

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 ПРАКТИЧЕСКОЙ РАБОТЕ №2.1
дисциплины «Программирование на Python»
Вариант 2

Выполнила:
Беседина Инга Олеговна
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника», очная
форма обучения

(подпись)

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

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Тема: Основы языка Python

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

Ход работы

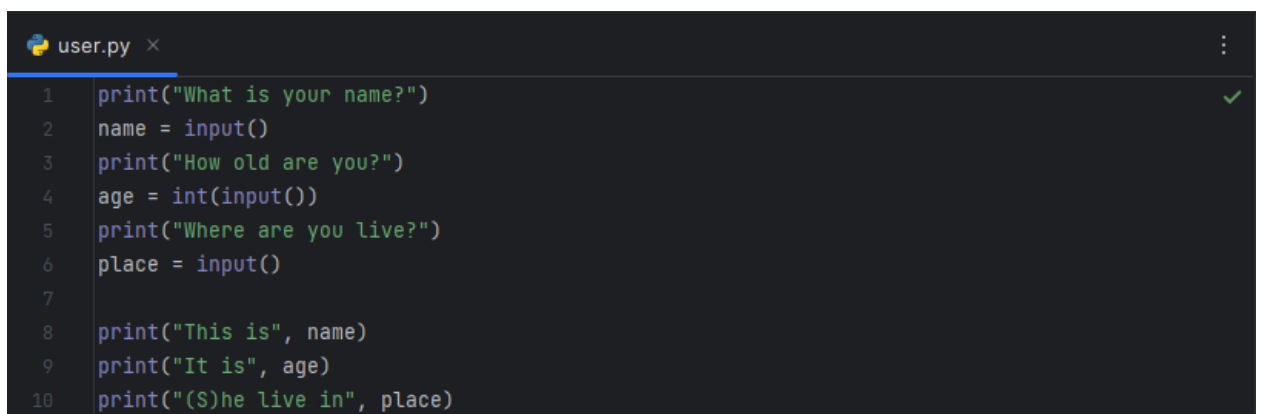
8. Напишите программу (файл *user.py*), которая запрашивала бы у пользователя:

- его имя (например, "What is your name?")
- возраст ("How old are you?")
- место жительства ("Where are you live?")

После этого выводила бы три строки:

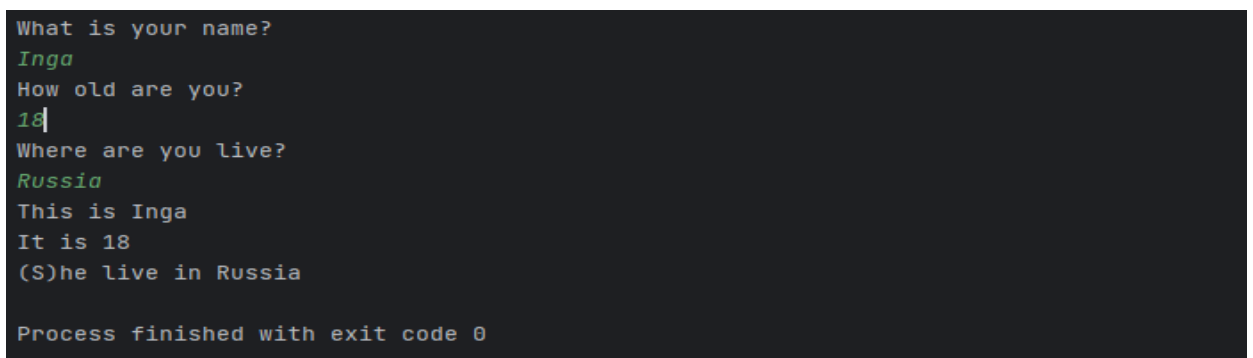
```
"This is `имя`"  
"It is `возраст`"  
"(S)he live in `место_жительства`"
```

Рисунок 1. Задание 1



```
user.py x  
1 print("What is your name?")  
2 name = input()  
3 print("How old are you?")  
4 age = int(input())  
5 print("Where are you live?")  
6 place = input()  
7  
8 print("This is", name)  
9 print("It is", age)  
10 print("(S)he live in", place)
```

Рисунок 2. Код программы



```
What is your name?  
Inga  
How old are you?  
18  
Where are you live?  
Russia  
This is Inga  
It is 18  
(S)he live in Russia  
  
Process finished with exit code 0
```

Рисунок 3. Результат работы программы

9. 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.

Рисунок 4. Задание 2

```
arithmetic.py ×
1 print("Решите пример: 4 * 100 - 54")
2 ans = input()
3 print("Ваш ответ:", ans)
4 print("Правильный ответ:", 4 * 100 - 54)
5
```

Рисунок 5. Код программы

```
Решите пример: 4 * 100 - 54
346
Ваш ответ: 346
Правильный ответ: 346

Process finished with exit code 0
```

Рисунок 6. Результат работы программы

10. 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.

Рисунок 7. Задание 3

```
numbers.py ×
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 print("%.2f" % (sum1/sum2))
9
```

Рисунок 8. Код программы

```
Введите 4 числа
8
15
4
21
0.92
```

Рисунок 9. Результат работы программы

2. Даны стороны прямоугольника. Найти его периметр и длину диагонали.

Рисунок 10. Индивидуальное задание

```

individual.py x
1 import math
2
3 print("Введите стороны прямоугольника:")
4 a = int(input())
5 b = int(input())
6 print("Периметр =", 2 * (a + b))
7 print("Длина диагонали =", '%.2f' % math.sqrt(a*a+b*b))

```

Рисунок 11. Код программы

```

Введите стороны прямоугольника:
7
4
Периметр = 22
Длина диагонали = 8.06

Process finished with exit code 0

```

Рисунок 12. Результат работы программы

2. Даны цифры двух двузначных чисел, записываемых в виде a_2a_1 и b_2b_1 , где a_1 и b_1 – число единиц, a_2 и b_2 – число десятков. Получить цифры числа, равного сумме заданных чисел (известно, что это число двузначное). Слагаемое – двузначное число и число-результат не определять; условный оператор не использовать.

Рисунок 13. Задание повышенной сложности

```

individual2.py x
1 a1 = 5
2 b1 = 6
3 a2 = 2
4 b2 = 4
5 u = a1 + b1
6 t = a2 + b2
7 t += u // 10
8 u %= 10
9 print(t, u)
10

```

Рисунок 14. Код программы

```

7 1

```

Рисунок 15. Результат работы программы

Контрольные вопросы

1. Основные этапы установки Python: скачать дистрибутив; запустить скачанный установочный файл; выбрать способ установки; выбрать место установки. При установке Python в Linux 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. Пакет Anaconda включает в себя интерпретатор языка Python (есть версии 2 и 3), набор наиболее часто используемых библиотек и удобную среду разработки и исполнения, запускаемую в браузере.

3. Для выполнения проверки работоспособности Anaconda необходимо вначале запустить командный процессор с поддержкой виртуальных окружений Anaconda. В Windows это можно сделать, выбрав следующий пункт главного меню системы Пуск Anaconda3 (64-bit) Anaconda Prompt. В появившейся командной строке необходимо ввести `jupyter notebook`, в результате чего отобразится процесс загрузки веб-среды Jupyter Notebook, после чего запустится веб-сервер и среда разработки в браузере.

4. Указать используемый интерпретатор языка Python в IDE PyCharm можно во время создания нового проекта.

5. В IDE PyCharm запуск программы осуществляется нажатием кнопки «Run»

6. Интерактивный режим работы Python предназначен для выполнения команд в интерактивной оболочке, где результаты вычислений выводятся непосредственно на экран. Этот режим позволяет быстро проверять отдельные участки кода и экспериментировать с ними. Пакетный режим работы Python предназначен для выполнения программ, которые сохранены в файле. В этом режиме Python выполняет все команды из файла и выводит результаты на экран или сохраняет их в файл. Пакетный режим позволяет автоматизировать выполнение программ и использовать Python для создания скриптов и приложений.

7. Python называется языком с динамической типизацией, так как тип переменной определяется непосредственно при выполнении программы

8. Основные типы данных: неопределенное значение переменной: none; логические переменные: boolean; числа: int, float, complex; списки: list, tuple, range; строки: str; бинарные списки: bytes, bytearray, memoryview; множества: set, frozenset; Словари: dict

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. Целочисленное значение 5 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.

10. Список ключевых слов можно получить непосредственно в программе, для этого нужно подключить модуль keyword и воспользоваться командой keyword.kwlist.

11. С помощью функции type() можно определить тип переменной, с помощью функции id() идентификатор объекта

12. Неизменяемость типа данных означает, что созданный объект больше не изменяется. 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). Если тип данных изменяемый, то можно менять значение объекта. К изменяемым (mutable) типам относятся: списки (list), множества (set), словари (dict)

13. Результатом целочисленного деления всегда является целое число, результатом обычного деления может быть вещественное число.

14. Функция `complex(a, b)`, в которую, в качестве первого аргумента, передается действительная часть, в качестве второго – мнимая. Либо записать число в виде $a + bj$. Комплексные числа можно складывать, вычитать, умножать, делить и возводить в степень. У комплексного числа можно извлечь действительную и мнимую части. Для получения комплексносопряженного числа необходимо использовать метод `conjugate()`.

15. В стандартную поставку Python входит библиотека `math`, в которой содержится большое количество часто используемых математических функций таких как `math.ceil(x)`, `math.fabs(x)`, `math.factorial(x)`, `math.floor(x)`, `math.exp(x)`, `math.log2(x)`, `math.pow(x, y)`, `math.sqrt(x)` и т.д.

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

17. Форматирование может выполняться в так называемом старом стиле или с помощью строкового метода `format`. В старом стиле вместо трех комбинаций символов `%s`, `%d`, `%f` подставляются значения переменных `upil`, `old`, `grade`. Буквы `s`, `d`, `f` обозначают типы данных – строку, целое число, вещественное число.

18. Ввод с консоли значения целочисленной и вещественной переменной в языке Python осуществляется с помощью функции `int(input())` и `float(input())`. Сначала выполняется функция `input()`. Она возвращает строку, которую функция `int()` или `float()` сразу преобразует в число. Только после этого происходит присваивание переменной, то есть она сразу получает численное значение.

Вывод: в ходе выполнения практической работы было проведено исследование процесса установки и базовых возможностей языка Python версии 3.x.

