

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 №4
дисциплины «Программирование на Python»

Выполнила:
Кубанова Ксения 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.

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Тема: основы языка Python

Цель: исследование базовых возможностей языка Python версии 3.x.

Порядок выполнения работы

1. User.py

Задание: написать программу, которая запрашивала бы у пользователя:

- Имя
- Возраст
- Место жительства

А после выдавало полное предложение с данными пользователя.

Решение:

```
1 name=input("What is your name? ")
2 old=input("How old are you? ")
3 city=input("Where are you live? ")
4 print("This is {0}. It is {1}. (S)he live in {2}.".format(name, old, city))
```

Рисунок 1 – задание 1

```
What is your name? Ksenya
How old are you? 19
Where are you live? Stavropol
This is Ksenya. It is 19. (S)he live in Stavropol.
```

Рисунок 2 – вывод задания 1

2. Arithmetic.py

Задание: написать программу, которая предлагала бы пользователю решить пример $4*100-54$, а после выводила бы правильный результат и результат пользователя.

Решение:

```
1 a=input("4*100-54= ")
2 print("Your answer: {0}, Correct answer: 346".format(a))
```

Рисунок 3 – задание 2

```
4*100-54= 66
Your answer: 66, Correct answer: 346
```

Рисунок 4 – вывод задания 2

3. Individual.py

Задание: даны координаты на плоскости двух точек. Найти расстояние между этими точками.

Решение:

```
1 from cmath import sqrt
2
3 print("finding the distance between points")
4 xa=int(input("Xa= "))
5 ya=int(input("Ya= "))
6 xb=int(input("Xb= "))
7 yb=int(input("Yb= "))
8 AB=sqrt((xb-xa)**2+(yb-ya)**2)
9 print("First point A({0},{1}) and B({2},{3}), distance AB={4}".format(xa,ya,xb,yb,f"{AB:.3f}"))
```

Рисунок 5 – индивидуальное задание

```
finding the distance between points
Xa= 3
Ya= 5
Xb= 8
Yb= 9
First point A(3,5) and B(8,9), distance AB=6.403+0.000j
```

Рисунок 6 – вывод индивидуального задания

4. Hardmode.py

Задание: с начала суток часовая стрелка повернулась на y градусов ($0 \leq y \leq 2\pi$, y – вещественное число). Определить число полных часов и число полных минут, прошедших с начала суток.

Решение:

Для того, чтобы определить количество прошедших часов, достаточно общее количество градусов поделить на 30 (если представить единичную окружность, то 30 градусов – это 1 час). Соответственно, для определения минут, количество полученных часов нужно умножить на 60.

```
1 y=float(input("the arrow turned to = "))
2 x=y/30
3 print("Hours = {0}".format(f"{x:.0f}"))
4 z=x*60
5 print("Minuts = {0}".format(f"{z:.0f}"))
```

Рисунок 7 – задание повышенной сложности

```
the arrow turned to = 720
Hours = 24
Minuts = 1440
```

Рисунок 8 – вывод задания повышенной сложности

Контрольные вопросы

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

Windows:

С официального сайта скачать установщик Python для Windows, соответствующий версии операционной системы, запустить загруженный установщик Python, выбрать опцию "Add Python to PATH", нажать кнопку "Install Now" и следовать инструкциям установщика.

Linux:

Открыть терминал на системе Linux, установить пакеты, необходимые для установки Python, с помощью менеджера пакетов дистрибутива.

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

- **Установка и управление пакетами:** Anaconda поставляется с инструментом управления пакетами conda, который позволяет легко устанавливать, обновлять и удалять пакеты Python и их зависимости. Это упрощает процесс установки и управления сторонними пакетами.

- **Управление версиями Python:** Anaconda позволяет легко управлять версиями Python и создавать виртуальные среды, которые изолируют зависимости и позволяют работать с разными версиями Python и пакетами в одной системе.

3. Как осуществить проверку работоспособности пакета Anaconda?

Открыть командную строку или терминал и ввести команду `python` или `python3`, в зависимости от версии Python, установленной с помощью Anaconda. Если интерактивная оболочка Python успешно запускается и видно приглашение `>>>`, это означает, что Anaconda установлена и работает. В Linux нужно открыть окно терминала и в нем ввести `python3` (или `python`).

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

- 1) Открыть проект в PyCharm.
- 2) В верхней панели меню выбрать "File" (Файл), затем "Settings" (Настройки).
- 3) В окне настроек выбрать "Project" (Проект) в левой панели и выбрать "Python Interpreter" (Интерпретатор Python).
- 4) В правой части окна настроек нажать на выпадающий список "Python Interpreter" (Интерпретатор Python) и выбрать нужный интерпретатор Python из списка доступных интерпретаторов.

5. Как осуществить запуск программы с помощью IDE PyCharm?

- 5) Открыть проект в PyCharm.
- 6) Убедиться, что файл с кодом программы открыт в редакторе PyCharm.
- 7) Нажать комбинацию клавиш **Shift+Alt+F10** или выбрать пункт меню "Run" (Запуск) в верхней панели.
- 8) Если есть несколько конфигураций запуска, выбрать нужную конфигурацию из списка.
- 9) Если конфигурация запуска не настроена, нажать на пункт меню "Edit Configurations" (Изменить конфигурации) и настроить новую конфигурацию запуска.
- 10) После выбора конфигурации запуска нажать кнопку "Run" (Запуск) или использовать соответствующую комбинацию клавиш, чтобы запустить программу.

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

Интерактивный режим – работа в терминале напрямую. При пакетном режиме файл запускается (или создается) через терминал.

7. Почему язык программирования Python называется языком динамической типизации?

Python называют языком динамической типизации потому что тип переменной определяется непосредственно при выполнении программы.

8. Какие существуют основные типы в языке программирования Python?

None (неопределённое значение переменной), логические переменные (Boolean type), числа (int, float, complex), списки (list, tuple, range), строки (str), бинарные строки (bytes, bytearray, memoryview), множества (set, frozenset), словари (dict).

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

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("Python keywords: ", keyword.kwlist)
```

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

Для того, чтобы посмотреть на объект с каким идентификатором ссылается данная переменная, можно использовать функцию id(). Тип переменной можно определить с помощью функции type().

12. Что такое изменяемые и неизменяемые типы в Python.

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

13. Чем отличаются операции деления и целочисленного деления?

При делении в вывод выходит полученное число полностью, а при целочисленном – только его целая часть.

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

Для создания комплексного числа можно использовать функцию `complex(a, b)`, в которую, в качестве первого аргумента, передается действительная часть, в качестве второго – мнимая.

Либо записать число в виде $a + bj$.

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. Основными из них являются: `math.ceil(x)`, `math.fabs(x)`, `math.factorial(x)`, `math.floor(x)`, `math.exp(x)` и другие. `Cmath` расширяет функции `math`, и в него входят такие основные функции, как: `cmath.sqrt(z)`, `cmath.exp(z)`, `cmath.log(z, base)`, `cmath.sin(z)`, `cmath.cos(z)`, `cmath.tan(z)`.

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

Используются для настройки форматирования вывода. `sep` определяет разделитель между значениями, которые нужно вывести. По умолчанию `sep` равен пробелу. Параметр `end` определяет символ, который будет добавлен в конце вывода. По умолчанию `end` равен символу новой строки (`\n`).

17. Каково назначение метода format()? Какие еще существуют средства для форматирования строк в Python?

Метод format() в Python используется для форматирования строк. Он позволяет вставлять значения переменных в определенные места в строке, указанные с помощью фигурных скобок {}. format() также позволяет управлять форматированием значений, таких как числа, даты и времена, с помощью специальных спецификаторов формата.

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

Для ввода целочисленной и вещественной переменной с консоли в языке Python можно использовать функцию input(), а затем преобразовать введенное значение в нужный тип данных.

Вывод: в ходе выполнения лабораторной работы, были исследованы базовые возможности языка Python версии 3.x.