

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

**Кафедра инфокоммуникаций**

**«Основы ветвления Git»**

**Отчет по лабораторной работе № 2.1**  
**по дисциплине «Основы программной инженерии»**

Выполнил студент группы ПИЖ-б-о-21-1

Образцова М. Д. .«08» октября 2022г.

Подпись студента \_\_\_\_\_

Работа защищена « » \_\_\_\_\_ 20\_\_ г.

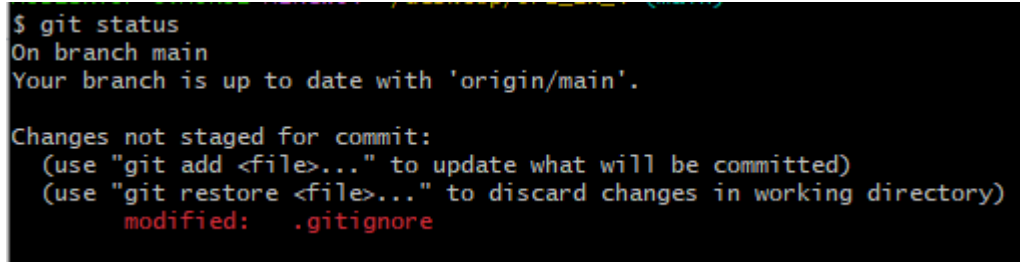
Проверил Воронкин Р.А. \_\_\_\_\_

(подпись)

Ставрополь 2022

МЕТОДИКА И ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

Дополните файл .gitignore необходимыми правилами для работы с IDE PyCharm.

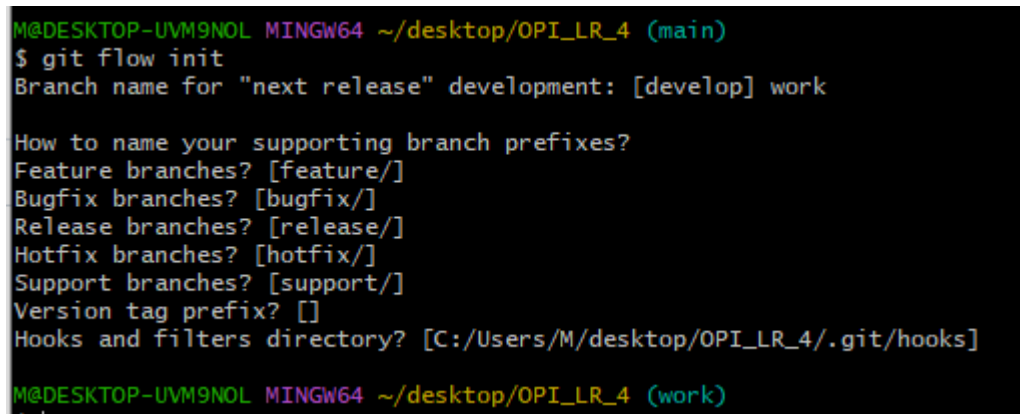


```
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   .gitignore
```

Рисунок 1– добавление файла .gitignore

5. Организуйте свой репозиторий в соответствии с моделью ветвления git-flow.



```
M@DESKTOP-UVM9NOL MINGW64 ~/desktop/OPI_LR_4 (main)
$ git flow init
Branch name for "next release" development: [develop] work

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/M/desktop/OPI_LR_4/.git/hooks]

M@DESKTOP-UVM9NOL MINGW64 ~/desktop/OPI_LR_4 (work)
```

Рисунок 2– моделью ветвления git-flow

6. Создайте проект PyCharm в папке репозитория.

7. Решите следующие задачи с помощью языка программирования Python3 и IDE PyCharm:

8. 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:

Вместо имя, возраст, место\_жительства должны быть данные, введенные пользователем.

Примечание: можно писать фразы на русском языке, но если вы планируете стать профессиональным программистом, привыкайте к английскому.

Код программы

```
name = input("What is your name? ")
age = input("How old are you? ")
city = input("Where are you live? ")
print("my name is {0},\ni am old {1}.\ni live in {2}.".format(name, age, city))
```

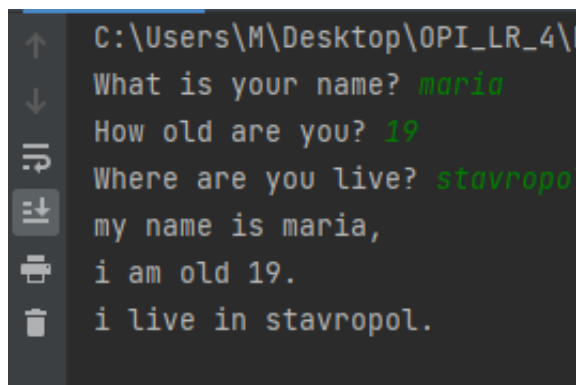


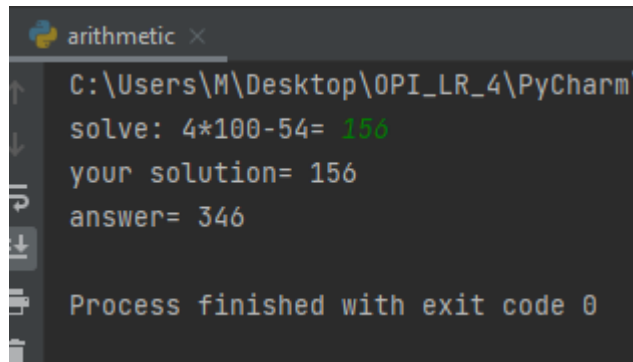
Рисунок 3—результат ее выполнения в IDE PyCharm

9. 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.

Код программы

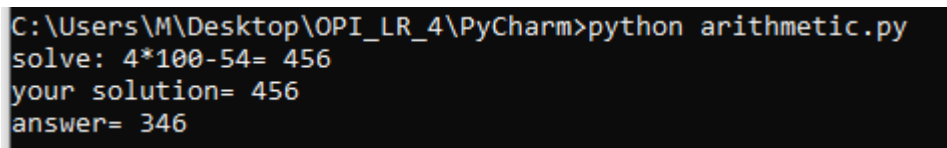
```
answer=int(input("solve: 4*100-54= "))
print("your solution= {0}\nanswer= {1}".format(answer, 4*100-54))
```



```
arithmetic x
C:\Users\M\Desktop\OPI_LR_4\PyCharm\
solve: 4*100-54= 156
your solution= 156
answer= 346

Process finished with exit code 0
```

Рисунок 4— результат ее выполнения в IDE PyCharm

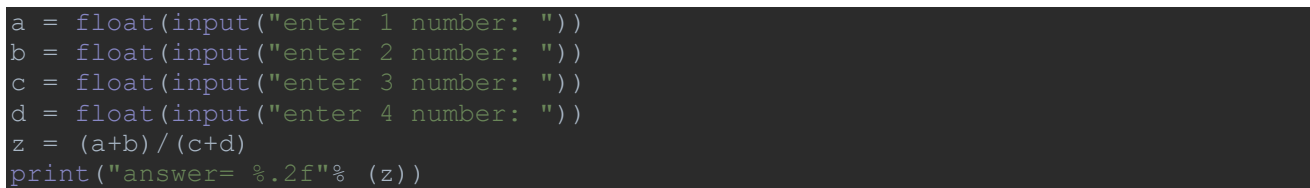


```
C:\Users\M\Desktop\OPI_LR_4\PyCharm>python arithmetic.py
solve: 4*100-54= 456
your solution= 456
answer= 346
```

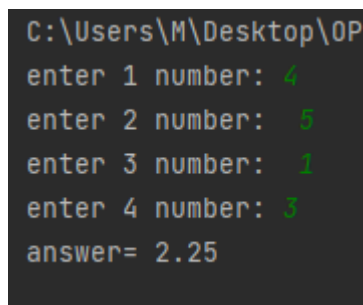
Рисунок 5— результат ее выполнения в командной строке

10. 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.

Код программы



```
a = float(input("enter 1 number: "))
b = float(input("enter 2 number: "))
c = float(input("enter 3 number: "))
d = float(input("enter 4 number: "))
z = (a+b)/(c+d)
print("answer= %.2f"% (z))
```



```
C:\Users\M\Desktop\OPI_LR_4\PyCharm\
enter 1 number: 4
enter 2 number: 5
enter 3 number: 1
enter 4 number: 3
answer= 2.25
```

Рисунок 6— результат ее выполнения в IDE PyCharm

```
cmd: Командная строка
answer= 346

C:\Users\M\Desktop\OPI_LR_4>python individual.py
enter 1 number: 5
enter 2 number: 6
enter 3 number: 8
enter 4 number: 9
answer= 0.65
```

Рисунок 7— результат ее выполнения в командной строке

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

(20) Составить программу решения линейного уравнения  $ax + b = 0, (a \neq 0)$ .

Код программы

```
a =int(input("enter a "))
b = int(input("enter b "))
x=(-b / a)
print("otvet= {0}".format(x))

C:\Users\M\Desktop\OPI_LR_4>python individual.py
enter a 5
enter b 10
otvet= -2.0
```

Рисунок 8— результат ее выполнения в IDE PyCharm

```
cmd: Командная строка
командой, исполняемой программой или пакетным файлом.

C:\Users\M\Desktop\OPI_LR_4\PyCharm>python individual.py
enter a 5
enter b 1
otvet= -0.2
```

Рисунок 9– результат ее выполнения в командной строке

12. Выполните коммит файлов user.py, arithmetic.py, numbers.py и individual.py в репозиторий git в ветку для разработки.

```
M@DESKTOP-UVM9NOL MINGW64 ~/desktop/OPI_LR_4 (work)
$ git add .

M@DESKTOP-UVM9NOL MINGW64 ~/desktop/OPI_LR_4 (work)
$ git commit --amend
hint: Waiting for your editor to close the file... dos2unix: convert
Users/M/Desktop/OPI_LR_4/.git/COMMIT_EDITMSG to Unix format...
[work e92938b] Progr
Date: Sat Oct 22 03:43:06 2022 +0300
6 files changed, 17 insertions(+), 16 deletions(-)
create mode 100644 PyCharm/.idea/.name
create mode 100644 PyCharm/arithmetic.py
create mode 100644 PyCharm/individual.py
delete mode 100644 PyCharm/main.py
create mode 100644 PyCharm/numbers.py
create mode 100644 PyCharm/user.py
```

13. Выполните слияние ветки для разработки с веткой master.

```
M@DESKTOP-UVM9NOL MINGW64 ~/desktop/OPI_LR_4 (main)
$ git merge work
Updating 4a344b1..e29ef5a
Fast-forward
 PyCharm/.idea/.gitignore      | 3 +++
 PyCharm/.idea/.name           | 1 +
 PyCharm/.idea/PyCharm.iml     |10 ++++++++
 PyCharm/.idea/inspectionProfiles/profiles_settings.xml | 6 +++++
 PyCharm/.idea/misc.xml        | 4 ++++
 PyCharm/.idea/modules.xml     | 8 ++++++++
 PyCharm/.idea/vcs.xml         | 6 ++++++
 PyCharm/arithmetic.py         | 2 ++
 PyCharm/individual.py         | 4 ++++
 PyCharm/numbers.py           | 6 ++++++
 PyCharm/pov_sl_z.py           | 8 ++++++++
 PyCharm/user.py               | 4 ++++
12 files changed, 62 insertions(+)
create mode 100644 PyCharm/.idea/.gitignore
```

14. Отправьте сделанные изменения на сервер GitHub.

### ЗАДАЧА ПОВЫШЕННОЙ СЛОЖНОСТИ

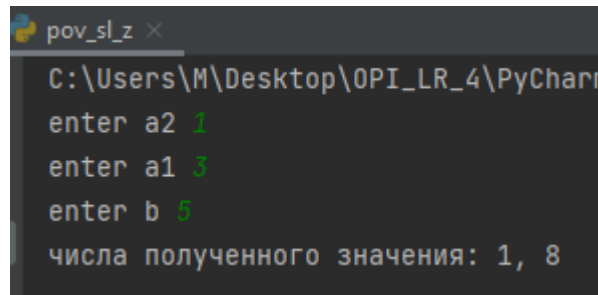
Даны цифры двух целых чисел: двузначного  $a_2a_1$  и однозначного  $b_1$ , где  $a_1$ – число единиц,  $a_2$ – число десятков. Получить цифры числа, равного сумме заданных чисел (известно, что это число двузначное). Слагаемое – двузначное число и число-результат не определять; условный оператор не использовать.

## Код программы

```
a2 = int(input('enter a2 '))
a1 = int(input('enter a1 '))
b = int(input('enter b '))

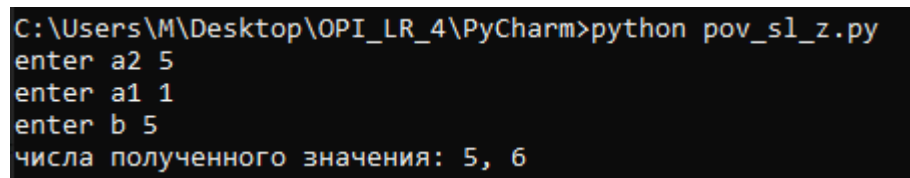
c2 = a2 + (a1+b)//10
c1 = (a1+b)%10

print("числа полученного значения: {0}, {1}".format(c2, c1))
```



```
pov_sl_z x
C:\Users\M\Desktop\OPI_LR_4\PyCharm
enter a2 1
enter a1 3
enter b 5
числа полученного значения: 1, 8
```

Рисунок 10– результат ее выполнения в IDE PyCharm



```
C:\Users\M\Desktop\OPI_LR_4\PyCharm>python pov_sl_z.py
enter a2 5
enter a1 1
enter b 5
числа полученного значения: 5, 6
```

Рисунок 11– результат ее выполнения в командной строке

## Вопросы для защиты работы

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

- Для установки интерпретатора Python первое, что нужно сделать – это скачать дистрибутив.
- Запустить скачанный установочный файл.
- Выбрать способ установки.
- Отметить необходимые опций установки (доступно при выборе Customize installation).
- Выберете место установки (доступно при выборе Customize installation).

При установки для Linux, в случае ошибки необходимо либо собрать Python из исходников либо взять из репозитория. Для установки из

репозитория в Ubuntu воспользуйтесь командой «`sudo apt-get install python3`»

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

Anaconda является дистрибутивом языков программирования таких как Python и R для науки о данных и машинного обучения, а Python — это язык программирования высокого уровня общего назначения.

Этот пакет 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?

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.

Создайте ноутбук для разработки, для этого нажмите на кнопку New (в правом углу окна) и в появившемся списке выберете Python. В результате будет создана новая страница в браузере с ноутбуком. Введите в первой ячейке команду «`print("Hello, World!")`» и нажмите Alt+Enter на клавиатуре. Ниже ячейки должна появиться соответствующая надпись.

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

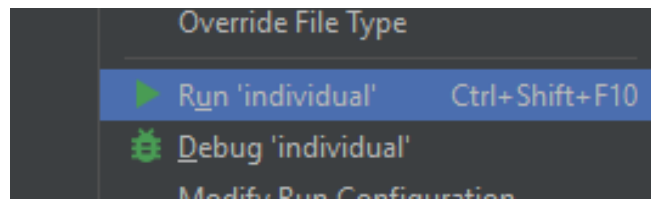
При создании нового проекта в PyCharm есть возможность выбрать путь до проекта и интерпретатор.

5. Как осуществить запуск программы с помощью IDE PyCharm?

Правой кнопкой в любом месте или по файлу слева и выбрать из



появившегося меню пункт «Run»



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

Интерактивный режим позволяет вводить команды, которые сразу же будут обрабатываться, это можно использовать в качестве калькулятора.

Пакетный режим позволяет запустить файл с исходным кодом.

7. Почему язык программирования Python называется языком динамической типизации?

Это означает, что одна и та же переменная в разное время может ссылаться на данные разного типа.

8. Какие существуют основные типы в языке программирования Python?

1. None (неопределенное значение переменной)

2. Логические переменные (Boolean Type)

3. Числа (Numeric Type)

3.1. int – целое число

3.2. float – число с плавающей точкой

### 3.3. complex – комплексное число

## 4. Списки (Sequence Type)

### 4.1. list – список

### 4.2. tuple – кортеж

### 4.3. range – диапазон

## 5. Строки (Text Sequence Type )

## 6. Бинарные списки (Binary Sequence Types)

### 6.1. bytes – байты

### 6.2. bytearray – массивы байт

6.3. memoryview – специальные объекты для доступа к внутренним данным объекта через protocol buffer

## 7. Множества (Set Types)

### 7.1. set – множество

### 7.2. frozenset – неизменяемое множество

## 8. Словари (Mapping Types)

### 8.1. dict – словарь

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

При создании переменной, вначале создается объект, который имеет уникальный идентификатор, тип и значение, после этого переменная может ссылаться на созданный объект.

Создание переменных и объектов в Python происходит с помощью оператора присваивания («=»). Записывается имя переменной, оператор и значение, например, «a = 5». Множественное присваивание (позиционное присваивание) в Python реализуется следующим образом: «a, b, c = 5, 3, 1»

#### 10. Как получить список ключевых слов в Python?

Нужно подключить модуль `keyword` и воспользоваться командой `keyword.kwlist`.

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

`id()` – возвращает уникальный идентификатор объекта в программе.

`type()` – возвращает тип переменной.

#### 12. Что такое изменяемые и неизменяемые типы в Python.

Изменяемый объект можно изменить после его создания, а неизменяемый – нет. Во втором случае фактически мы не изменяем значение переменной, а создаем другой объект с тем же именем и присваиваем ему другое значение. Мы связываем имя переменной с новым значением. Теперь, при ее вызове, он будет выводить новое значение и ссылаться на новое местоположение.

#### 13. Чем отличаются операции деления и целочисленного деления?

Во втором случае не учитывается остаток от деления.

#### 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`?

По аналогии с модулем `math` изучите самостоятельно назначение и основные функции модуля `cmath`.

Библиотека `math` содержит большое количество часто используемых математических функций, такие как округление, логарифмы, факториал, модуль числа, экспонента ( $e^x$ ), степень, квадратный корень, синус/косинус, числа  $\pi$  и  $e$  и т.д.

Модуль `cmath` – предоставляет функции для работы с комплексными числами. Из отличных функций можно выделить преобразование к полярным координатам и преобразование из полярных координат.

**`cmath.isfinite(x)`** - True, если действительная и мнимая части конечны.

**`cmath.isinf(x)`** - True, если либо действительная, либо мнимая часть бесконечна.

**`cmath.isnan(x)`** - True, если либо действительная, либо мнимая часть NaN.

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

`Sep()` устанавливает отличный от пробела разделитель строк.

`End()` указывает, что делать, после вывода строки (по умолчанию стоит переход на новую строку)

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()` позволяет форматировать выводимые строки.

Рассмотрим пример:

```
print("This is a {0}. It's {1}.".format("ball", "red"))
```

В строке в фигурных скобках указаны номера данных, которые будут сюда подставлены. Далее к строке применяется метод `format()`. В его скобках указываются сами данные (можно использовать переменные). На нулевое место подставится первый аргумент метода `format()`, на место с номером 1 – второй и т. д.

Форматирование также может осуществляться в старом (Си-) стиле. Он схож с тем, как происходит вывод на экран в языке C. Пример:

```
print("It's %s, %d. Level: %f" % (pupil, old, grade))
```

Здесь вместо трех комбинаций символов %s, %d, %f подставляются значения переменных pupil, old, grade. Буквы s, d, f обозначают типы данных

– строку, целое число, вещественное число.

F-Строки являются упрощенной версией метода format(). F-strings являются строковыми литералами с «f» в начале и фигурные скобки, содержащие выражения, которые в дальнейшем будут заменены своими значениями. Пример:

```
print(f"Hello, {name}. You are {2*8}.") //name – переменная
```

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

С помощью функции input() можно получить вводимые с консоли данные. Однако они будут принадлежать к строковому типу, чтоб получить число нужно использовать функции преобразования типов:

```
c = float(input("Enter temperature (Celsius) : "))
```