

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»

Выполнила:
Ламская Ксения Вячеславовна
2 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. Создала общедоступный репозиторий с лицензией MIT.

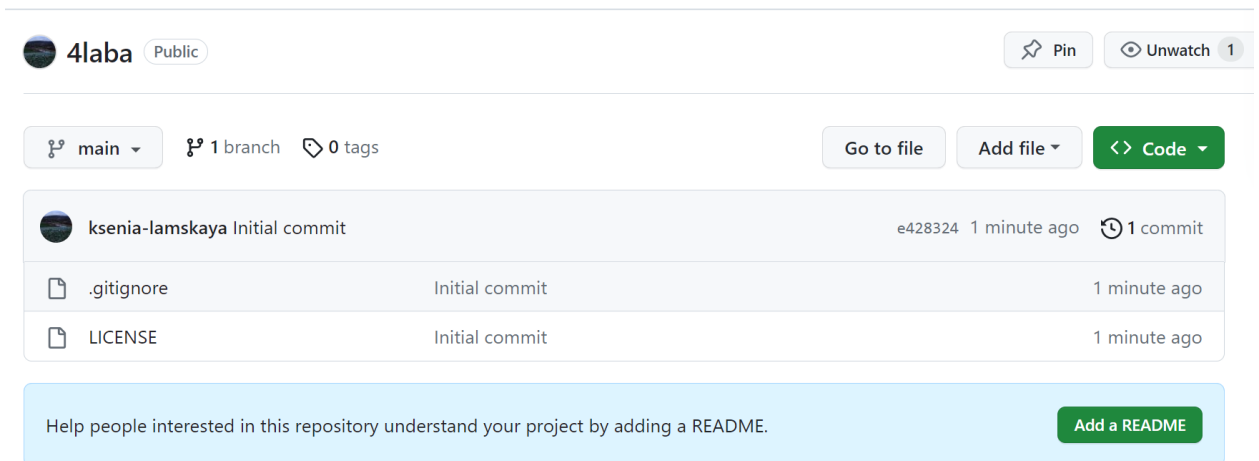


Рисунок 1 – Создание репозитория Git

2. Выполнила клонирование созданного репозитория.

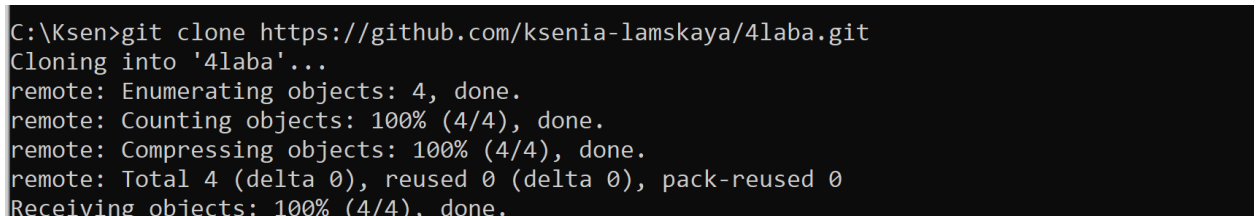


Рисунок 2 – Клонирование репозитория

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

```
.mypy_cache/  
.dmypy.json  
dmypy.json
```

Рисунок 3 – Файл «.gitignore»

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

```
C:\Ksen\4laba>git checkout develop
Switched to branch 'develop'

C:\Ksen\4laba>git branch
* develop
  main

C:\Ksen\4laba>
```

Рисунок 4 – Ветки репозитория

5. Написала программу «user.py».

```
C: > Ksen > 4laba > zadachi > user.py > ...
1  #!/usr/bin/env python
2
3  name = input("What is your name?\n")
4  age = input("How old are you\n")
5  place = input("Where do you live?\n")
6  print("This is", name)
7  print("It is", age)
8  print("S(he) live in", place)
9
10
```

Рисунок 5.1 – Код программы

```
PS C:\Ksen\2> & C:/Users/irbis/AppData/Local/Programs/Python/Python311/python.exe c:/Ksen/4laba/
zadachi/user.py
What is your name?
Ksenia
How old are you
19
Where do you live?
Stavropol
This is Ksenia
It is 19
S(he) live in Stavropol
```

Рисунок 5.2 – Вывод программы

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

```

C: > Ksen > 4laba > zadachi > arithmetic.py > ...
1  #!/usr/bin/env python
2
3  print("Solve this problem: (4*100 - 54)")
4  decis = input()
5  print(f"Right decision:{4*100 - 54}")
6  print("Your answer:", decis)
7

```

Рисунок 6.1 – Код программы

```

Solve this problem: (4*100 - 54)
346
Right decision:346
Your answer: 346

```

Рисунок 6.2 – Вывод программы

7. Написала 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.

```

C: > Ksen > 4laba > zadachi > numbers.py > ...
1  #!/usr/bin/env python
2
3  print("Enter four nambers")
4  a = int(input("Number1:"))
5  b = int(input("Number2:"))
6  c = int(input("Number3:"))
7  d = int(input("Number4:"))
8  print(f"Result:{round((a + b) / (c + d), 2)}")
9

```

Рисунок 7.1 – Код программы

```

Enter four nambers
Number1:1
Number2:12
Number3:8
Number4:1
Result:1.44

```

Рисунок 7.2 – Вывод программы

8. Написала программу «individual.py» (Вариант 9). Треугольник задан координатами своих вершин. Найти периметр и площадь треугольника.

```
1  #!/usr/bin/env python
2  import math
3
4  def distance(x1, y1, x2, y2):
5      return math.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
6
7  def calculate_perimeter(x1, y1, x2, y2, x3, y3):
8      side1 = distance(x1, y1, x2, y2)
9      side2 = distance(x1, y1, x3, y3)
10     side3 = distance(x2, y2, x3, y3)
11     return side1 + side2 + side3
12
13 def calculate_area(x1, y1, x2, y2, x3, y3):
14     side1 = distance(x1, y1, x2, y2)
15     side2 = distance(x1, y1, x3, y3)
16     side3 = distance(x2, y2, x3, y3)
17     poluperimeter = (side1 + side2 + side3) / 2
18     return math.sqrt(poluperimeter * (poluperimeter - side1) * (poluperimeter - side2) * (poluperimeter - side3))
19
20 x1, y1 = map(float, input("Введите координаты вершины А через пробел: ").split())
21 x2, y2 = map(float, input("Введите координаты вершины В через пробел: ").split())
22 x3, y3 = map(float, input("Введите координаты вершины С через пробел: ").split())
23
24 perimeter = calculate_perimeter(x1, y1, x2, y2, x3, y3)
25 area = calculate_area(x1, y1, x2, y2, x3, y3)
26
27 print("Периметр треугольника равен", perimeter)
28 print("Площадь треугольника равна", area)
```

Рисунок 8.1 – Код программы

```
Введите координаты вершины А через пробел: 2 5
Введите координаты вершины В через пробел: 5 7
Введите координаты вершины С через пробел: 6 7
Периметр треугольника равен 9.07768723046357
Площадь треугольника равна 1.0000000000000004
```

Рисунок 8.2 – Вывод программы

9. Выполнила задачу повышенной сложности (вариант 1). Даны цифры двух целых чисел: двузначного a_2a_1 и однозначного b_1 , где a_1 – число единиц, a_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.

```
1  a2, a1, b1 = int(input()), int(input()), int(input())
2  a = int(str(a2) + str(a1)) + b1
3  print(str(a)[0], str(a)[1])
4
```

Рисунок 9.1 – Код программы

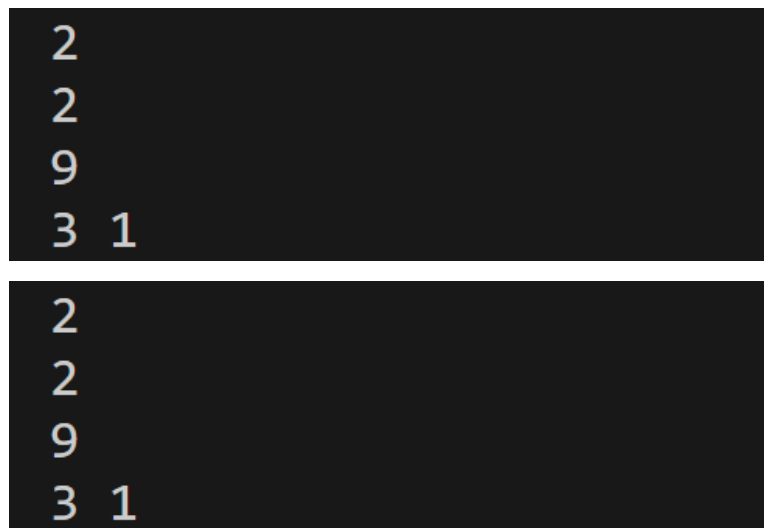


Рисунок 9.2 – Вывод программы

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 отобразится 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()`.

