

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:

Борсуков Владислав Олегович
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:

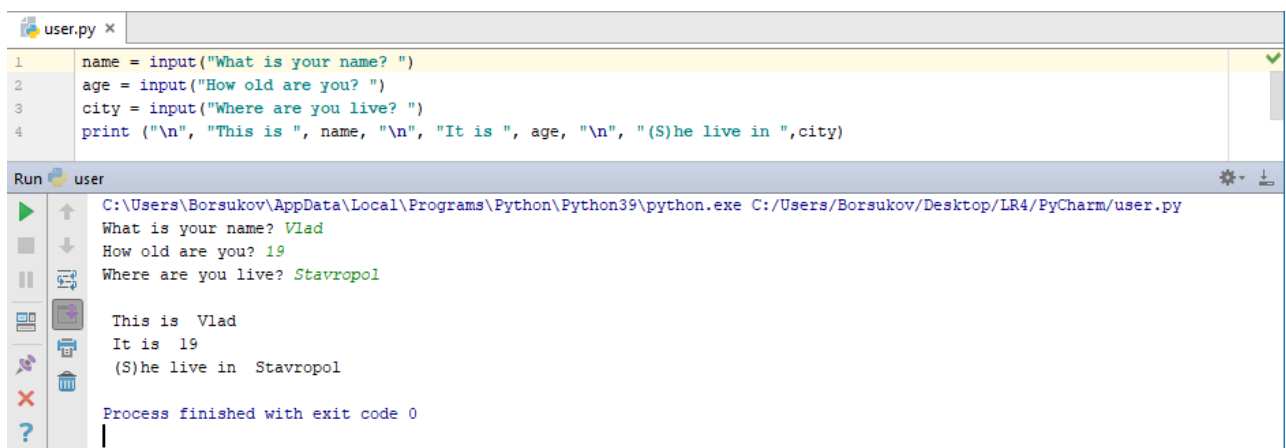
Его имя (например, "What is your name?")

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 window titled 'user.py'. The code in the editor is as follows:

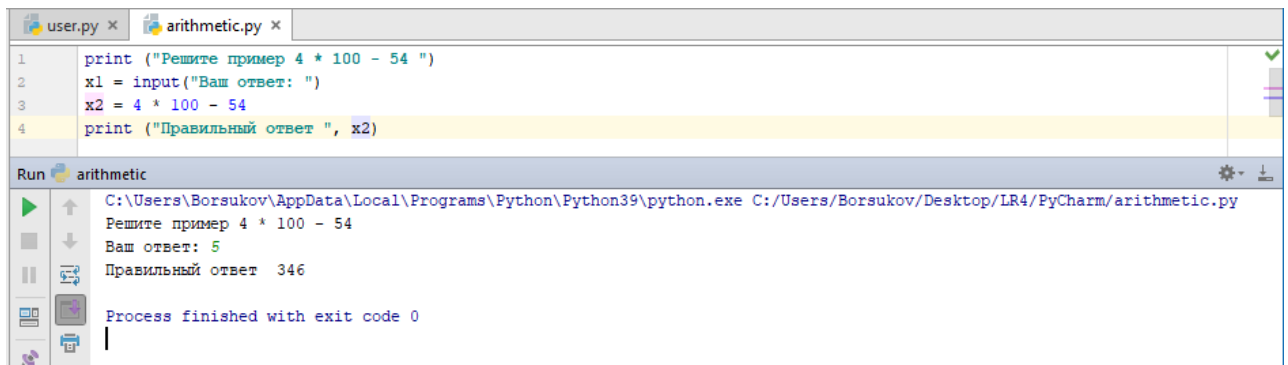
```
1 name = input("What is your name? ")  
2 age = input("How old are you? ")  
3 city = input("Where are you live? ")  
4 print ("\n", "This is ", name, "\n", "It is ", age, "\n", "(S)he live in ", city)
```

Below the editor, the 'Run' console shows the execution of the program. The user input is 'Vlad', '19', and 'Stavropol'. The output is:

```
This is Vlad  
It is 19  
(S)he live in Stavropol  
  
Process finished with exit code 0
```

Рисунок 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.



```
1 print ("Решите пример 4 * 100 - 54 ")
2 x1 = input("Ваш ответ: ")
3 x2 = 4 * 100 - 54
4 print ("Правильный ответ ", x2)
```

Run arithmetic

C:\Users\Borsukov\AppData\Local\Programs\Python\Python39\python.exe C:/Users/Borsukov/Desktop/LR4/PyCharm/arithmetic.py

Решите пример 4 * 100 - 54

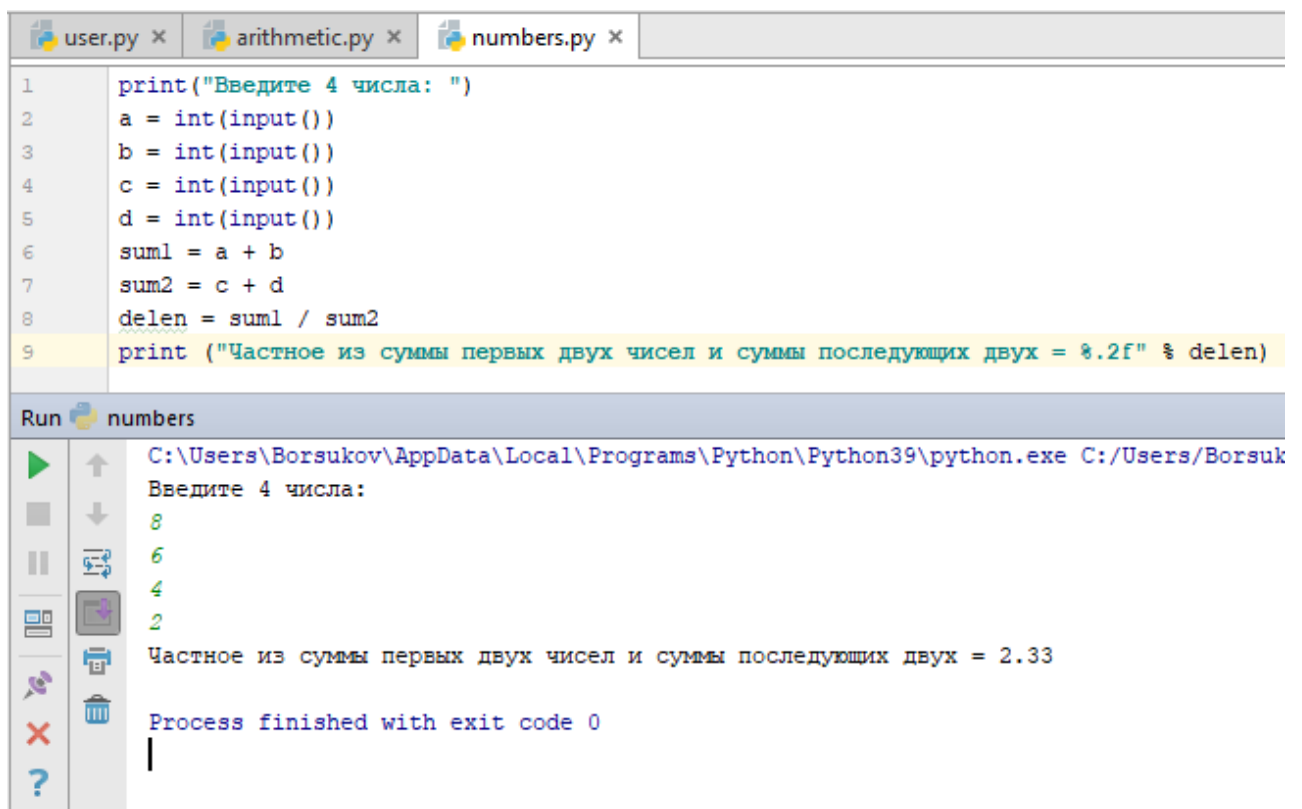
Ваш ответ: 5

Правильный ответ 346

Process finished with exit code 0

Рисунок 2 – 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.



```
1 print("Введите 4 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 delen = sum1 / sum2
9 print ("Частное из суммы первых двух чисел и суммы последующих двух = %.2f" % delen)
```

Run numbers

C:\Users\Borsukov\AppData\Local\Programs\Python\Python39\python.exe C:/Users/Borsukov/Desktop/LR4/PyCharm/numbers.py

Введите 4 числа:

8

6

4

2

Частное из суммы первых двух чисел и суммы последующих двух = 2.33

Process finished with exit code 0

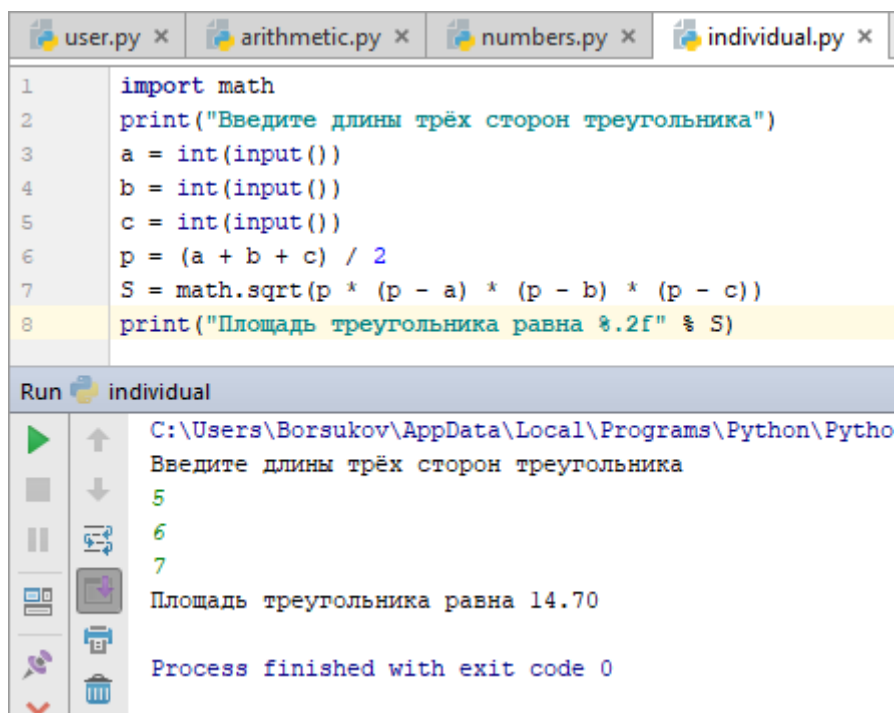
Рисунок 3 – Результат работы программы numbers.py

4. Напишите программу (файл individual.py) для решения индивидуального задания. Вариант 3.

3. Даны стороны треугольника a , b и c найти площадь треугольника по формуле Герона

$$S = \sqrt{p(p-a)(p-b)(p-c)},$$

где $p = (a + b + c)/2$.



```
1 import math
2 print("Введите длины трёх сторон треугольника")
3 a = int(input())
4 b = int(input())
5 c = int(input())
6 p = (a + b + c) / 2
7 S = math.sqrt(p * (p - a) * (p - b) * (p - c))
8 print("Площадь треугольника равна %.2f" % S)
```

Run individual

C:\Users\Borsukov\AppData\Local\Programs\Python\Python39\python.exe C:\Users\Borsukov\AppData\Local\Programs\Python\Python39\python.exe C:\Users\Borsukov\AppData\Local\Programs\Python\Python39\python.exe C:\Users\Borsukov\AppData\Local\Programs\Python\Python39\python.exe

Введите длины трёх сторон треугольника

5

6

7

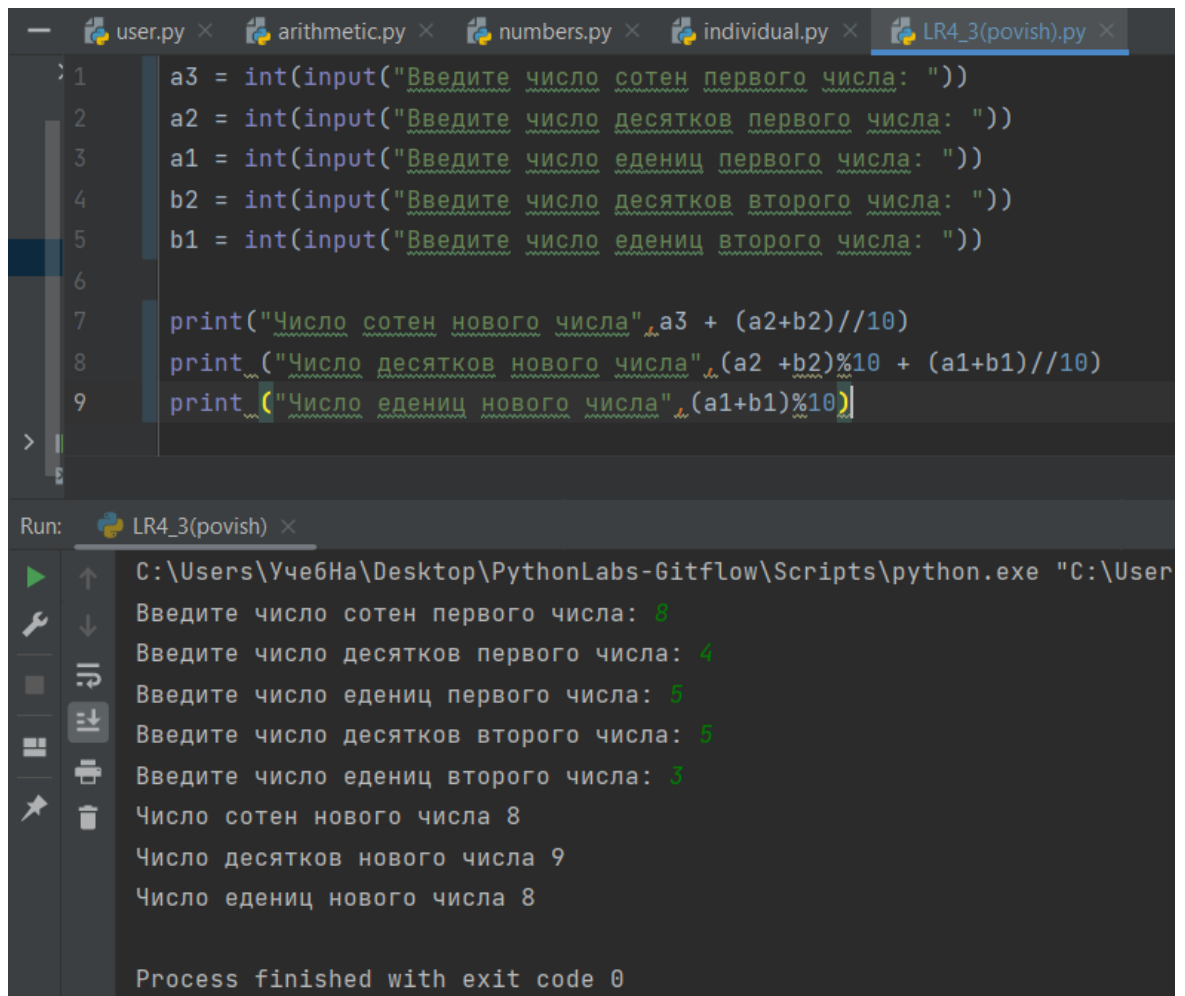
Площадь треугольника равна 14.70

Process finished with exit code 0

Рисунок 4 – Результат работы программы individual.py

5. Выполнение задания повышенной сложности. Вариант 3:

3. Даны цифры двух десятичных целых чисел: трехзначного $a_3 a_2 a_1$ и двузначного $b_2 b_1$, где a_1 и b_1 – число единиц, a_2 и b_2 – число десятков, a_3 – число сотен. Получить цифры числа, равного сумме заданных чисел (известно, что это число трехзначное). Числа-слагаемые и число-результат не определять; условный оператор не использовать.



The image shows a Python IDE with several tabs open: user.py, arithmetic.py, numbers.py, individual.py, and LR4_3(povish).py. The active tab is LR4_3(povish).py, which contains the following Python code:

```
1 a3 = int(input("Введите число сотен первого числа: "))
2 a2 = int(input("Введите число десятков первого числа: "))
3 a1 = int(input("Введите число единиц первого числа: "))
4 b2 = int(input("Введите число десятков второго числа: "))
5 b1 = int(input("Введите число единиц второго числа: "))
6
7 print("Число сотен нового числа", a3 + (a2+b2)//10)
8 print("Число десятков нового числа", (a2 +b2)%10 + (a1+b1)//10)
9 print("Число единиц нового числа", (a1+b1)%10)
```

Below the code editor, the 'Run' panel shows the execution of the program. The command executed is: `C:\Users\УчебHa\Desktop\PythonLabs-Gitflow\Scripts\python.exe "C:\User`. The output of the program is as follows:

```
Введите число сотен первого числа: 8
Введите число десятков первого числа: 4
Введите число единиц первого числа: 5
Введите число десятков второго числа: 5
Введите число единиц второго числа: 3
Число сотен нового числа 8
Число десятков нового числа 9
Число единиц нового числа 8
Process finished with exit code 0
```

Рисунок 5 – Результат работы программы special.py

Контрольные вопросы:

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

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

Linux: а) собрать Python из исходников; б) 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?

Запуск → 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)
 1. int – целое число
 2. float – число с плавающей точкой
 3. complex – комплексное число
4. Списки (Sequence Type)
 1. list – список
 2. tuple – кортеж
 3. range – диапазон
5. Строки (Text Sequence Type)
 1. str
6. Бинарные списки (Binary Sequence Types)
 1. bytes – байты
 2. bytearray – массивы байт
 3. memoryview – специальные объекты для доступа к внутренним данным объекта через protocol buffer
7. Множества (Set Types)
 1. set – множество
 2. frozenset – неизменяемое множество
8. Словари (Mapping Types)

1. 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 именных 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())`