

и 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»
Вариант ____

Выполнил:

Репкин Александр Павлович
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной
техники и автоматизированных
систем», очная форма обучения

(подпись)

Руководитель практики:

Воронкин Р.А., канд. техн. наук,
доцент кафедры инфокоммуникаций

(подпись)

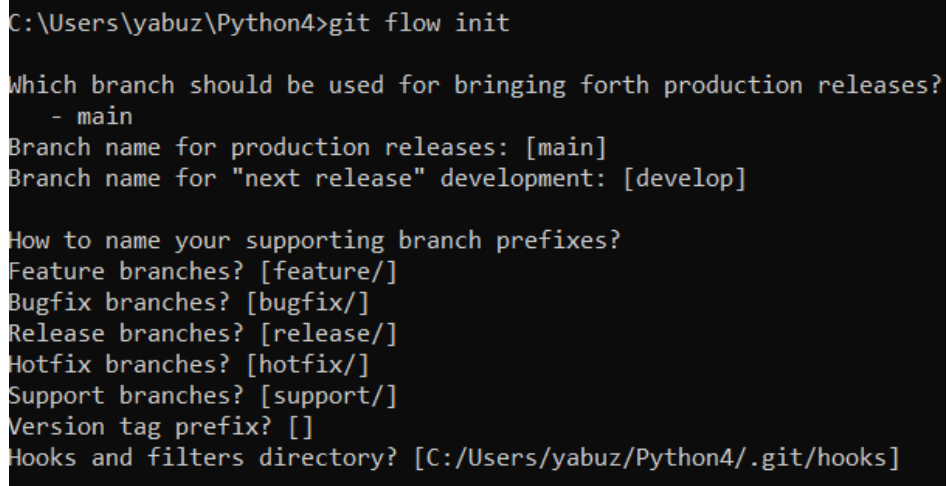
Отчет защищен с оценкой _____ Дата защиты _____
Ставрополь, 2023 г.

Тема: Основы языка Python.

Цель: исследовать процесс установки и базовых возможностей языка Python версии 3.x.

Порядок выполнения работы:

1. Репозиторий организован в соответствии с git-flow.



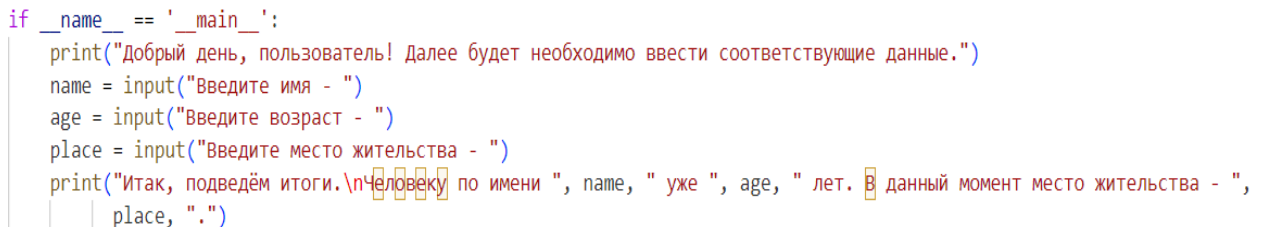
```
C:\Users\yabuz\Python4>git flow init

Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main]
Branch name for "next release" development: [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? [C:/Users/yabuz/Python4/.git/hooks]
```

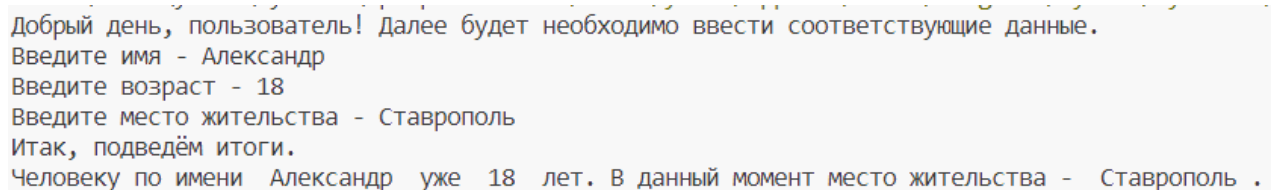
Рисунок 1. Инициализация git-flow.

2. Выполнено первое общее задание. В нём требовалось: “Напишите программу, которая запрашивала бы у пользователя его имя, возраст, место жительства. После этого нужно вывести полученные значения.”.



```
if __name__ == '__main__':
    print("Добрый день, пользователь! Далее будет необходимо ввести соответствующие данные.")
    name = input("Введите имя - ")
    age = input("Введите возраст - ")
    place = input("Введите место жительства - ")
    print("Итак, подведём итоги.\nЧеловеку по имени ", name, " уже ", age, " лет. В данный момент место жительства - ",
          place, ".")
```

Рисунок 2. Полученный код общего задания №1.



```
Добрый день, пользователь! Далее будет необходимо ввести соответствующие данные.
Введите имя - Александр
Введите возраст - 18
Введите место жительства - Ставрополь
Итак, подведём итоги.
Человеку по имени Александр уже 18 лет. В данный момент место жительства - Ставрополь .
```

Рисунок 3. Полученный результат общего задания №1.

3. Выполнено второе общее задание. В нём требовалось: “Напишите программу, которая предлагала бы пользователю решить пример $4 * 100 - 54$. Потом выводила бы на экран правильный ответ и ответ пользователя.”.

```

if __name__ == '__main__':
    print("Good day, user! Let us begin the math test!\nHow do you think, what is the answer for: 4 * 100 - 54 ?")
    answer = input("Your answer is... - ")
    print("Attention! Now, the moment of truth!\nYour answer is - ",
          answer, "\nWhilst right answer is - 346.")
    input()

```

Рисунок 4. Полученный код общего задания №2.

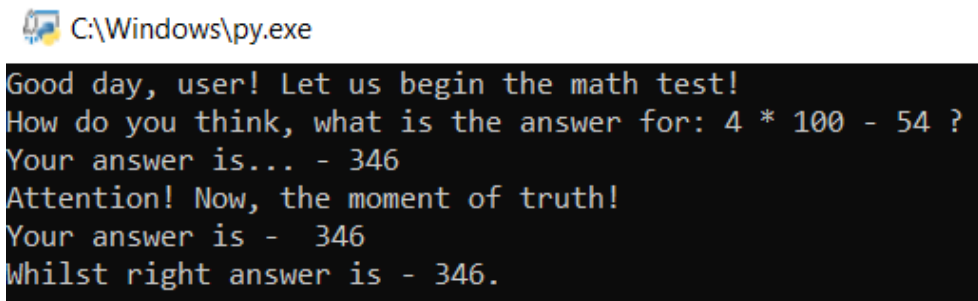


Рисунок 5. Полученный результат общего задания №2.

4. Выполнено третье общее задание. В нём требовалось: “Запросите у пользователя четыре числа. 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.”.

```

if __name__ == '__main__':
    print("Good day, user! This time, we need you to enter some values for us. Let the experiment begin!")
    first = int(input("Your first number is - "))
    second = int(input("Your second number is - "))
    third = int(input("Your third number is - "))
    fourth = int(input("Your fourth number is - "))
    first_sum = first + second
    second_sum = third + fourth
    answer = first_sum / second_sum
    print("Interesting...\nSo, according to our calculations, the answer is - ", round(answer, 2), ".")
    input()

```

Рисунок 6. Полученный код общего задания №3.

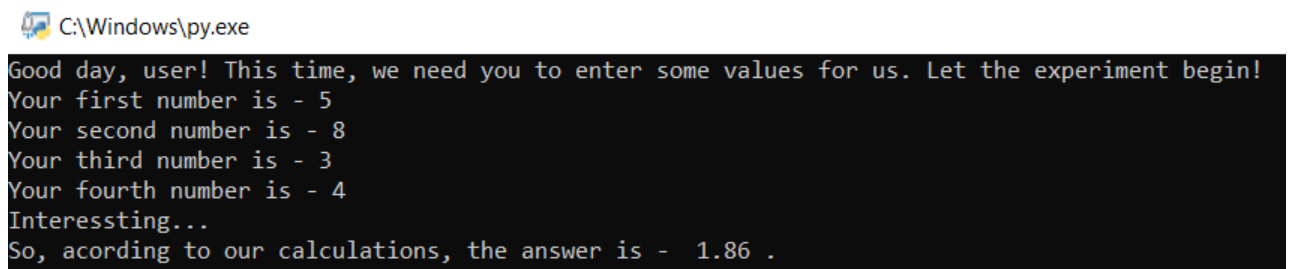


Рисунок 7. Полученный результат общего задания №3.

5. Выполнено индивидуальное задание: “Даны два числа. Найти их сумму, разность, произведение, а также частное от деления первого числа на второе.”.

```

print("Задание №4: Даны два числа. Найти их сумму, разность, произведение, а также частное от деления первого "
      "числа на второе.")
value1 = int(input("Первое число - "))
value2 = int(input("Второе число - "))
answer1 = value1 + value2
answer2 = value1 - value2
print("Ответ:\nSumm - ", answer1, "\nDifference - ", answer2, )
answer1 = value1 * value2
answer2 = value1 / value2
print("Multiplication - ", answer1, "\nQuotient - ", answer2, )

```

Рисунок 8. Полученный код индивидуального задания.

```

Задание №4: Даны два числа. Найти их сумму, разность, произведение, а также частное от деления первого числа на второе.
Первое число - 10
Второе число - 1
Ответ:
Summ - 21
Difference - 15
Multiplication - 54
Quotient - 6.0

```

Рисунок 9. Полученный результат индивидуального задания.

6. Выполнено задание повышенной сложности, №3: “Даны цифры двух десятичных целых чисел: трехзначного и двузначного, где a_1 и b_1 – число единиц, a_2 и b_2 – число десятков, a_3 – число сотен. Получить цифры числа, равного сумме заданных чисел (известно, что это число трехзначное). Числа-слагаемые и число-результат не определять; условный оператор не использовать.”.

```

if __name__ == '__main__':
    print("Добрый день, пользователь! Здесь выполнено усложненное задание:\nПолучить цифры числа, равного сумме "
          "заданных чисел.")
    a1 = int(input("Введите единицы первого числа - "))
    a2 = int(input("Введите десятки первого числа - "))
    a3 = int(input("Введите сотни первого числа - "))
    b1 = int(input("Введите единицы второго числа - "))
    b2 = int(input("Введите десятки второго числа - "))
    print("Ответ:\n ", a3 + ((a2+b2 + (a1+b1)//10)//10),
          ((a2+b2 + (a1+b1)//10) % 10), (a1+b1) % 10)

```

Рисунок 10. Полученный код усложнённого задания.

```

Добрый день, пользователь! Здесь выполнено усложненное задание:
Получить цифры числа, равного сумме заданных чисел.
Введите единицы первого числа - 6
Введите десятки первого числа - 9
Введите сотни первого числа - 3
Введите единицы второго числа - 9
Введите десятки второго числа - 6
Ответ:
4 6 5

```

Рисунок 11. Полученный результат усложнённого задания.

7. Ответы на вопросы.

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

Ответ: для операционной системы Windows дистрибутив распространяется либо в виде исполняемого файла (с расширением exe), либо в виде архивного файла (с расширением zip). Порядок установки: запустить скачанный установочный файл; выбрать способ установки. (Install Now или Customize installation. Install Now - Python установится в папку по указанному пути. Помимо самого интерпретатора будет установлен IDLE (интегрированная среда разработки), pip (пакетный менеджер) и документация, также будут созданы соответствующие ярлыки и установлены связи файлов, имеющие расширение .py с интерпретатором Python. Customize installation – вариант настраиваемой установки.); отметить необходимые опции установки (доступно при выборе Customize installation. На этом шаге нам предлагается отметить дополнения, устанавливаемые вместе с интерпретатором Python. Documentation – установка документаций. pip – установка пакетного менеджера pip. tcl/tk and IDLE – установка интегрированной среды разработки (IDLE) и библиотеки для построения графического интерфейса (tkinter)); выбрать место установки (доступно при выборе Customize installation. Помимо указания пути, данное окно позволяет внести дополнительные изменения в процесс установки с помощью опций: Install for all users – Установить для всех пользователей. Если не выбрать данную опцию, то будет предложен вариант инсталляции в папку пользователя, устанавливающего интерпретатор. Associate files with Python – Связать файлы, имеющие расширение .py, с Python. При выборе данной опции будут внесены изменения в Windows, позволяющие запускать Python скрипты по двойному щелчку мыши. Create shortcuts for installed applications – Создать ярлыки для запуска приложений. Add Python to environment variables – Добавить пути до интерпретатора Python в переменную PATH. Precompile standard library – Провести прекомпиляцию стандартной библиотеки). Установка Python в Linux: чаще всего интерпретатор Python уже входит в

состав дистрибутива. Это можно проверить, набрав в терминале `python` (Запустится Python2) или `python 3` (Запустится Python3). Если при попытке запустить Python, выдается сообщение о том, что он не установлен, или установлен, но не тот, то можно собрать Python из исходников или взять из репозитория (Для установки из репозитория в Ubuntu есть команда `sudo apt-get install python3`).

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

Ответ: Пакет Anaconda включает в себя интерпретатор языка Python (есть версии 2 и 3), набор наиболее часто используемых библиотек и удобную среду разработки и исполнения, запускаемую в браузере.

3) Как осуществить проверку работоспособности пакета Anaconda?

Ответ: для выполнения проверки необходимо запустить командный процессор с поддержкой виртуальных окружений Anaconda. В Windows это можно сделать, выбрав пункт Anaconda Prompt в Пуске. В появившейся командной строке необходимо ввести `jupyter notebook`, в результате чего отобразится процесс загрузки веб-среды Jupyter Notebook и запустится браузер со средой разработки.

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

Ответ: необходимо открыть настройки нужного проекта в Pycharm, выбрав пункт меню `File > Settings`. Найти "Python Interpreter", после чего на открывшейся странице "Python Interpreters" нажать на "+" и выбрать "Add...". Выбрать нужное расположение интерпретатора Python и нажать "OK".

5) Как осуществить запуск программы с помощью IDE PyCharm?

Ответ: для запуска есть команда "`python` название файла и его расширение". Или же можно просто вводить необходимые команды, например `print("Что-то")` (Это работает только в интерактивном режиме, в командной строке можно только запускать файлы).

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

Ответ: Интерактивный - Python используется для различных вычислений. Пакетный - сначала записывается вся программа, потом она выполняется полностью.

7) Почему язык программирования Python называется языком динамической типизации?

Ответ: нет необходимости определять тип переменной, т.к. это происходит автоматически в процессе выполнения программы.

8) Какие существуют основные типы в языке программирования Python?

Ответ: логические переменные (boolean), числа (int – целое число, float – число с плавающей точкой, complex – комплексное число), Списки (list – список, tuple – кортеж, range – диапазон), Строки (str), Бинарные списки (bytes – байты, bytearray – массивы байт, memoryview – специальные объекты для доступа к внутренним данным объекта через protocol buffer), Множества (set – множество, frozenset – неизменяемое множество), Словари (dict – словарь).

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

Ответ: для объявления и инициализации переменной необходимо написать её имя, потом поставить знак равенства и значение, с которым эта переменная будет создана. Передаваемое значение в Python является объектом (Объект – абстракция для представления данных, данные – числа, списки, строки и т.п. – объекты и отношения между ними. Каждый объект имеет три атрибута: идентификатор (уникальный признак объекта, позволяющий отличать объекты друг от друга), значение (информация, хранящаяся в памяти, которой управляет интерпретатор), тип).

10) Как получить список ключевых слов в Python?

Ответ: для этого нужно подключить keyword (import keyword) и вывести список ключевых слов (print(keyword.kwlist)).

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

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

12) Что такое изменяемые и неизменяемые типы в Python?

Ответ: изменяемые (Объект может измениться, например, добавиться или замениться новое значение) типы данных – списки, множества, словари. Неизменяемые (Объект неизменен, например – создание целочисленной переменной и присваивание ей значения) типы данных – числа (целые, с плавающей точкой, комплексные), логические переменные, строки, неизменяемые множества.

13) Чем отличаются операции деления и целочисленного деления?

Ответ: целочисленное деление возвращает только целую часть частного, в то время как обычное деление возвращает частное полностью, вместе с дробной частью.

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

Ответ: создание комплексного числа возможно при помощи функции `complex(действительная часть, мнимая часть)`. Над комплексными числами возможны операции, деления, умножения, суммы, вычитания.

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

Ответ: `math` – библиотека, содержащая математические функции (`sin`, `cos`, `floor`, `ceil`, `fabs` и т.д.).

`math.ceil(x)` - возвращает ближайшее целое число большее, чем `x`.

`math.fabs(x)` - возвращает абсолютное значение числа.

`math.factorial(x)` - вычисляет факториал `x`.

`math.floor(x)` - возвращает ближайшее целое число меньшее, чем `x`.

`math.exp(x)` - вычисляет e^{**x} .

`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.pow(x, y)` - вычисляет значение x в степени y .

`math.sqrt(x)` - корень квадратный от x .

`math.cos(x)` - косинус от x .

`math.sin(x)` - синус от x .

`math.tan(x)` - тангенс от x .

`math.acos(x)` - арккосинус от x .

`math.asin(x)` - арксинус от x .

`math.atan(x)` - арктангенс от x .

Константы: `math.pi` - число Π , `math.e` - число e .

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

Ответ: `sep` — с помощью этого параметра можно указать разделитель строк (По умолчанию в качестве разделителя используется пробел). `end` — 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?

Ответ: метод `format()` содержит в скобках данные, которые необходимо указать в фигурных скобках в `print`. На нулевое место подставится первый аргумент метода `format()`, на место с номером 1 – второй и т. д.

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

Ответ: За ввод в программу данных с клавиатуры в Python отвечает функция `input()`. Когда вызывается эта функция, программа останавливает свое выполнение и ждет, когда пользователь введет текст. После этого, когда будет нажат `Enter`, функция `input()` заберет введенный текст и передаст его программе, которая уже будет обрабатывать его согласно своим алгоритмам. Для преобразования полученных строковых значений в целочисленные и

вещественные, необходимо обернуть вызов `input` в `int()` или `float()` соответственно.

Вывод: в ходе выполнения лабораторной работы были исследованы процесс установки и базовые возможности языка Python версии 3.x. Выполнены общие, индивидуальные и усложнённые задания. Получены знания о начальных этапах работы с языком программирования Python.