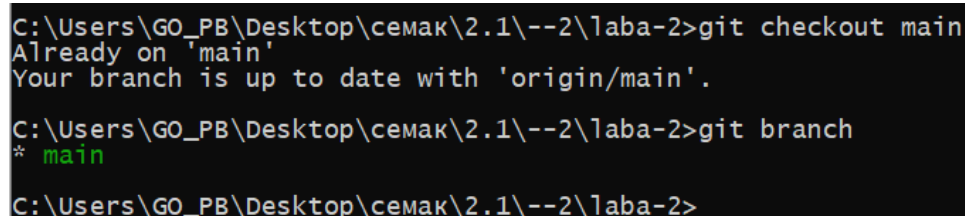
```
individual.py - C:/Users/GO_PB/Desktop/individual.py (3.10.4)
File Edit Format Run Options Window Help
import math
a = float(input("Введите длину большего основания a:"))
b = float(input("Введите длину меньшего основания b:"))
alpha = float(input("Введите угол при большем основании в градусах:"))
alphar = alpha*math.pi/180
print("Площадь трапеции равна", (a**2-b**2)/4*math.tan(alphar))

===== RESTART: C:/Users/GO_PB/Desktop/i
Введите длину большего основания a:27
Введите длину меньшего основания b:64
Введите угол при большем основании в градусах:48
Площадь трапеции равна -934.858084357473
> |
```

Рисунок 4. Код программы для individual.py и его результат

После выполнения всех заданий надо произвести слияние веток.

Для этого выполняю команду «git branch», чтобы узнать на какой ветке нахожусь.



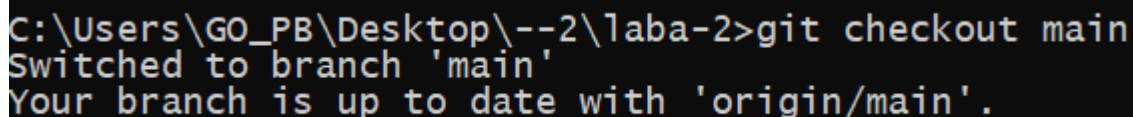
The screenshot shows a terminal window with the following commands and output:

```
C:\Users\GO_PB\Desktop\семак\2.1\--2\laba-2>git checkout main
Already on 'main'
Your branch is up to date with 'origin/main'.

C:\Users\GO_PB\Desktop\семак\2.1\--2\laba-2>git branch
* main

C:\Users\GO_PB\Desktop\семак\2.1\--2\laba-2>
```

Далее переключаюсь на ветку master командой «git checkout master» и командой «git branch» проверяю перешёл ли я на эту ветку



The screenshot shows a terminal window with the following command and output:

```
C:\Users\GO_PB\Desktop\--2\laba-2>git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
```

После этого ввожу команду «git merge develop» для слияния.

```
C:\Users\GO_PB\Desktop\--2\laba-2>git merge develop
Updating e5eb953..6a54dc6
Fast-forward
 README.md      | 4 +++-
 arithmetic.py  | 3 +++
 individual.py  | 7 +++++++
 numbers.py     | 9 ++++++++
 user.py        | 9 ++++++++
 5 files changed, 31 insertions(+), 1 deletion(-)
 create mode 100644 arithmetic.py
 create mode 100644 individual.py
 create mode 100644 numbers.py
 create mode 100644 user.py
```

Контрольные вопросы:

1. Опишите основные этапы установки Python в Windows и Linux.

Windows:

1. Запустите скачанный установочный файл.

2. Выберите способ установки.

3. Отметьте необходимые опции установки (доступно при выборе Customize installation)

4. Выберите место установки (доступно при выборе Customize installation)

5. После успешной установки вас ждет следующее сообщение.

Настройка прошла успешно.

Linux:

Python уже входит в состав дистрибутива. Набираем в терминале `$ python3`. Если при попытке запустить Python, выдается сообщение о том, что он не установлен, или установлен, но не тот, вас есть два пути: а) собрать Python из исходников; б) взять из репозитория.

Для установки из репозитория в Ubuntu воспользуйтесь командой `$ sudo apt-get install python3`

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

Anaconda - это дистрибутив языков программирования Python и R, а Python это язык программирования общего назначения высокого уровня.

3. Как осуществить проверку работоспособности пакета Anaconda?

Пуск>Anaconda3 (64-bit)>Anaconda Prompt. В появившейся командной строке необходимо ввести `>jupyter notebook`. После чего запустится веб-сервер и среда разработки в браузере. Затем на кнопку New (в правом углу окна) и в появившемся списке выберете Python. В результате будет создана новая страница в браузере

4. Как задать используемый интерпретатор языка Python в IDE PyCharm?

1. Запустите PyCharm и выберите CreateNewProject в появившемся окне.

2. Укажите путь до проекта Python и интерпретатор, который будет использоваться для запуска и отладки.

3. Добавьте Python файл в проект.

5. Как осуществить запуск программы с помощью IDEPyCharm?

1. Введите код программы.

2. Запустите программу.

3. В результате должно открыться окно с выводом программы.

6. В чем суть интерактивного и пакетного режимов работы Python?

Интерактивный режим (REPL) помогает нам проверять идеи и придумывать решение задач. Пакетный режим выполняет готовую программу.

В интерактивном режиме он постоянно выполняет одни и те же действия: читает команду программиста, выполняет ее и печатает результат. Такой режим называют Циклом Чтения-Выполнения-Печати

Интерпретатор выполняет команды одну за одной и в конце завершает свою работу. Такой режим работы интерпретатора называют *пакетным*.

7. Почему язык программирования Python называется языком динамической типизации?

Приём, используемый в языках программирования и языках спецификации, при котором переменная связывается с типом в момент присваивания значения, а не в момент объявления переменной. Таким образом, в различных участках программы одна и та же переменная может принимать значения разных типов.

8. Какие существуют основные типы в языке программирования Python?

Числа, списки, кортежи, строки, множества, словари, преобразование типов данных.

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

Написать её имя, потом поставить знак равенства и значение, с которым эта переменная будет создана.

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.

10. Как получить список ключевых слов в Python?

Подключить модуль *keyword* и воспользоваться командой *keyword.kwlist*.

11. Каково назначение функций *id()* и *type()*?

id() - возвращает целое число, представляющее его идентификатор
type() возвращает тип объекта

12. Что такое изменяемые и неизменяемые типы в Python.

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

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

13. Чем отличаются операции деления и целочисленного деления?

Целочисленное деление (*div*) отличается от обычной операции деления тем, что возвращает целую часть частного, дробная часть отбрасывается. Перед выполнением операции оба операнда округляются до целых значений. Результат целочисленного деления всегда равен нулю, если делимое меньше делителя.

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

Модуль `cmath`. Модуль содержит набор функций для обработки комплексных чисел.

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

Этот модуль обеспечивает доступ к математическим функциям, определенным стандартом C. Эти функции нельзя использовать с комплексными числами; используйте одноименные функции из `cmath` модуля, если вам требуется поддержка комплексных чисел.

16. Каково назначение именных параметров `sep` и `end` в функции `print()`?

`sep` – с помощью этого параметра вы можете указать разделитель строк. А по умолчанию в качестве разделителя используется пробел

`end` – этот параметр позволяет указать, что нужно добавить после последней строки. По умолчанию добавляется управляющий символ `'\n'` (перевод строки)

17. 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.

Пользователь может изменить положение строки в выводе

С оператором `%`, метод `format()`, f-строки

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

`input()`

Вывод: Я изучил базовые возможности языка Python. И выполнил задания на этом языке программирования.