**Московский государственный технический университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра «Системы обработки информации и управления»

**Отчёт по Лабораторной работе №1**

**“Основы языка Python”**

Отчёт

(вид документа)

Листы А4

(вид носителя)

6

(количество листов)

|  |  |
| --- | --- |
| Исполнитель: | Студент группы ИУ5-54Б  Енин Александр Олегович |

Москва - 2020

**Цель работы**: Изучения основ языка Python.

**Описание задания:** разработать программу для решения биквадратного уравнения.

1. Программа должна быть разработана в виде консольного приложения на языке Python.
2. Программа осуществляет ввод с клавиатуры коэффициентов А, В, С, вычисляет дискриминант и корни уравнения (в зависимости от дискриминанта).
3. Если коэффициент А, В, С введен некорректно, то необходимо проигнорировать некорректное значение и ввести коэффициент повторно.
4. Первой строкой программа выводит ФИО разработчика и номер группы.
5. ДОПОЛНИТЕЛЬНОЕ ТРЕБОВАНИЕ. Коэффициенты А, В, С задаются в виде параметров командной строки. Если они не указаны, то вводятся с клавиатуры в соответствии с пунктом 2. Проверка из пункта 3 в этом случае производится для параметров командной строки без повторного ввода с клавиатуры.

**Текст программы:**

модуль main.py

**import** argparse

**import** funcs

#from LR1 import funcs # для запуска через терминал изменить на "import funcs"

print("\033[34mРешаем биквадратные уравнения \033[0m")

stop = 0

parser = argparse.ArgumentParser()

parser.add\_argument("--a", help = "Коэффициент А Биквадратного уравнения", default=**None**, type=float)

parser.add\_argument("--b", help = "Коэффициент B Биквадратного уравнения", default=**None**, type=float)

parser.add\_argument("--c", help = "Коэффициент C Биквадратного уравнения", default=**None**, type=float)

args = parser.parse\_args()

firstRunWithArgs = **False**

**while** stop == 0:

argsValues = funcs.checkArgs(args.a, args.b, args.c, firstRunWithArgs)

**if** argsValues[3] == **False**:

a = argsValues[0]

b = argsValues[1]

c = argsValues[2]

firstRunWithArgs = **True**

**else**:

a = float(input("\nВведите первый аргумент: "))

b = float(input("Введите второй аргумент: "))

c = float(input("Введите третий аргумент: "))

print(f"\n\033[33mУравнение:\033[0m {a}X^4 + {b}X^2 + {c} = 0\n")

stop = funcs.checkCoefficients(a, b, c)

**if** stop == 0:

stop = funcs.discriminant(a, b, c)

**if** stop == -1:

stop = 0

**elif** stop == 1:

choice = ""

**while** **True**:

choice = input("\nХотите завершить? (y/n) ")

**if** choice == "y":

**break**

**elif** choice == "n":

stop = 0

**break**

**else**:

print("\033[33mВведите правильный символ\033[0m \n")

модуль funcs.py

**def** checkCoefficients(a, b, c):

**if** (a, b, c) == (0, 0, 0):

print("\033[33mВведите ненулевые аргументы\033[0m")

**return** -1

**if** (a, b) == (0, 0):

print("\033[31mКорней нет!\033[0m")

**return** 1

**if** (a, c) == 0 **and** (b, c) == 0:

print("\033[32mx = 0\033[0m")

**return** 1

**if** a == 0:

**if** c < 0:

root = pow(-c/b, 0.5)

print(f"\033[32mКорни: x1 = {root}, x2 = {-root}\033[0m")

**return** 1

**elif** c == 0:

print("\033[32mx = 0\033[0m")

**return** 1

**else**:

print("\033[31mКорней нет!\033[0m")

**return** 1

**if** b == 0:

**if** c < 0:

root = pow(pow(-c/b, 0.5), 0.5)

print(f"\033[32mКорни: x1 = {root}, x2 = {-root}\033[0m")

**return** 1

**elif** c == 0:

print("\033[32mx = 0\033[0m")

**return** 1

**else**:

print("\033[31mКорней нет!\033[0m")

**return** 1

**if** c == 0:

print("\033[33mПопробуйте другие коэфициенты...\033[0m")

**return** -1

**return** 0

**def** discriminant(a, b, c):

D = pow(b, 2) - (4 \* a \* c)

**if** D > 0:

# Производим замену t = x^2

t1 = (-b + pow(D, 0.5)) / (2 \* a)

t2 = (-b - pow(D, 0.5)) / (2 \* a)

**if** t1 < 0 **and** t2 < 0:

print("\033[31mКорней нет\033[0m")

**return** 1

**elif** t1 < 0 **and** t2 > 0:

root = pow(t2, 0.5)

print(f"\033[32mКорни: x1 = {root}, x2 = {-root}\033[0m")

**return** 1

**elif** t1 > 0 **and** t2 < 0:

root = pow(t1, 0.5)

print(f"\033[32mКорни: x1 = {root}, x2 = {-root}\033[0m")

**return** 1

**else**:

root1 = pow(t1, 0.5)

root2 = pow(t2, 0.5)

print(f"\033[32mКорни: x1 = {root1}, x2 = {-root1}\033[0m")

print(f"\033[32mКорни: x3 = {root2}, x4 = {-root2}\033[0m")

**return** 1

**elif** D == 0:

t = -b / (2 \* a)

**if** t < 0:

print("\033[31mКорней нет!\033[0m")

**return** 1

**else**:

root = pow(t, 0.5)

print(f"\033[32mКорни: x1 = {root}, x4 = {-root}\033[0m")

**return** 1

**else**:

print("\033[31mДействительных корней нет!\033[0m")

**return** 1

**def** checkArgs(a, b, c, firstRunWithArgs):

**if** a == **None** **and** b == **None** **and** c == **None**:

firstRunWithArgs = **True**

**elif** a == **None** **and** b == **None**:

a = 0; b = 0;

**elif** a == **None** **and** c == **None**:

a = 0; c= 0;

**elif** b == **None** **and** c == **None**:

b = 0; c = 0;

**elif** a == **None**:

a = 0

**elif** b == **None**:

b = 0

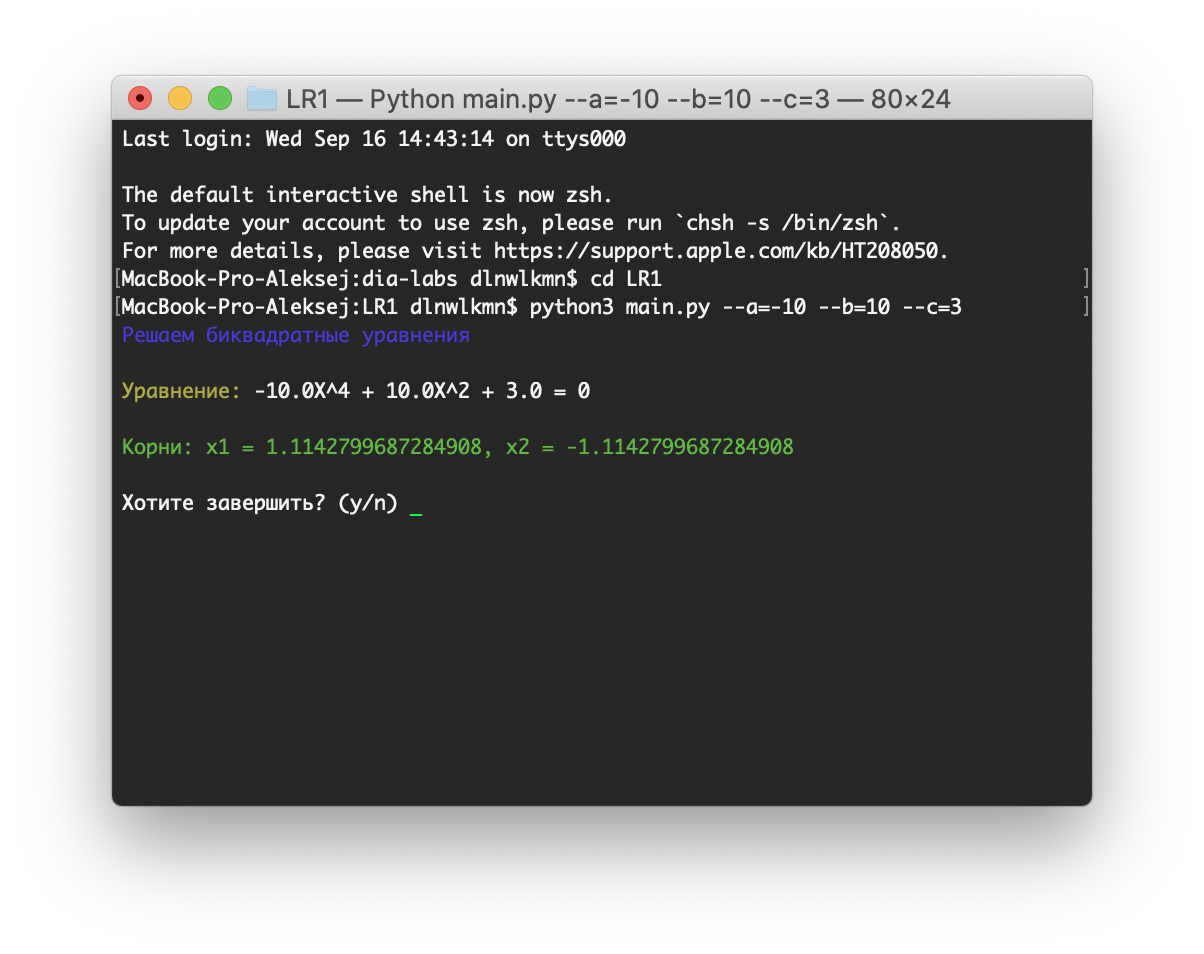
**elif** c == **None**:

c = 0

**return** (a, b, c, firstRunWithArgs)

**Окно терминала:**

Запуск программы с аргументами из командной строки



После выполнения расчета и вывода ответа на экран, программа предлагает завершить работу, если вводить n, то программа запросит ввод символов с клавиатуры, если ввести y, то программа прекратит свою работу;  
 Если ввести какой либо другой символ то программа это обработает и попросит ввести правильный.



Реализована обработка нулевых коэффициентов уравнения, предлагается ввести не нулевые.

