

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций
Информационные технологии в телекоммуникациях

Отчет по лабораторной работе №2

«Основы языка Python»

Выполнил студент группы ИТС-б-о-21-1

Крамаренко Илья Витальевич

« » _____ 20__ г.

Подпись студента _____

Проверил: Доцент, к.т.н, доцент кафедры
инфокоммуникаций

Воронкин А. В.

Работа защищена с оценкой: _____

(подпись)

Ставрополь, 2022

Лабораторная работа 2.1 Основы языка Python

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

Порядок выполнения работы:

- 1) Создадим общедоступный репозиторий на GitHub (<https://github.com/IliyaKr/Second-work.git>)
- 2) Решим задачи с помощью языка программирования Python3. И отправим их на GitHub.

Задача 1.

7. 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?")

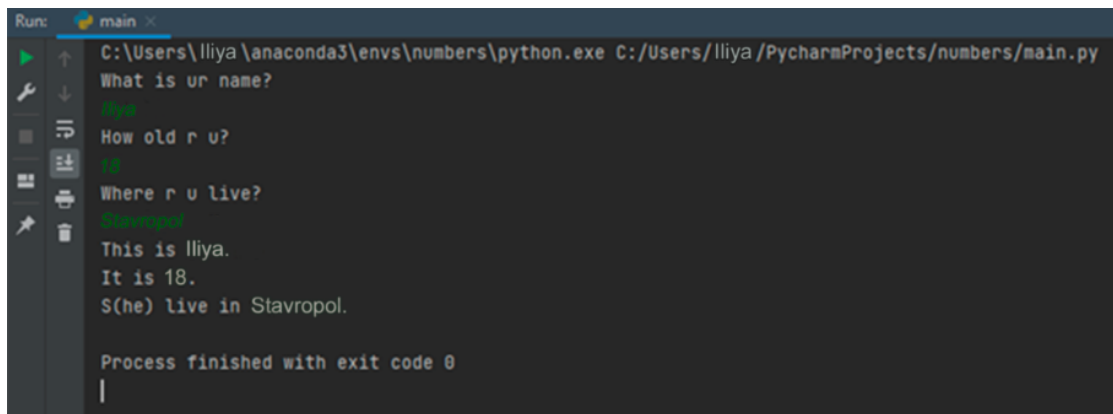
После этого выводила бы три строки:

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

Рис 1. Задача 1.

```
1 print("What is ur name?")  
2 nameUser = input()  
3 print("How old r u?")  
4 ageUser = input()  
5 print("Where r u live?")  
6 cityUser = input()  
7 print("This is {0}.".format(nameUser))  
8 print("It is {0}.".format(ageUser))  
9 print("S(he) live in {0}.".format(cityUser))  
10
```

Рис 2. Код для задачи 1.



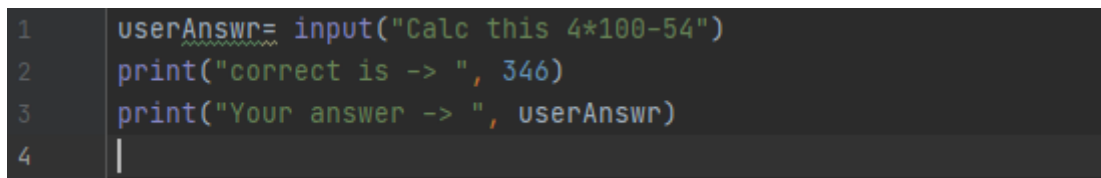
```
Run: main x
C:\Users\Iliya\anaconda3\envs\numbers\python.exe C:/Users/Iliya/PycharmProjects/numbers/main.py
What is ur name?
Iliya
How old r u?
18
Where r u live?
Stavropol
This is Iliya.
It is 18.
S(he) live in Stavropol.

Process finished with exit code 0
```

Рис 3. Работоспособность кода для первой задачи.

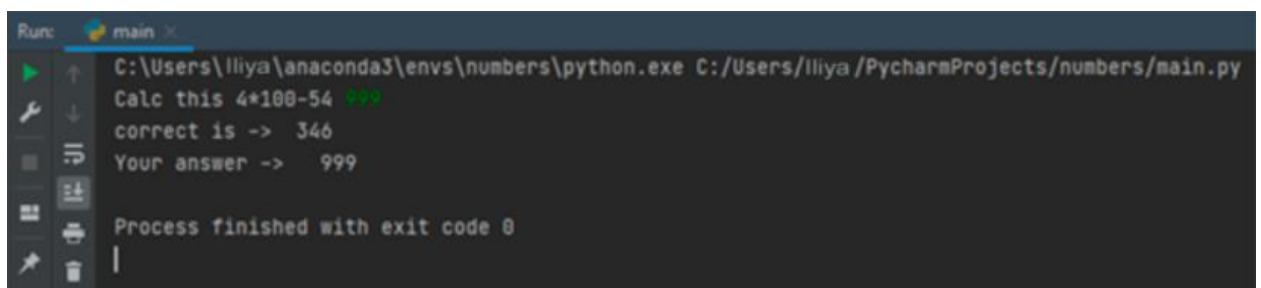
Задача 2.

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 userAnswr= input("Calc this 4*100-54")
2 print("correct is -> ", 346)
3 print("Your answer -> ", userAnswr)
4 |
```

Рис 4. Код для второй задачи.



```
Run: main x
C:\Users\Iliya\anaconda3\envs\numbers\python.exe C:/Users/Iliya/PycharmProjects/numbers/main.py
Calc this 4*100-54 999
correct is -> 346
Your answer -> 999

Process finished with exit code 0
```

Рис 4. Работоспособность кода для второй задачи.

Задача 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 summ1=a+b
7 summ2=c+d
8 x=summ1/summ2
9 print('%.2f' % x)
10 |
```

Рис 5. Код для третьей задачи.

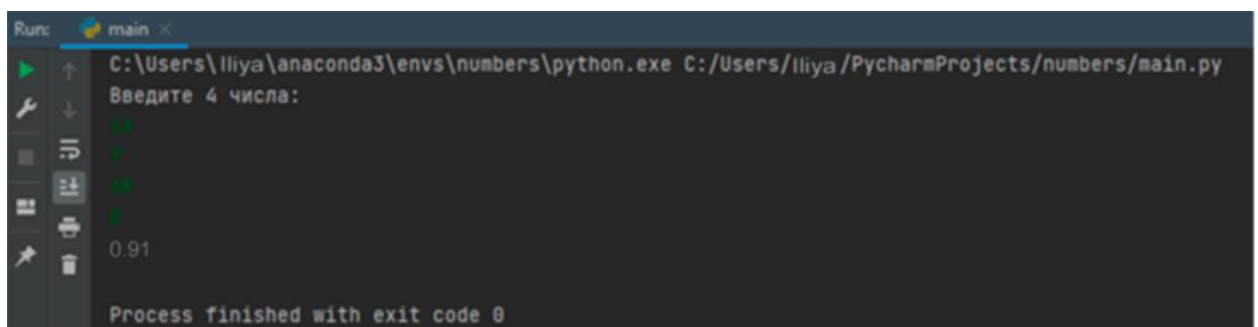


Рис 6. Работоспособность кода для третьей задачи.

Задача 4.

Напишите программу (файл individual.py) для решения индивидуального задания. Вариант индивидуального задания уточните у преподавателя.

Вариант 3.

7. Даны основания и высота равнобедренной трапеции. Найти периметр трапеции.

```

1  from math import sqrt
2
3  a = float(input("Длина большего основания:"))
4  b = float(input("Длина меньшего основания:"))
5  h = float(input("Высота:"))
6  print("P =", a + b + 2 * sqrt(sqrt(h) + sqrt(a - b) / 4))

```

Рис 7. Код для индивидуальной задачи.

```

Длина большего основания:15
Длина меньшего основания:10
Высота:9
P = 28.77307142491363

```

Рис 8. Работоспособность кода для четвертой задачи.

Задача повышенной сложности:

7. Часовая стрелка образует угол γ с лучом, проходящим через центр и через точку, соответствующую 12 часам на циферблате, $0 < \gamma \leq 2\pi$. Определить значение угла для минутной стрелки, а также количество полных часов и полных минут.

```

1  import datetime
2  from math import floor
3  now = datetime.datetime.now()
4  m = float(now.minute) + float(now.second/60)
5  ugol = m/5*30
6  print(str(now.hour) + " час " + str(now.minute) + " минут")
7  print([str(floor(ugol)) + " Угол"])

```

Рис 9. Код для задачи повышенной сложности(вариант 3).

```

22 час 4 минут
28 Угол

```

Рис10. Работоспособность кода для задачи повышенной сложности.

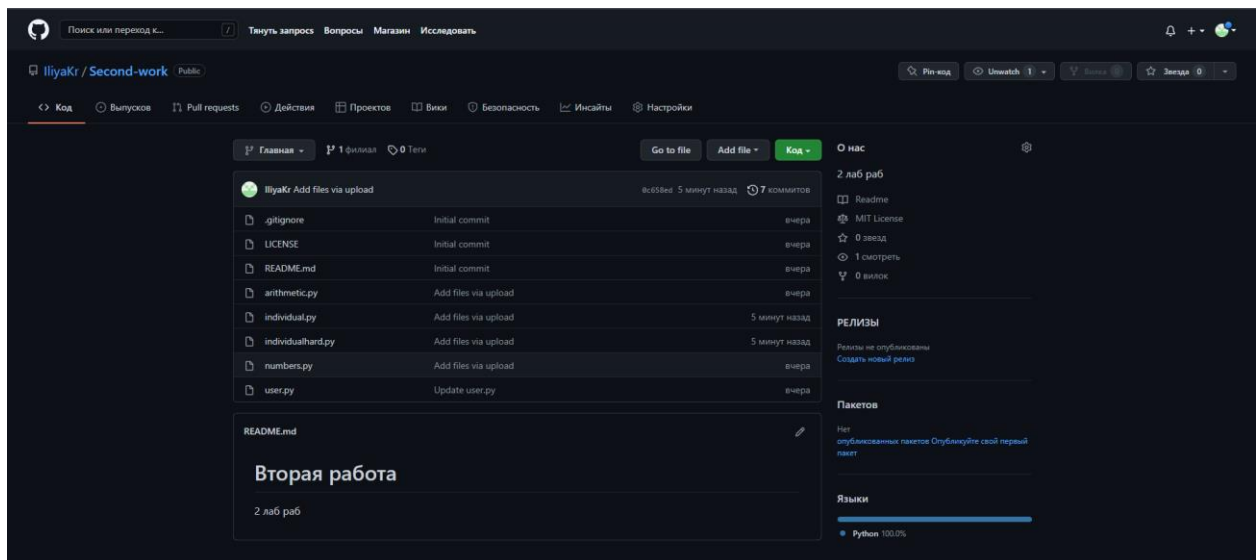


Рис 11. Страница GitHub с выполненными задачами
(<https://github.com/IliyaKr/Second-work.git>)

Ответы на контрольные вопросы:

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

Ответ: Для начала скачиваем дистрибутив по ссылке. Запускаем скачанный файл. Выбираем способ установки (install now или customize installation). Отмечаем необходимые опции для установки. Выбираем место установки (доступно при выборе Customize installation). successful

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

Ответ: Anaconda наиболее известна как дистрибутив Python со встроенным в него пакетным менеджером conda. Она позволяет изолировать окружение проекта от системной версии Python, который критически необходим для работы системы. Использование `sudo pip` считается плохой практикой. Также conda позволяет без проблем переносить окружение с одной машины на другую. Кроме того, если вы что-то сломаете, то с Anaconda вы всегда сможете откатиться на более старую версию окружения. Конечно, если вы позаботитесь о регулярных бэкапах. С системной версией Python это гораздо сложнее и может потребовать переустановки системы.

3. Как осуществить проверку работоспособности пакета Anaconda?

Проверка 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

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

Ответ: Запускаем прогу. И он не содержит ни одного файла с текстом программы. Создадим его. Нажимаем правую кнопку мыши, выбираем «New» -> «Python File». Вводим имя файла, например, ex1 и этот файл автоматически добавляется в наш проект. Здесь мы можем писать наши программы, на Питоне.

5. Как осуществить запуск программы с помощью IDE PyCharm?

Ответ: Здесь при первом запуске необходимо создать новый проект. Нажимаем «Create New Project». В поле «Location» указывается расположение проекта и его имя.

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

Ответ: Его суть в удобстве использования языка Питон, а именно его внешний вид и простота, сопровождаемая автоматикой действий, позволяющей пользователям работать с данными программами.

7. Почему язык программирования Python называется языком динамической типизации?

Ответ: потому что в Питоне тип переменной определяется непосредственно при выполнении программы

8. Какие существуют основные типы в языке программирования Python?

Ответ:

В Python типы данных можно разделить на встроенные в интерпретатор (*built-in*) и не

встроенные, которые можно использовать при импортировании соответствующих модулей.

К основным встроенным типам относятся:

1. *None* (неопределенное значение переменной)

2. Логические переменные (*Boolean Type*)

3. Числа (*Numeric Type*)

- *int* – целое число
- *float* – число с плавающей точкой
- *complex* – комплексное число

4. Списки (*Sequence Type*)

- *list* – список
- *tuple* – кортеж
- *range* – диапазон

5. Строки (*Text Sequence Type*)

- *str*

6. Бинарные списки (*Binary Sequence Types*)

- *bytes* – байты
- *bytearray* – массивы байт
- *memoryview* – специальные объекты для доступа к внутренним данным объекта через protocol buffer

7. Множества (*Set Types*)

- *set* – множество
- *frozenset* – неизменяемое множество

8. 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?

Ответ: 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`

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.

10. Как получить список ключевых слов в Python?

Ответ: 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()*?

Ответ: 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.

Ответ: 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?

Ответ: Целочисленное деление (*div*) отличается от обычной операции деления тем, что возвращает целую часть частного, дробная часть отбрасывается. Перед выполнением операции оба операнда округляются до целых значений.

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

Ответ: 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*.

Ответ: В стандартную поставку Python входит библиотека *math*, в которой содержится большое количество часто используемых математических функций. Для работы с данным модулем его предварительно нужно импортировать.

Рассмотрим наиболее часто используемые функции.

math.ceil(x)

Возвращает ближайшее целое число большее, чем x.

```
>>> math.ceil(3.2)
4
```

math.fabs(x)

Возвращает абсолютное значение числа.

```
>>> math.fabs(-7)
7.0
```

math.factorial(x)

Вычисляет факториал x.

```
>>> math.factorial(5)
120
```

math.floor(x)

Возвращает ближайшее целое число меньшее, чем x.

```
>>> math.floor(3.2)
3
```

math.exp(x)

Вычисляет e^x .

```
>>> math.exp(3)
20.085536923187668
```

math.log2(x)

Логарифм по основанию 2.

math.log10(x)

Логарифм по основанию 10.

math.log(x[, base])

По умолчанию вычисляет логарифм по основанию e, дополнительно можно указать основание логарифма.

```
>>> math.log2(8)
3.0
>>> math.log10(1000)
3.0
>>> math.log(5)
1.6094379124341003
>>> math.log(4, 8)
0.6666666666666667
```

math.pow(x, y)

Вычисляет значение x в степени y.

```
>>> math.pow(3, 4)
81.0
```

И так далее.

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

Ответ: `sep` — с помощью этого параметра вы можете указать разделитель строк. А по умолчанию в качестве разделителя используется пробел

`end` — этот параметр позволяет указать, что нужно добавить после последней строки. По умолчанию добавляется управляющий символ `'\n'` (перевод строки)

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.

Ответ: Метод `format()`, который определен у строк, позволяет форматировать строку, вставляя в нее на место плейсхолдеров

определенные **значения**. Для вставки в строку используются специальные параметры, которые обрамляются фигурными скобками (`{}`).

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

Ответ: для **ввода** нужно нажать Enter после завершения набора текста. Обычно Enter добавляет символ новой строки (`\n`), но не в этом случае. Введенная строка просто будет передана приложению.

Вывод: я исследовал процесс установки и базовые возможности языка Python3.