

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 «Основы программной инженерии»

Выполнил:
Плугатырев Владислав Алексеевич
1 курс, группа ПИЖ-б-о-22-1,
09.03.04 «Программная инженерия»,
направленность (профиль) «Разработка и
сопровождение программного
обеспечения», очная форма обучения

(подпись)

Доцент кафедры инфокоммуникаций
Воронкин Роман Александрович

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Тема: основы языка Python.

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

Порядок выполнения работы

1. Создал общедоступный репозиторий с лицензией MIT.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk ().*

Owner * **Repository name ***

BigLofanbl4 / lab4

✔ lab4 is available.

Great repository names are short and memorable. Need inspiration? How about [scaling-giggle](#) ?

Description (optional)

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

☒ **Add a README file**
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: **Python**

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: **MIT License**

Рисунок 1 – Создание репозитория Git

2. Выполнил клонирование созданного репозитория.

```
C:\Users\MSI\Desktop\Основы программной инженерии\4>git clone https://github.com/BigLofanbl4/lab4.git
Cloning into 'lab4'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.
```

Рисунок 2 – Клонирование репозитория

3. Дополнил файл «.gitignore» необходимыми правилами для работы с IDE PyCharm.

```
# option (not recommended)
.idea/
```

Рисунок 3 – Файл «.gitignore»

4. Организовал репозиторий в соответствии с моделью ветвления «git-flow».

```
C:\Users\MSI\Desktop\Основы программной инженерии\4\lab4>git branch
* develop
main
```

Рисунок 4 – Ветки репозитория

5. Написал программу «user.py».

```
#!/usr/bin/env python

name = input("What is your name?\n")
age = input("How old are you?\n")
home = input("Where do you live?\n")

print("This is " + name)
print("It is " + age)
print("S(he) live in " + home)
```

Рисунок 5.1 – Код программы

```
What is your name?  
Lofanbl4  
How old are you?  
18  
Where do you live?  
Stavropol  
This is Lofanbl4  
It is 18  
S(he) live in Stavropol
```

Рисунок 5.2 – Вывод программы

6. Написал программу «arithmetic.py». Написать программу, которая предлагала бы пользователю решить пример $4 * 100 - 54$. Потом выводила бы на экран правильный ответ и ответ пользователя.

```
#!/usr/bin/env python  
  
print("Solve this problem: 4 * 100 - 54")  
ans = input("Your answer: ")  
print(f"Right answer: {4 * 100 - 54}")
```

Рисунок 6.1 – Код программы

```
Solve this problem: 4 * 100 - 54  
Your answer: 346  
Right answer: 346
```

Рисунок 6.2 – Вывод программы

7. Написал программу «numbers.py». Запросить у пользователя четыре числа. Отдельно сложить первые два и отдельно вторые два.

Разделить первую сумму на вторую. Вывести результат на экран так, чтобы ответ содержал две цифры после запятой.

```
#!/usr/bin/env python  
  
a = int(input("Number 1: "))  
b = int(input("Number 2: "))  
c = int(input("Number 3: "))  
d = int(input("Number 4: "))  
  
print(f"Result: {round((a + b) / (c + d), 2)}")
```

Рисунок 7.1 – Код программы

```
Number 1: 5347  
Number 2: 457  
Number 3: 43  
Number 4: 36  
Result: 73.47|
```

Рисунок 7.2 – Вывод программы

8. Написал программу «individual.py» (Вариант 14).

```
#!/usr/bin/env python|  
  
speed1 = float(input("Speed of first car (km/h): "))  
speed2 = float(input("Speed of second car (km/h): "))  
distance = float(input("Distance between cars (kms): "))  
  
print(f"Time of meeting: {round(distance / (speed1 + speed2), 2)}h")
```

Рисунок 8.1 – Код программы

```
Speed of first car (km/h): 44.67
Speed of second car (km/h): 32
Distance between cars (kms): 60.45
Time of meeting: 0.79h
```

Рисунок 8.2 – Вывод программы

9. Выполнил задачу повышенной сложности (вариант 6). . С начала суток часовая стрелка повернулась y на градусов ($0 \leq y < 360$, y – вещественное число). Определить число полных часов и число полных минут, прошедших с начала суток

```
#!/usr/bin/env python
import math

y = float(input("Enter angle: "))
hours = math.floor(y * 2 / 60)
minutes = math.floor(y * 2 - (hours * 60))

print(f"{hours} hours {minutes} minutes")
```

Рисунок 9.1 – Код программы

```
Enter angle: 135.69
4 hours 31 minutes
```

Рисунок 9.2 – Вывод программы

Ответы на контрольные вопросы

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

Установка Python на Windows:

- 1) Перейти на сайт официальный сайт Python по адресу <https://python.org> и скачать установочный файл.
- 2) Запустить скачанный установочный файл.
- 3) Выбрать способ установки (Install Now и Custom installation).
- 4) Отметить необходимые опции установки (доступно при выборе Customize installation).
- 5) Выбрать место установки (доступно при выборе Customize installation).

Установка Python на Linux:

Для установки из репозитория в Ubuntu необходимо воспользоваться командой «`sudo apt-get install python3`».

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

Anaconda включает в себя не только сам интерпретатор Python, но и набор наиболее используемых библиотек и удобную среду разработки и исполнения, запускаемую в браузере.

3. Как осуществить проверку работоспособности пакета Anaconda?

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 отобразиться процесс загрузки веб-среды Jupyter Notebook после чего запустится веб-сервер и среда разработки в браузере.

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

Перейти в настройки, в открывшемся окне выбрать раздел Project, выбрать Project Interpreter, в выпадающем меню указать путь до интерпретатора.

5. Как осуществить запуск программы с помощью IDE PyCharm?

Нажать на кнопку Run.

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

Интерактивный режим работы Python позволяет вам взаимодействовать с интерпретатором Python непосредственно вводя команды в интерактивной оболочке. Пакетный режим, или скриптовый режим, используется для выполнения программ, которые содержат набор инструкций Python, помещенных в файл. В этом режиме скрипт Python сохраняется в файле с расширением .py, а затем запускается интерпретатор Python, который последовательно выполняет инструкции в скрипте сверху вниз.

7. Почему язык программирования Python называется языком динамической типизации?

Язык программирования Python называется языком динамической типизации, потому что в нем типы переменных определяются автоматически во время выполнения программы, а не во время компиляции. Это означает, что можно присваивать переменным значения разных типов данных, и интерпретатор Python будет автоматически определять их типы во время выполнения программы.

8. Какие существуют основные типы в языке программирования Python?

- None (неопределенное значение переменной);
- Логические переменные (Boolean Type);
- Числа (Numeric Type): int – целое число, float – число с плавающей точкой, complex – комплексное число;
- Списки (Sequence Type): list – список, tuple – кортеж, range – диапазон;

- Строки (Text Sequence Type): str;
- Бинарные списки (Binary Sequence Types) bytes – байты, bytearray
- массивы байт, memoryview – специальные объекты для доступа к внутренним данным объекта через protocol buffer;
- Множества (Set Types) set – множество frozenset – неизменяемое множество;
- 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?

Для того, чтобы объявить и сразу инициализировать переменную необходимо написать её имя, потом поставить знак равенства и значение, с которым эта переменная будет создана. Например строка: «b = 5» объявляет переменную b и присваивает ей значение 5. 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 (об этом чуть позже). 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?

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

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

Функция `id()` возвращает уникальный идентификатор (адрес) объекта в памяти.

Функция `type()` возвращает тип объекта.

12. Что такое изменяемые и неизменяемые типы в Python?

Изменяемые типы позволяют изменять свои значения после создания, а неизменяемые типы не могут быть изменены после создания.

13. Чем отличаются операции деления и целочисленного деления?

Python операция деления (`/`) выполняет обычное деление чисел с плавающей точкой и возвращает результат в виде числа с плавающей точкой. Целочисленное деление (`//`), с другой стороны, выполняет деление и возвращает только целую часть результата, отбрасывая дробную 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`. Комплексные числа можно складывать, вычитать, умножать, делить и возводить в степень.

15. Каково назначение и основные функции библиотеки (модуля) `math`?

Модуль `math` языка Python предоставляет функции для выполнения математических операций. Он содержит большое количество математических функций, таких как тригонометрические функции (`sin`, `cos`, `tan`), функции для работы с логарифмами, экспонентами, квадратным корнем и другими.

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

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

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

Метод `format()` в языке Python используется для форматирования строк. Он позволяет вставлять значения в определенные места в строке и задавать формат вывода этих значений. Метод `format()` позволяет создавать более динамичные строки, в которых значения могут меняться в зависимости от контекста.

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

Для ввода значений целочисленной и вещественной переменной в языке Python можно использовать функции `input()` и `float()` или `int()`. Функция `input()` позволяет считывать введенные пользователем данные с консоли в виде строки. Чтобы преобразовать введенную строку в целочисленное значение, используется функция `int()`, а для преобразования вещественного значения - `float()`.