

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

Институт цифрового развития
Кафедра инфокоммуникаций

«Основы языка Python»

ОТЧЕТ
по лабораторной работе №4
дисциплины
«Основы программной инженерии»

Выполнил:

Гълбачева Доротея Андреева
2 курс, группа ПИЖ-б-о-21-1,
09.03.04 «Программная инженерия»,
направленность (профиль) «Разработка
и сопровождение программного
обеспечения», очная форма обучения

(подпись)

Проверил:

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2022 г.

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

1. Напишите программу (файл user.py), которая запрашивала бы у пользователя:

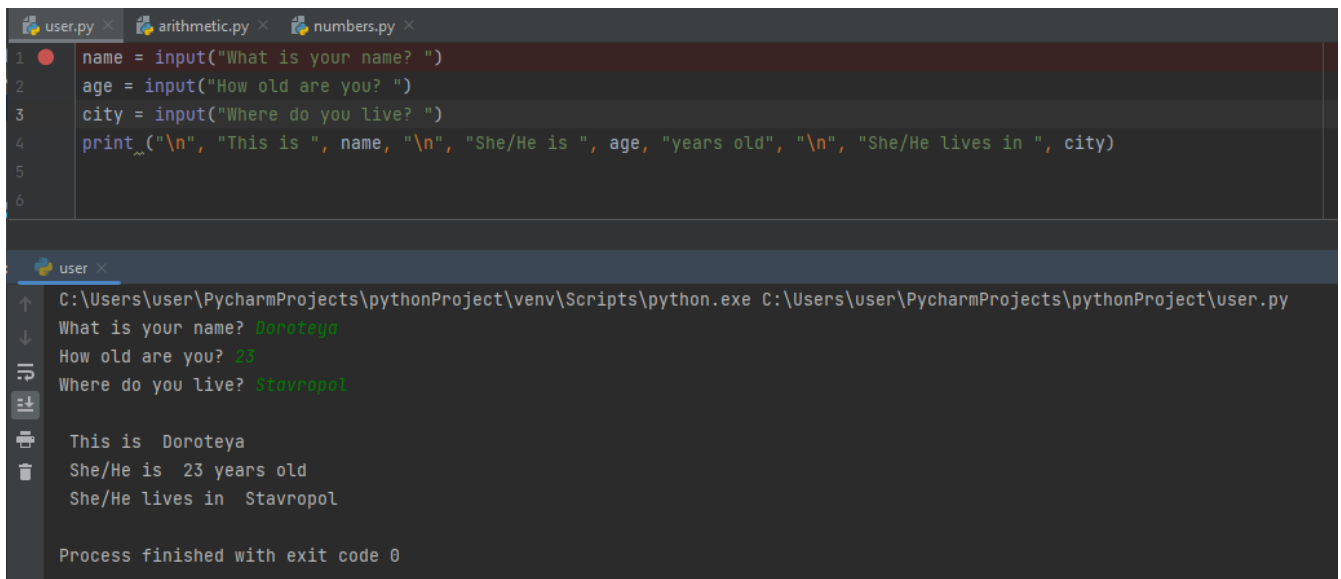
Его имя (например, "What is your name?")

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`"
```



The screenshot shows a Python IDE with three tabs: user.py, arithmetic.py, and numbers.py. The user.py tab is active, displaying the following code:

```
1 name = input("What is your name? ")  
2 age = input("How old are you? ")  
3 city = input("Where do you live? ")  
4 print("\n", "This is ", name, "\n", "She/He is ", age, "years old", "\n", "She/He lives in ", city)  
5  
6
```

Below the code editor, the output window shows the execution of the program. The prompts and user input are as follows:

```
What is your name? Doroteya  
How old are you? 23  
Where do you live? Stavropol
```

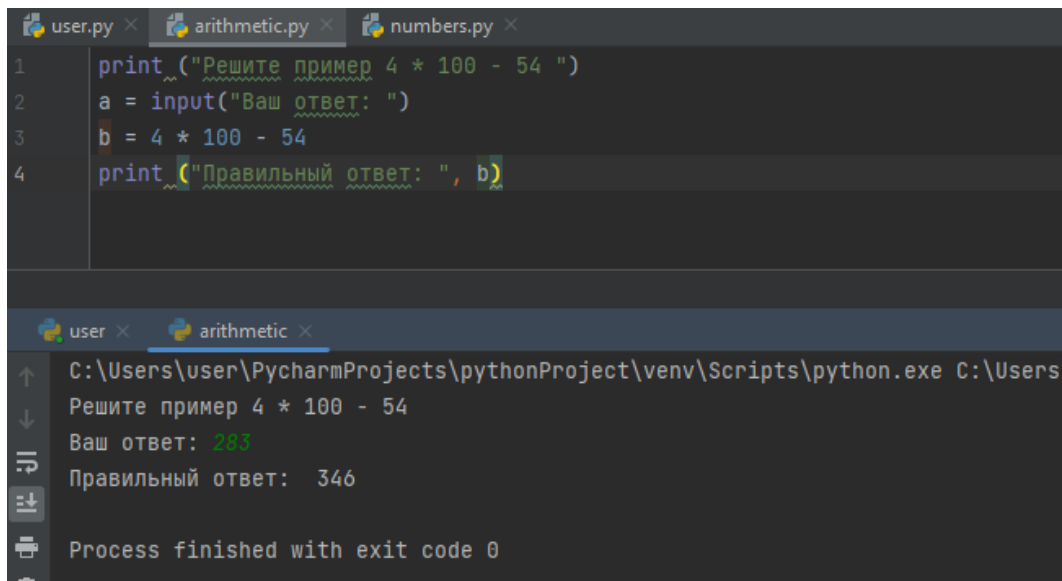
The output of the program is:

```
This is Doroteya  
She/He is 23 years old  
She/He lives in Stavropol
```

The process finished with exit code 0.

Рисунок 4.1 – Результат работы программы user.py

2. Напишите программу (файл arithmetic.py), которая предлагала бы пользователю решить пример $4 * 100 - 54$. Потом выводила бы на экран правильный ответ и ответ пользователя. Подумайте, нужно ли здесь преобразовывать строку в число.

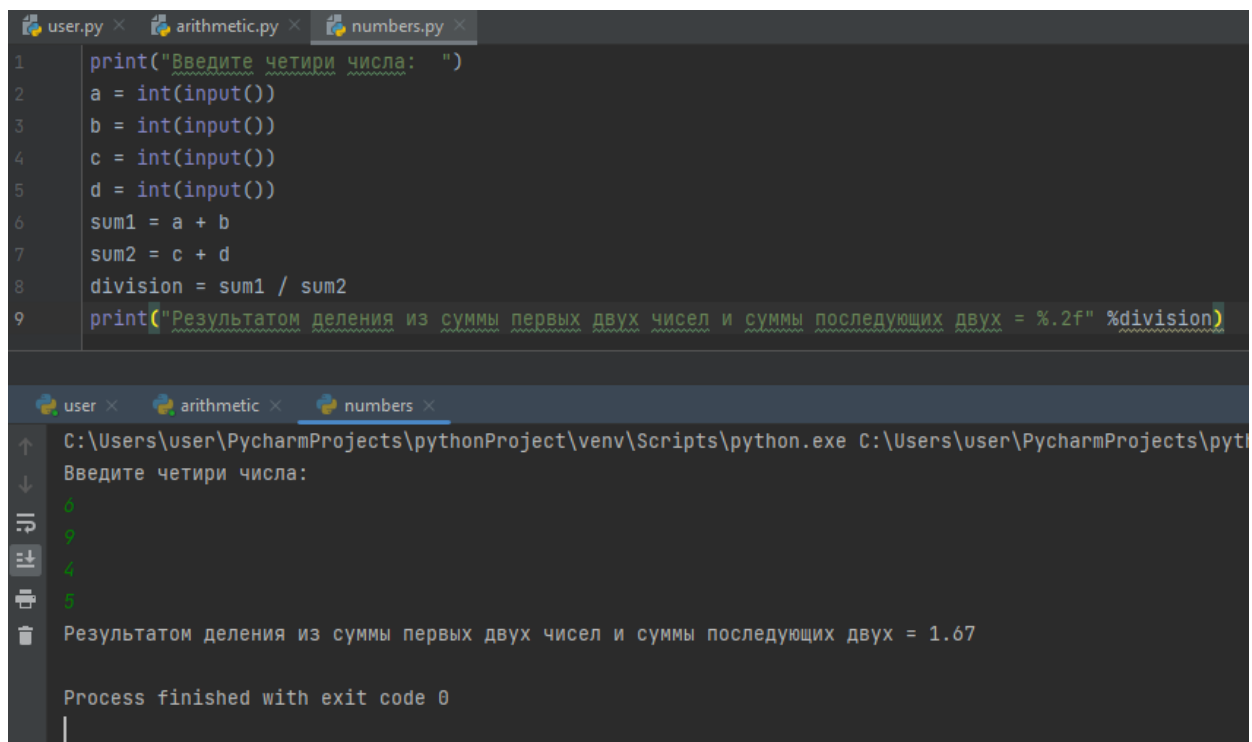


```
user.py x arithmetic.py x numbers.py x
1 print ("Решите пример 4 * 100 - 54 ")
2 a = input("Ваш ответ: ")
3 b = 4 * 100 - 54
4 print ("Правильный ответ: ", b)

C:\Users\user\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users
Решите пример 4 * 100 - 54
Ваш ответ: 283
Правильный ответ: 346
Process finished with exit code 0
```

Рисунок 4.1 – Результат работы программы arithmetic.py

3. Запросите у пользователя четыре числа (файл numbers.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.



```
user.py x arithmetic.py x numbers.py x
1 print("Введите четыре числа: ")
2 a = int(input())
3 b = int(input())
4 c = int(input())
5 d = int(input())
6 sum1 = a + b
7 sum2 = c + d
8 division = sum1 / sum2
9 print("Результатом деления из суммы первых двух чисел и суммы последующих двух = %.2f" %division)

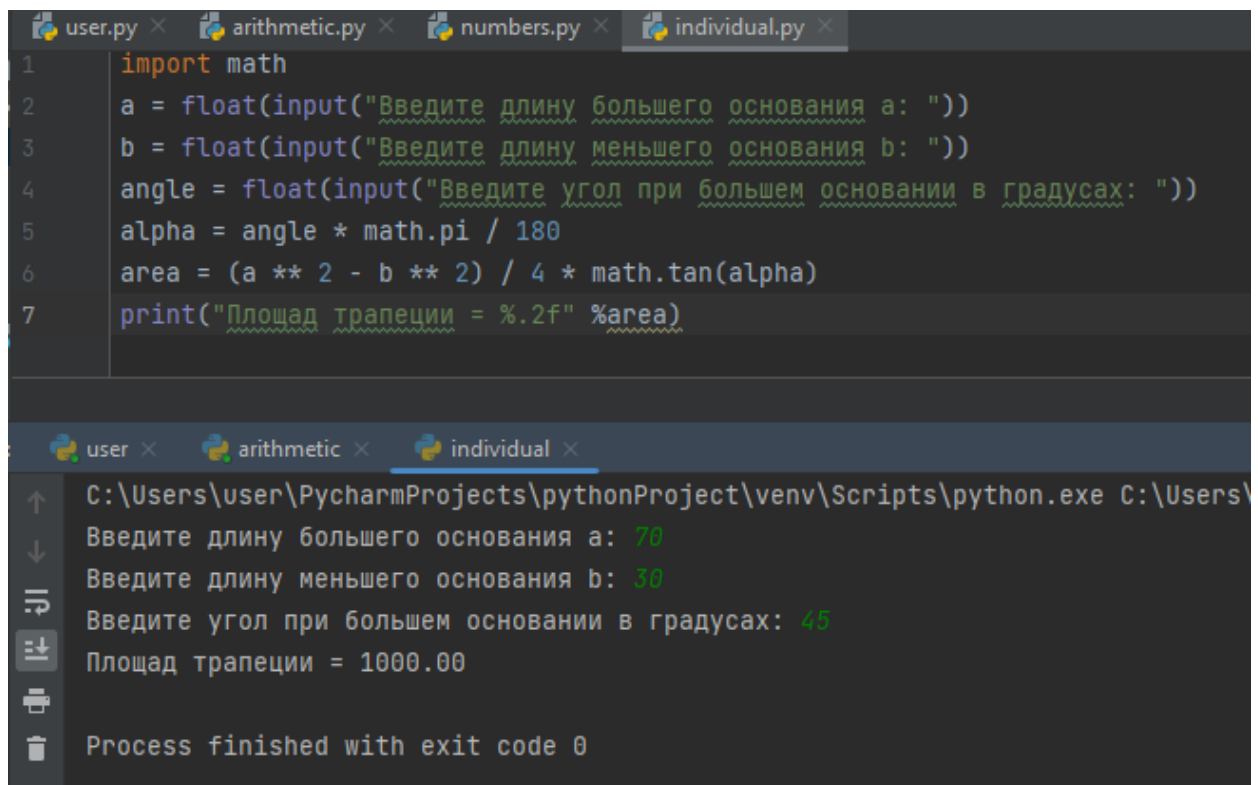
C:\Users\user\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\user\PycharmProjects\pyt
Введите четыре числа:
6
9
4
5
Результатом деления из суммы первых двух чисел и суммы последующих двух = 1.67
Process finished with exit code 0
|
```

Рисунок 4.3 – Результат работы программы numbers.py

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

Вариант 8

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



The screenshot displays a Python IDE with two windows. The top window, titled 'individual.py', contains the following code:

```
1 import math
2 a = float(input("Введите длину большего основания a: "))
3 b = float(input("Введите длину меньшего основания b: "))
4 angle = float(input("Введите угол при большем основании в градусах: "))
5 alpha = angle * math.pi / 180
6 area = (a ** 2 - b ** 2) / 4 * math.tan(alpha)
7 print("Площадь трапеции = %.2f" % area)
```

The bottom window shows the execution output for the 'individual' script:

```
C:\Users\user\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\
Введите длину большего основания a: 70
Введите длину меньшего основания b: 30
Введите угол при большем основании в градусах: 45
Площадь трапеции = 1000.00
Process finished with exit code 0
```

Рисунок 4.4 – Результат работы программы individual.py

5. Выполните коммит файлов user.py, arithmetic.py, numbers.py и individual.py в репозиторий git в ветку для разработки.

```

user@DESKTOP-KOLBPRO MINGW64 ~/LabR_4 (main)
$ git log
commit 85ed31ff957677d740614e06e6505a5f83e190d9 (HEAD -> main, origin/main, origin/HEAD)
Author: DoroteyaGalbacheva <dgalbacheva@gmail.com>
Date: Tue Jan 24 09:10:06 2023 +0300

    individual.py

commit f75379ba1d7c24c4a980e3553dd2eb992c248083
Author: DoroteyaGalbacheva <dgalbacheva@gmail.com>
Date: Tue Jan 24 09:09:39 2023 +0300

    numbers.py

commit 9b22d5e1a5cda9d80cbb6124e0258344d5b42a7
Author: DoroteyaGalbacheva <dgalbacheva@gmail.com>
Date: Tue Jan 24 09:09:09 2023 +0300

    arithmetic.py

commit fee99cd9af550726ffc41f7fc150baf3b68558cd
Author: DoroteyaGalbacheva <dgalbacheva@gmail.com>
Date: Tue Jan 24 09:07:54 2023 +0300

    user.py

```

Рисунок 4.5 – Выполнение коммит в репозиторий git

Ответы на вопросы:

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

Windows: зайти на официальный сайт, скачать exe файл, установить.

Linux: 1) собрать Python из исходников; 2) взять из репозитория при помощи команды \$ 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.

Кроме того, Anaconda в основном используется для анализа данных и машинного обучения. Python используется во множестве приложений, таких как встроенные системы, компьютерное зрение, веб-разработка, сетевое программирование, включая машинное обучение и науку о данных.

3. Как осуществить проверку работоспособности пакета Anaconda?

Пуск → Anaconda3 (64-bit) → Anaconda Prompt. В появившейся командной строке необходимо ввести: `jupyter notebook`

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

File → Settings → python interpreter, нажать на add Interpreter, указать путь к установленному интерпретатору языка python

5. Как осуществить запуск программы с помощью IDE PyCharm?

ПКМ по рабочей области, нажать пункт «Run (имя файла)»

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

Интерактивный режим позволяет использовать python, как калькулятор, не запоминая код программы в файл. Пакетный же режим исполняет код, написанный в файле.

7. Почему язык программирования Python называется языком динамической типизации?

В языке программирования python переменная связывается с типом в момент присваивания значения, а не в момент объявления переменной.

8. Какие существуют основные типы в языке программирования Python?

1. None (неопределенное значение переменной)

2. Логические переменные (Boolean Type)

3. Числа (Numeric Type)

- `int` – целое число
- `float` – число с плавающей точкой
- `complex` – комплексное число

4. Списки (Sequence Type)

- list – список
- tuple – кортеж
- range – диапазон

5. Строки (Text Sequence Type)

- str

6. Бинарные списки (Binary Sequence Types)

- bytes – байты
- bytearray – массивы байт
- memoryview – специальные объекты для доступа к внутренним данным объекта через protocol buffer

7. Множества (Set Types)

- set – множество
- frozenset – неизменяемое множество

8. Словари (Mapping Types)

- dict – словарь

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

Целочисленное значение в рамках языка Python по сути своей является объектом. Объект, в данном случае – это абстракция для представления данных, данные – это числа, списки, строки и т.п. При этом, под данными следует понимать, как непосредственно сами объекты, так и отношения между ними. 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, на уровне интерпретатора, происходит следующее:

Создается целочисленный объект «5» (можно представить, что в этот момент создается ячейка И 5 кладется в эту ячейку);

Данный объект имеет некоторый идентификатор, значение: 5, и тип: целое число;

Посредством оператора “=” создается ссылка между переменной b и целочисленным объектом 5 (переменная b ссылается на объект 5).

10.Как получить список ключевых слов в Python?

```
import keyword  
  
print(keyword.kwlist)
```

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

Функция id() возвращает целое число которое для данного объекта является уникальным и постоянным.

Функция type() возвращает тип конкретного 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 (frozen set).

К изменяемым (mutable) типам относятся: списки (list), множества (set), словари (dict). Как уже было сказано ранее, при создании переменной, вначале создается объект, который имеет уникальный

идентификатор, тип и значение, после этого переменная может ссылаться на созданный объект.

13. Чем отличаются операции деления и целочисленного деления?

При целочисленном делении отбрасывается остаток.

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

В python есть возможность создавать переменную с комплексным значением, их можно складывать, вычитать, умножать, делить и возводить в степень. Так же можно извлечь действительную и мнимую части, при помощи метода `conjugate()` можно получить комплексно-сопряженное число.

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

В стандартную поставку Python входит библиотека `math`, в которой содержится большое количество часто используемых математических функций. Таких как вычисление тригонометрических функций, работа с логарифмами, вычисление факториала и экспоненты и т.д.

Отличие модуля `cmath` от `math` заключается в том, что модуль `cmath` работает с комплексными числами, а модуль `math` работает с математическими операциями.

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

При помощи параметра `sep` можно указать отличный от пробела разделитель строк.

Параметр `end` позволяет указывать, что делать, после вывода строки. По умолчанию происходит переход на новую строку. Однако это действие можно отменить, указав любой другой символ или строку

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.

Метод `format()` предназначен для того что бы сделать строку, подставив в неё некоторые данные, полученные в процессе выполнения программы (пользовательский ввод, данные из файлов и т. д.).

Подстановку данных можно сделать с помощью форматирования строк. Форматирование можно сделать не только использованием метода `format()` но и конкатенацией, %-форматированием, Template-строками и f-строками. f-строки. Способ похожий на `format()`, но более гибкий и читабельный.

Они поддерживают расширенное форматирование чисел, могут форматировать дату без метода `strftime()`, поддерживают базовые арифметические операции прямо в строках, позволяют обращаться к значениям списков по индексам, к элементам словарей по ключу, вызывать функции и методы объектов.

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

Целочисленный ввод: `int(input())`

Вещественный ввод: `float(input())`