МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт цифрового развития

Кафедра инфокоммуникаций

Основы кроссплатформенного программирования

Отчет по лабораторной работе №2.1

Основы языка Python.

Выполнил студент группы

ИВТ-б-о-21-1 (2)

Стригалов Д.М. « »_____20__г.

Подпись студента_____

Работа защищена « »_____20__г.

Проверил доцент
Кафедры инфокоммуникаций, старший преподаватель
Воронкин Р.А.

_____
(подпись)

Ставрополь 2022

Тема: Основы языка Python.

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

Порядок выполнения работы:

```
# Created by https://www.toptal.com/developers/gitignore/api/python,pycharm
# Edit at https://www.toptal.com/developers/gitignore?templates=python,pycharm

### PyCharm ###
# Covers JetBrains IDEs: IntelliJ, RubyMine, PhpStorm, AppCode, PyCharm, CLion, Android Studio, WebStorm and Rider
# Reference: https://intellij-support.jetbrains.com/hc/en-us/articles/206544839

# User-specific stuff
.idea/**/workspace.xml
.idea/**/tasks.xml
.idea/**/usage.statistics.xml
.idea/**/dictionaries
.idea/**/shelf

# AWS User-specific
.idea/**/aws.xml

# Generated files
.idea/**/contentModel.xml

# Sensitive or high-churn files
.idea/**/dataSources/
.idea/**/dataSources.ids
.idea/**/dataSources.local.xml
.idea/**/sqlDataSources.xml
.idea/**/dynamic.xml
.idea/**/uiDesigner.xml
.idea/**/dbnavigator.xml

# Gradle
.idea/**/gradle.xml
.idea/**/libraries

# Gradle and Maven with auto-import
# When using Gradle or Maven with auto-import, you should exclude module files,
# since they will be recreated, and may cause churn.  Uncomment if using
# auto-import.
```

Рисунок 1 -Добавление правил в .gitignore

```
f:\Programs\GitHub\Lab-2.1>git flow init

Which branch should be used for bringing forth production releases?
   - main
Branch name for production releases: [main] main
Branch name for "next release" development: [develop] develop

How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [F:/Programs/GitHub/Lab-2.1/.git/hooks]
```

Рисунок 2 – Организация репозитория согласно модели ветвления git-flow

```
print("What is your name?")
name = str(input())
print("How old are you?")
age = int(input())
print("Where are you from?")
place = str(input())
print("This is ", name)
print("(S)He is", age, "years old")
print("(S)He live in", place)
```

Рисунок 3 – Задача №1

```
print("4 * 100 - 54 =")
print("Please,write an answer:")
answer = int(input())
print("Correct answer is", 4 * 100 - 54)
print("Your answer:", answer)
```

Рисунок 4 – Задача №2

```
print("Write 4 numbers:")
a = int(input())
b = int(input())
c = int(input())
d = int(input())
sum1 = a+b
sum2 = c+d
s = sum1/sum2
d = round(s,2)
print(d)
```

Рисунок 5 – Задача №3

```python
import math
a = float(input("Длина первого катета: "))
b = float(input("Длина второго катета: "))
g = math.sqrt(a**2+b**2)
G = round(g,2)
print("Гипотенуза равна:",G)
```

Рисунок 6 – Индивидуальное задание

```
f:\Programs\GitHub\Lab-2.1>git merge develop
Updating ac3dc54..fc86342
Fast-forward
 Python/arithmetic.py | 10 ++++++++++
 Python/individual.py |  6 ++++++
 Python/numbers.py    |  7 +++++++
 Python/user.py       |  5 +++++
 4 files changed, 28 insertions(+)
 create mode 100644 Python/arithmetic.py
 create mode 100644 Python/individual.py
 create mode 100644 Python/numbers.py
 create mode 100644 Python/user.py
```

Рисунок 7 – Слияние веток

Рисунок 8 – Задание повышенной сложности

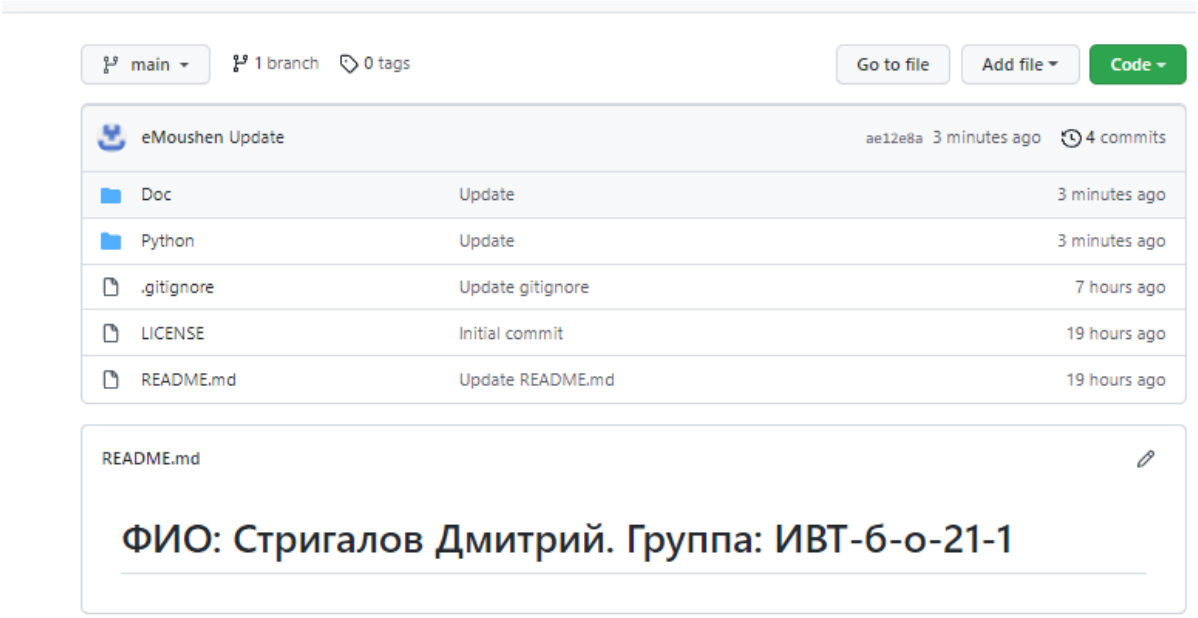

Рисунок 9 – Изменения в удалённом репозитории

Контрольные вопросы:

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

Основные этапы установки Python в Windows и Linux:

1) Скачивание дистрибутива с сайта python.org;
2) Запуск установочного файла;
3) Выбор необходимых опций установки;
4) Указание пути установки;5) Установка Python.

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

Пакет Anaconda содержит в себе интерпретатор языка Python 2 и 3 версии, набор наиболее часто используемых библиотек и удобную среду разработки и исполнения.

3. Как осуществить проверку работоспособности пакета Anaconda?

Для проверки работоспособности пакета Anaconda нужно запустить программу Anaconda Prompt, после в появившейся командной строке ввести «jupyter notebook». После этих действий отобразится процесс загрузки веб-среды Jupyter Notebook. В открывшейся среде нужно создать новый ноутбук, после в поле для кода ввести пробный код, например «print(“Hello, World!”)».

Если после запуска кода на экране появилась надпись «Hello, World!», то Anaconda установлен правильно.

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

Используемый интерпретатор языка Python в IDE PyCharm задается при создании нового проекта в соответствующей строке.

5. Как осуществить запуск программы с помощью IDE PyCharm?

Для запуска программы с помощью IDE PyCharm нужно открыть код программы, после чего в правом верхнем углу нажать кнопку «Run».

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

В интерактивном режиме работы Python ожидает ввода команд пользователя. При вводе команды интерпретатор выполнит строку и отобразит строкой ниже результат своей работы.

В пакетном режиме работы Python будет только выполнять уже написанный код. Для этого нужно набрать в командной строке ”python «название файла».py”.

7. Почему язык программирования Python называется языком динамической типизации?

Язык Python называется языком динамической типизации потому, что тип переменной определяется непосредственно при выполнении программы, а не на этапе компиляции, как в языках статической типизации.

8. Какие существуют основные типы в языке программирования Python?

Основные типы данных:

- None (неопределенное значение переменной);
- логические переменные;
- числа;
- списки;
- строки;
- бинарные списки (байты, массивы байт);
- множества;- словари.

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

Объект – это абстракция для представления данных, данные – это числа, списки, строки и т.д. Для создания объекта нужно написать его имя, потом поставить знак равенства и значение, с которым объект будет создан.

10. Как получить список ключевых слов в Python?

Для получения списка ключевых слов в Python нужно подключить модуль keyword и воспользоваться командой keyword.kwlist.

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

Функции id() и type() задают идентификатор объекту, тип переменной.

12. Что такое изменяемые и неизменяемые типы в Python.

Изменяемые типы данных – типы данных, которые могут изменяться в процессе выполнения кода программы. К ним относятся списки (list), множества (set) и словари (dict).

Неизменяемые типы данных – типы данных, неизменяемые в процессе выполнения кода программы. К ним относятся целые числа (int), числа с плавающей точкой (float), комплексные числа (complex), логические переменные (bool), кортежи (tuple), строки (str) и неизменяемые множества

(frozen set).

13. Чем отличаются операции деления и целочисленного деления? Деление возвращает частное с его дробной частью (при наличии).

Целочисленное деление возвращает целую часть частного, а дробная часть отбрасывается.

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

Для работы с комплексными числами используются функции complex(a, b), x.real, x.imag, x.conjugate().

15. Каково назначение и основные функции библиотеки (модуля) math? По аналогии с модулем math изучите самостоятельно назначение и основные функции модуля cmath.

Библиотека math содержит в себе большое количество часто используемых математических функций, например math.celi(x), math.fabs(x), math.factorial(x) и т.д.

Основные функции библиотеки: math.celi(x), math.fabs(x), math.factorial(x), math.floor(x), math.exp(x), math.log2(x), math.log10(x), math.log(x[,base]), math.pow(x,y), math.sqrt(x), math.sin(x),

math.cos(x), math.tan(x), math.acos(x), math.asin(x), math.atan(x), math.pi, math.e.

Модуль cmath предоставляет функции для работы с комплексными числами.

Основные функции модуля cmath: cmath.phase(x), cmath.polar(x), cmath.rect(x), cmath.exp(x), cmath.log(x[, base]), cmath.acosh(x), cmath.asinh(x), cmath.atanh(x).

16. Каково назначение именных параметров sep и end в функции print()? Параметр sep указывает отличный от пробела разделитель строк.

Параметр end указывает, что делать после вывода строки (поставить знак, сделать перенос на несколько строк).

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.

Метод format() применяется к строке и позволяет подставлять типы данных в неё (%s, %d, %f), указывать количество знаков после запятой для чисел (%1f).

F-строки позволяют форматировать строки схожим способом, как format(), но с некоторыми отличиями. Они позволяют подставить значение в строку, лишь указав имя переменной в фигурных скобках, использовать расширенное форматирование чисел, форматировать дату без вызова метода

strftime(), совершать базовые арифметические операции прямо в строке, обращаться к значениям списков по индексу и элементам словаря по ключу, и т.д.

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

Ввод с консоли значения переменной производится при помощи функции input().

Вывод: Исследовали процесс установки и базовый возможности языка Python3, написали несколько задач.