**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №1

«Основные конструкции языка python»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-32Б |  | преподаватель каф. ИУ5 |
| Турдиев Ихтияр |  | Гапанюк Юрий Евгеньевич |
|  |  |  |
|  |  |  |

Москва, 2024 г.

Постановка задачи

Задание.

Разработать программу для решения биквадратного уравнения.

Программа должна быть разработана в виде консольного приложения на языке Python.

Программа осуществляет ввод с клавиатуры коэффициентов А, В, С, вычисляет дискриминант и ДЕЙСТВИТЕЛЬНЫЕ корни уравнения (в зависимости от дискриминанта).

Коэффициенты А, В, С могут быть заданы в виде параметров командной строки ( вариант задания параметров приведен в конце файла с примером кода ). Если они не заданы, то вводятся с клавиатуры в соответствии с пунктом 2. Описание работы с параметрами командной строки.

Если коэффициент А, В, С введен или задан в командной строке некорректно, то необходимо проигнорировать некорректное значение и вводить коэффициент повторно пока коэффициент не будет введен корректно. Корректно заданный коэффициент - это коэффициент, значение которого может быть без ошибок преобразовано в действительное число.

Дополнительное задание 1 (\*). Разработайте две программы на языке Python - одну с применением процедурной парадигмы, а другую с применением объектно-ориентированной парадигмы.

Дополнительное задание 2 (\*). Разработайте две программы - одну на языке Python, а другую на любом другом языке программирования (кроме С++).

Текст программы

Main.py

from math import sqrt  
from sys import argv  
  
  
def get\_coef(ind: int, promt: str) -> float:  
 try:  
 coef = argv[ind]  
 except IndexError:  
 coef = input(promt)  
  
 while True:  
 try:  
 return float(coef)  
 except ValueError:  
 coef = input(promt)  
  
  
def get\_result(a: float, b: float, c: float) -> set:  
  
 result = set()  
 try:  
 sdisc = sqrt(b \* b - 4 \* a \* c)  
  
 x1 = (-b + sdisc) / (2 \* a)  
 if x1 >= 0:  
 result.add(sqrt(x1))  
 result.add(-sqrt(x1))  
  
 x2 = (-b - sdisc) / (2 \* a)  
 result.add(sqrt(x2))  
 result.add(-sqrt(x2))  
 return result  
 except ValueError or TypeError:  
 if len(result) == 0:  
 raise RuntimeWarning  
 return result  
  
  
def main():  
 a = get\_coef(1, "Enter coefficient a: ")  
 b = get\_coef(2, "Enter coefficient b: ")  
 c = get\_coef(3, "Enter coefficient c: ")  
 result = get\_result(a, b, c)  
 if result:  
 print(result)  
 else:  
 print("No roots found")  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()

main\_oop.py

from math import sqrt

from sys import argv

class BiSquareCalculator:

def \_\_init\_\_(self):

self.roots = set()

self.a\_coef = 0.0

self.b\_coef = 0.0

self.c\_coef = 0.0

def get\_coef(self, ind, promt):

try:

coef = argv[ind]

except IndexError:

coef = input(promt)

while True:

try:

return float(coef)

except ValueError:

coef = input(promt)

def get\_coefs(self):

self.a\_coef = self.get\_coef(1, "Enter coef a: ")

self.b\_coef = self.get\_coef(2, "Enter coef b: ")

self.c\_coef = self.get\_coef(3, "Enter coef c: ")

def find\_roots(self):

try:

sdisc = sqrt(self.b\_coef \* self.b\_coef - 4 \* self.a\_coef \* self.c\_coef)

x1 = (-self.b\_coef + sdisc) / (2 \* self.a\_coef)

if x1 > 0:

self.roots.add(sqrt(x1))

self.roots.add(-sqrt(x1))

x2 = (-self.b\_coef - sdisc) / (2 \* self.a\_coef)

self.roots.add(sqrt(x2))

self.roots.add(-sqrt(x2))

except ValueError or TypeError:

return

def print\_roots(self):

if self.roots:

print(self.roots)

else:

print("No roots")

def main():

calc = BiSquareCalculator()

calc.get\_coefs()

calc.find\_roots()

calc.print\_roots()

return 0

if \_\_name\_\_ == "\_\_main\_\_":

main()

main.feature

Feature: Biquadratic equation solving  
 As a user I want to solve equation  
  
 Scenario: User gets correct ans with 4 roots  
 When User passes arguments "1 -13 36"  
 Then Result is "3 -3 2 -2  
  
 Scenario: User gets correct ans with 3 ans  
 When User passes arguments "-4 36 0"  
 Then Result is "0 3.0 -3.0  
  
 Scenario: User gets correct ans with 2 ans  
 When User passes arguments "1 -50 625"  
 Then Result is "5.0 -5.0  
  
 Scenario: User gets correct ans with 1 ans  
 When User passes arguments "170 0 0"  
 Then Result is "0

Lab1\_bdd.py

from behave import when, then  
from main import get\_result  
  
  
@when('User passes arguments "{args}"')  
def step\_impl(context, args):  
 coef\_list = [int(i) for i in args.split()]  
 context.response = get\_result(coef\_list[0], coef\_list[1], coef\_list[2])  
  
  
@then('Result is "{result}')  
def step\_impl(context, result):  
 assert ({float(i) for i in result.split()} == context.response)

Обработка результатов 