

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ
ФЕДЕРАЦИИ**
**Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРОКАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

Институт цифрового развития

ОТЧЁТ

по лабораторной работе №2.1

Дисциплина: «Основы кроссплатформенного программирования»

Тема: «Основы языка Python»

Выполнил: студент 1 курса

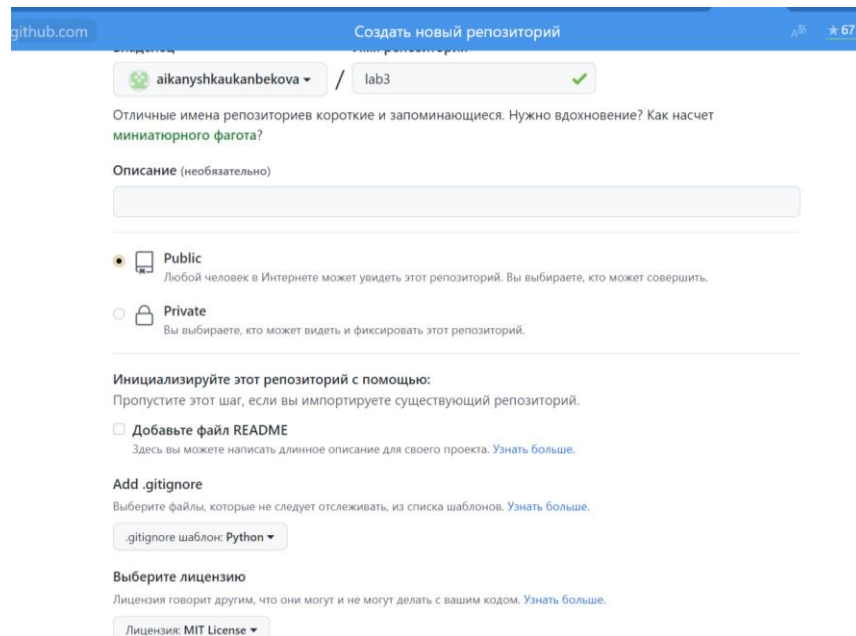
группы ИВТ-б-о-21-1

Уланбекова Айканыш Уланбековна

Ставрополь 2022

Выполнение работы:

1. Создал репозиторий GitHub с лицензией MIT, добавил .gitignore с ЯП python, клонировал репозиторий на ПК и организовал репозиторий согласно модели ветвления git-flow:



github.com Создать новый репозиторий

aikanyshkaukanbekova / lab3

Отличные имена репозитория короткие и запоминающиеся. Нужно вдохновение? Как насчет **миниатюрного факота**?

Описание (необязательно)

☒ Public
Любой человек в Интернете может увидеть этот репозиторий. Вы выбираете, кто может совершить.

☐ Private
Вы выбираете, кто может видеть и фиксировать этот репозиторий.

Инициализируйте этот репозиторий с помощью:
Пропустите этот шаг, если вы импортируете существующий репозиторий.

☐ Добавьте файл README
Здесь вы можете написать длинное описание для своего проекта. [Узнать больше.](#)

Add .gitignore
Выберите файлы, которые не следует отслеживать, из списка шаблонов. [Узнать больше.](#)

.gitignore шаблон: Python

Выберите лицензию
Лицензия говорит другим, что они могут и не могут делать с вашим кодом. [Узнать больше.](#)

Лицензия: MIT License

Рисунок 1.1 Создание репозитория

```
C:\Users\User>cd C:\Users\User\Desktop
C:\Users\User\Desktop>cd lab3
C:\Users\User\Desktop\lab3>git clone https://github.com/aikanyshkaukanbekova/lab3.git
Cloning into 'lab3'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), done.
C:\Users\User\Desktop\lab3>A_
```

Рисунок 1.2 Клонирование репозитория

```
C:\Users\User\Desktop\lab3>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/User/Desktop/lab3/.git/hooks]
```

Рисунок 1.3 Организация репозитория согласно модели ветвления git-flow

2. Написал программу user.py, которая запрашивала бы у пользователя имя, возраст и место жительства, после этого выводила бы 3 строки

```
"This is `имя`"
"It is `возраст`"
```

"(S)he live in `место_жительства`"



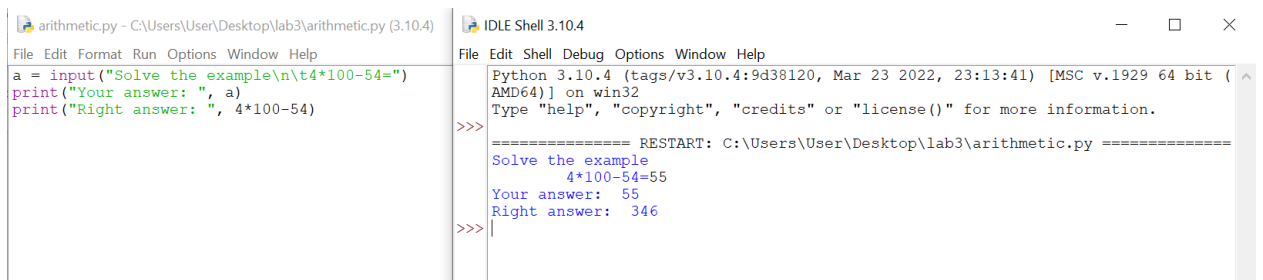
```
user.py - C:\Users\User\Desktop\lab3\user.py (3.10.4)
File Edit Format Run Options Window Help
n = input("Write your name:\n")
a = input("Write your age:\n")
p = input("Write your place of birth:\n")
print("\nThis is", n)
print("It's ", a)
print("(S)he live in ", p)

IDLE Shell 3.10.4
File Edit Shell Debug Options Window Help
Python 3.10.4 (tags/v3.10.4:9d38120, Mar 23 2022, 23:13:41) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\User\Desktop\lab3\user.py =====
Write your name:
Aikanysh
Write your age:
19
Write your place of birth:
Kyrgyzstan

This is Aikanysh
It's 19
(S)he live in Kyrgyzstan
>>>
```

Рисунок 1.1 Программа user в репозитории

4. . Написал программу (файл arithmetic.py), которая предлагала бы пользователю решить пример $4 * 100 - 54$. Потом выводила бы на экран правильный ответ и ответ пользователя.

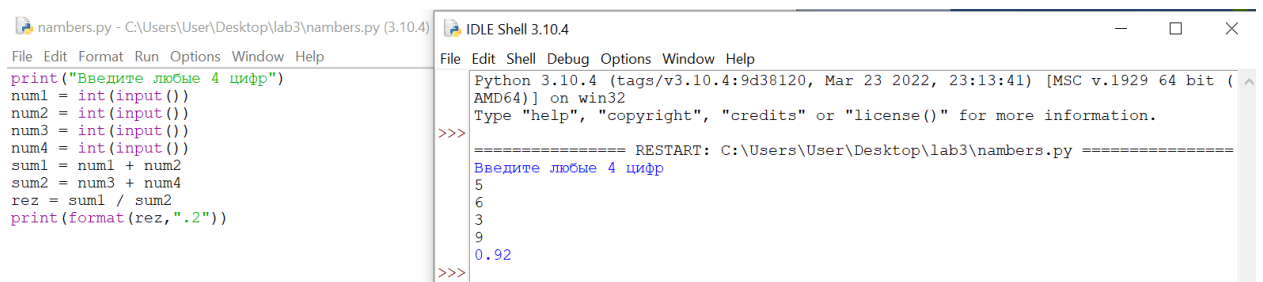


```
arithmetic.py - C:\Users\User\Desktop\lab3\arithmetic.py (3.10.4)
File Edit Format Run Options Window Help
a = input("Solve the example\n\t4*100-54=")
print("Your answer: ", a)
print("Right answer: ", 4*100-54)

IDLE Shell 3.10.4
File Edit Shell Debug Options Window Help
Python 3.10.4 (tags/v3.10.4:9d38120, Mar 23 2022, 23:13:41) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\User\Desktop\lab3\arithmetic.py =====
Solve the example
4*100-54=55
Your answer: 55
Right answer: 346
>>>
```

Рисунок 4.1 Программа arithmetic.py

5. Написал программу numbers.py, которая запрашивает у пользователя 4 числа, отдельно складывает первые два и вторые два, затем делит первую сумму на вторую, после выводит рез-т на экран с точностью до сотен.



```
numbers.py - C:\Users\User\Desktop\lab3\numbers.py (3.10.4)
File Edit Format Run Options Window Help
print("Введите любые 4 цифр")
num1 = int(input())
num2 = int(input())
num3 = int(input())
num4 = int(input())
sum1 = num1 + num2
sum2 = num3 + num4
rez = sum1 / sum2
print(format(rez, ".2"))

IDLE Shell 3.10.4
File Edit Shell Debug Options Window Help
Python 3.10.4 (tags/v3.10.4:9d38120, Mar 23 2022, 23:13:41) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\User\Desktop\lab3\numbers.py =====
Введите любые 4 цифр
5
6
3
9
0.92
>>>
```

Рисунок 5.1 Программа numbers.py

5. Написал программу для индивидуального задания:

Найти площадь кольца по заданным внешнему и внутреннему радиусам.

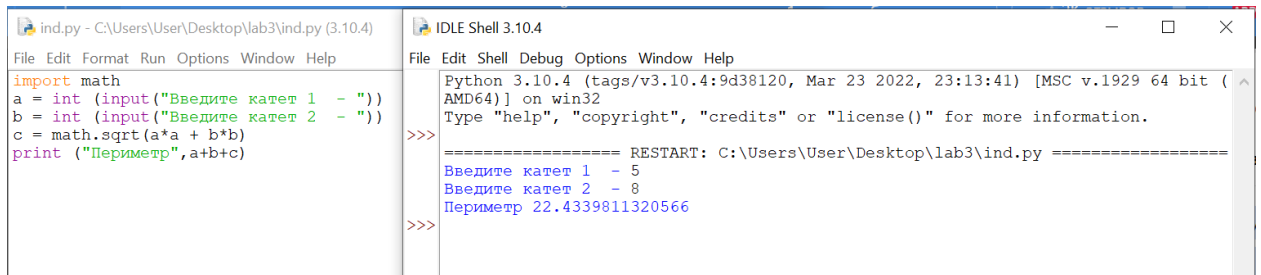


Рисунок 5.1 Программа индивидуального задания

6. Сделал коммит изменений в ветку разработки, выполнил ее слияние с веткой main и отправил сделанные изменения на уд. репозиторий.

```
C:\Users\User\Desktop\lab3>git commit -m "o"
[develop 3c06b97] o
5 files changed, 25 insertions(+)
create mode 100644 arithmetic.py
create mode 100644 ind.py
create mode 160000 lab3
create mode 100644 numbers.py
create mode 100644 user.py
```

Рисунок 6.1 Коммит изменений в ветку develop

```
C:\Users\User\Desktop\lab3>git checkout main
warning: unable to rmdir 'lab3': Directory not empty
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

C:\Users\User\Desktop\lab3>git merge develop
Updating 6e74c72..3c06b97
Fast-forward
 arithmetic.py | 3 +++
 ind.py        | 6 ++++++
 lab3          | 1 +
 numbers.py    | 9 ++++++++
 user.py       | 6 ++++++
 5 files changed, 25 insertions(+)
 create mode 100644 arithmetic.py
 create mode 100644 ind.py
 create mode 160000 lab3
 create mode 100644 numbers.py
 create mode 100644 user.py
C:\Users\User\Desktop\lab3>Z_
```

Рисунок 6.2 Слияние ветки develop с веткой main

```
C:\Users\User\Desktop\lab3>git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 891 bytes | 891.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/aikanyshkaukanbekova/lab3.git
 6e74c72..3c06b97 main -> main
```

Рисунок 6.3 push коммитов на уд. репозиторий

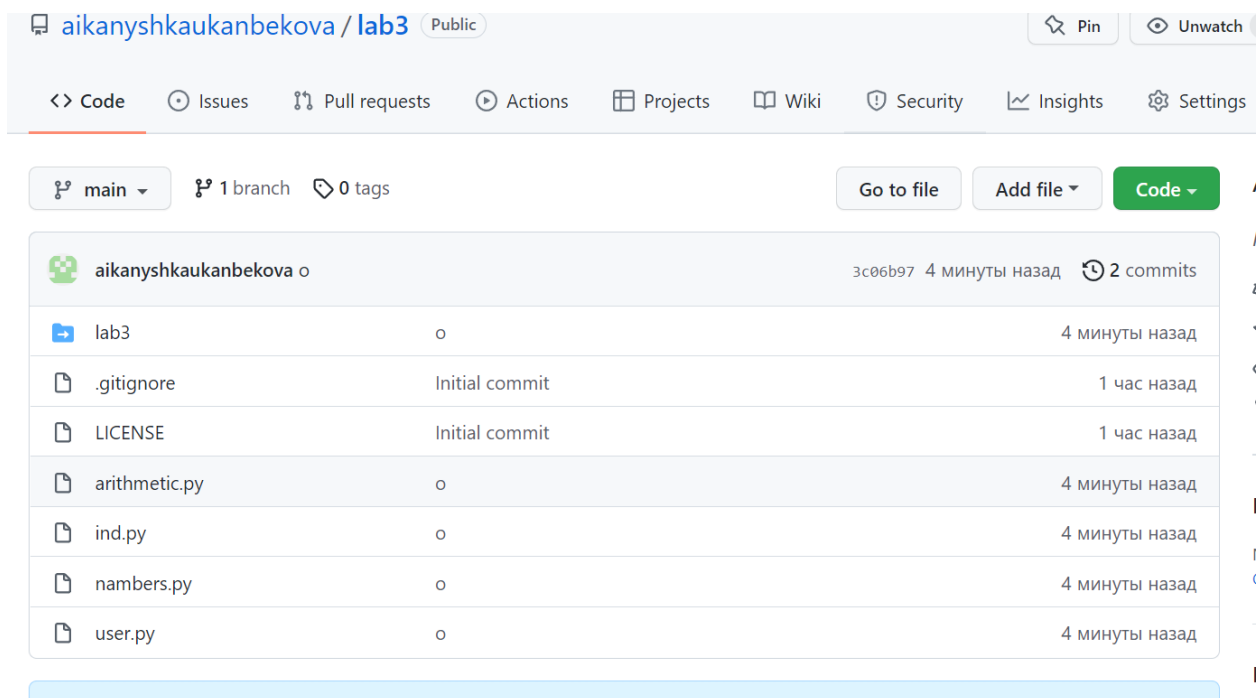


Рисунок 6.4 Изменения на уд. сервере

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

Linux: Чаще всего интерпретатор Python уже входит в состав дистрибутива.

Windows: Оsn. этапы установки Python на Windows:

- 1) Скачать дистрибутив с официального сайта;
- 2) Запустить скачанный установочный файл;
- 3) Выбрать способ установки;
- 4) Отметить необходимые опции установки;
- 5) Выбрать место устновки;
- 6) Готово.

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

Пакет Anaconda содержит версии языка Python 2 и 3, набор наиболее часто используемых библиотек и удобную среду разработки и исполнения, запускаемую в браузере, а также на Anaconda удобнее запускать примеры.

3. Как осуществить проверку работоспособности пакета Anaconda?

Для выполнения проверки работоспособности Anaconda необходимо вначале запустить командный процессор с поддержкой виртуальных окружений Anaconda. 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. Создать ноутбук для разработки, для этого нажать на кнопку New и в появившемся списке выбрать Python. В результате будет создана новая страница в браузере с ноутбуком. Ввести в первой ячейке команду `print("Hello, World!")` и нажать Alt+Enter на компьютере. Ниже ячейки должна появиться соответствующая надпись.

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

Указать путь до интерпретатора в настройках IDE, для этого:

- 1) Нажмите на шестеренку в верхнем правом углу, выберите "Add..".
- 2) Далее выберите "System Interpreter";
- 3) Нажмите на 3 точки "..." справа от поля в выборе интерпретатора;
- 4) Укажите путь до интерпретатора.

5. Как осуществить запуск программы с помощью IDE PyCharm?

Сочетанием клавиш Shift+F10.

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

Интерактивный.

Python можно использовать как калькулятор для различных вычислений, а если дополнительно подключить необходимые математические библиотеки, то по своим возможностям он становится практически равным таким пакетам как Matlab, Octave и т.п.

Проектный.

В этом режиме сначала записывается вся программа, а потом эта программа выполняется полностью.

7. Почему язык программирования Python называется языком динамической типизации?

Т. к. в ЯП Python проверка типа происходит во время выполнения, а не компиляции.

8. Какие существуют основные типы в языке программирования Python?

Типы в ЯП Python:

1. None
2. Логические переменные
3. Числа
4. Списки
5. Строки
6. Бинарные списки
7. Множества
8. Словари

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.

При инициализации переменной, на уровне интерпретатора, создается целочисленный объект, который имеет некоторый идентификатор, значение и тип. Посредством оператора “=” создается ссылка между переменной и объектом.

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() предназначена для получения значения идентичности объекта.

С помощью функции type() можно получить тип конкретного объекта.

12. Что такое изменяемые и неизменяемые типы в Python.

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).

13. Чем отличаются операции деления и целочисленного деления?

При целочисленном делении отбрасывается дробная часть от деления чисел, при операции деления дробная часть не отбрасывается.

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. У комплексного числа можно извлечь действительную (`x.real`) и мнимую части (`x.imag`).

Для получения комплексносопряженного число необходимо использовать метод `conjugate()`.

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

Для выполнения математических операций необходим модуль `math`.

Осн. операции библиотеки `math`:

`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 .

`math.pi` - число π .

`math.e` - число e .

16. Каково назначение именных 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? Примечание: в дополнение к рассмотренным средствам изучите самостоятельно работу с f-строками в Python.

Форматирование может выполняться в так называемом старом стиле или с помощью строкового метода `format`.

Символы `%s` , `%d` , `%f` подставляются значения переменных. Буквы `s`, `d`, `f` обозначают типы данных – строку, целое число, вещественное число.

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

Указать перед `input` тип данных: `int(input())`.