**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 ИУ5-33Б |  | преподаватель каф. ИУ5 |
| Беляев Иван |  | Нардид А.Н. |
| Подпись и дата: |  | Подпись и дата: |

**Задание:**

Разработать программу для решения [биквадратного уравнения.](https://ru.wikipedia.org/wiki/%D0%A3%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5_%D1%87%D0%B5%D1%82%D0%B2%D1%91%D1%80%D1%82%D0%BE%D0%B9_%D1%81%D1%82%D0%B5%D0%BF%D0%B5%D0%BD%D0%B8#%D0%91%D0%B8%D0%BA%D0%B2%D0%B0%D0%B4%D1%80%D0%B0%D1%82%D0%BD%D0%BE%D0%B5_%D1%83%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5)

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 ДЕЙСТВИТЕЛЬНЫЕ корни уравнения (в зависимости от дискриминанта).
3. Коэффициенты А, В, С могут быть заданы в виде параметров командной строки ( [вариант задания параметров приведен в конце файла с примером кода](https://github.com/ugapanyuk/BKIT_2022/blob/main/code/lab1_code) ). Если они не заданы, то вводятся с клавиатуры в соответствии с пунктом 2. [Описание работы с параметрами командной строки.](https://realpython.com/python-command-line-arguments/#the-command-line-interface)
4. 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.
5. Дополнительное задание 1 (\*). Разработайте две программы на языке Python - одну с применением процедурной парадигмы, а другую с применением объектно-ориентированной парадигмы.
6. Дополнительное задание 2 (\*). Разработайте две программы - одну на языке Python, а другую на любом другом языке программирования (кроме С++).

**Программа с применением процедурной парадигмы:**

**Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описаниеИзображение выглядит как текст, снимок экрана, Шрифт, документ

Автоматически созданное описание**

import sys

def get\_coef(index, prompt):

'''

Читаем коэффициент из командной строки или вводим с клавиатуры

Args:

index (int): Номер параметра в командной строке

prompt (str): Приглашение для ввода коэффицента

Returns:

float: Коэффициент квадратного уравнения

'''

while True:

try:

# Пробуем прочитать коэффициент из командной строки

coef\_str = sys.argv[index]

except:

# Вводим с клавиатуры

print(prompt)

coef\_str = input()

try:

# Переводим строку в действительное число

coef = float(coef\_str)

return coef

except ValueError:

print("Ошибка: Введите корректное числовое значение коэффициента.")

def solve\_quadratic(a, b, c):

# Вычисляем дискриминант

discriminant = b\*\*2 - 4\*a\*c

if discriminant > 0:

# Два действительных корня

x1 = (-b + discriminant\*\*0.5) / (2\*a)

x2 = (-b - discriminant\*\*0.5) / (2\*a)

q = []

if x1 > 0:

x11 = x1 \*\* 0.5

x12 = -x1 \*\* 0.5

q.append(x11)

q.append(x12)

elif x1 == 0:

q.append(0)

if x2 > 0:

x21 = x2 \*\* 0.5

x22 = -x2 \*\* 0.5

q.append(x21)

q.append(x22)

elif x2 == 0:

q.append(0)

return q

elif discriminant == 0:

q = []

# Один действительный корень

x = -b / (2\*a)

if x > 0:

x1 = x \*\* 0.5

x2 = -x \*\* 0.5

q.append(x1)

q.append(x2)

elif x == 0:

q.append(0)

return q

else:

q = []

return q

def main():

# Получаем коэффициенты A, B, C из командной строки или с клавиатуры

a = get\_coef(1, "Введите коэффициент A: ")

while a == 0:

a = get\_coef(1, "Это не квадратное уравнение! \nВведите еще раз коэффициент А:")

b = get\_coef(2, "Введите коэффициент B: ")

c = get\_coef(3, "Введите коэффициент C: ")

solutions = solve\_quadratic(a, b, c)

if len(solutions) == 0:

print("Корней нет")

else:

print("Корни: ", end='')

for i in solutions:

print(i, end=", ")

# Если сценарий запущен из командной строки

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

main()

**Результаты работы программы**

**Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание**

**Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание**

**Программа с применением ООП:**

**Изображение выглядит как текст, снимок экрана, документ, Шрифт

Автоматически созданное описание**

**Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание**

**Изображение выглядит как текст, снимок экрана, Шрифт, документ

Автоматически созданное описание**

**Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание**

import sys

import math

class SquareRoots:

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

'''

Конструктор класса

'''

# Объявление коэффициентов

self.coef\_A = 0.0

self.coef\_B = 0.0

self.coef\_C = 0.0

# Количество корней

self.num\_roots = 0

# Список корней

self.roots\_list = []

def get\_coef(self, index, prompt):

'''

Читаем коэффициент из командной строки или вводим с клавиатуры

Args:

index (int): Номер параметра в командной строке

prompt (str): Приглашение для ввода коэффицента

Returns:

float: Коэффициент квадратного уравнения

'''

while True:

try:

# Пробуем прочитать коэффициент из командной строки

coef\_str = sys.argv[index]

except:

# Вводим с клавиатуры

print(prompt)

coef\_str = input()

try:

# Переводим строку в действительное число

coef = float(coef\_str)

return coef

except ValueError:

print("Ошибка: Введите корректное числовое значение коэффициента.")

def get\_coefs(self):

'''

Чтение трех коэффициентов

'''

self.coef\_A = self.get\_coef(1, 'Введите коэффициент А:')

while self.coef\_A == 0:

self.coef\_A = self.get\_coef(1, "Это не квадратное уравнение! \nВведите еще раз коэффициент А:")

self.coef\_B = self.get\_coef(2, 'Введите коэффициент B:')

self.coef\_C = self.get\_coef(3, 'Введите коэффициент C:') def calculate\_roots(self):

'''

Вычисление корней квадратного уравнения

'''

a = self.coef\_A

b = self.coef\_B

c = self.coef\_C

# Вычисление дискриминанта и корней

D = b\*b - 4\*a\*c

if D > 0:

# Два действительных корня

x1 = (-b + D\*\*0.5) / (2\*a)

x2 = (-b - D\*\*0.5) / (2\*a)

self.roots\_list = []

if x1 > 0:

x11 = x1 \*\* 0.5

x12 = -x1 \*\* 0.5

self.roots\_list.append(x11)

self.roots\_list.append(x12)

elif x1 == 0:

self.roots\_list.append(0)

if x2 > 0:

x21 = x2 \*\* 0.5

x22 = -x2 \*\* 0.5

self.roots\_list.append(x21)

self.roots\_list.append(x22)

elif x2 == 0:

self.roots\_list.append(0)

return self.roots\_list

elif D == 0:

self.roots\_list = []

# Один действительный корень

x = -b / (2\*a)

if x > 0:

x1 = x \*\* 0.5

x2 = -x \*\* 0.5

self.roots\_list.append(x1)

self.roots\_list.append(x2)

elif x == 0:

self.roots\_list.append(0)

return self.roots\_list

else:

self.roots\_list = []

return self.roots\_list

def print\_roots(self):

if len(self.roots\_list) == 0:

print('Нет корней')

else:

print("Корни: ", end='')

for i in self.roots\_list:

print(i, end=", ")

def main():

'''

Основная функция

'''

# Создание объекта класса

r = SquareRoots()

# Последовательный вызов необходимых методов

r.get\_coefs()

r.calculate\_roots()

r.print\_roots()

# Если сценарий запущен из командной строки

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

main()

# Пример запуска

# roots\_oop.py 1 0 -4

**Результаты работы программы**

**Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание**

**Программа с применением процедурной парадигмы на другом языке (Паскаль):**

program zadacha;

uses

SysUtils;

function get\_coef(index: Integer; prompt: string): Real;

var

coef\_str: string;

coef: Real;

begin

while True do

begin

try

coef\_str := ParamStr(index);

except

WriteLn(prompt);

ReadLn(coef\_str);

end;

try

coef := StrToFloat(coef\_str);

Result := coef;

Exit;

except

WriteLn('Ошибка: Введите корректное числовое значение коэффициента.');

end;

end;

end;

function solve\_quadratic(a, b, c: Real): array of Real;

var

discriminant: Real;

x1, x2: Real;

q: array of Real;

begin

discriminant := b\*b - 4\*a\*c;

if discriminant > 0 then

begin

x1 := (-b + Sqrt(discriminant)) / (2\*a);

x2 := (-b - Sqrt(discriminant)) / (2\*a);

SetLength(q, 0);

if x1 > 0 then

begin

q := [Sqrt(x1), -Sqrt(x1)];

end

else if x1 = 0 then

begin

q := [0];

end;

if x2 > 0 then

begin

q := q + [Sqrt(x2), -Sqrt(x2)];

end

else if x2 = 0 then

begin

q := q + [0];

end;

Result := q;

end

else if discriminant = 0 then

begin

SetLength(q, 0);

x := -b / (2\*a);

if x > 0 then

begin

q := [Sqrt(x), -Sqrt(x)];

end

else if x = 0 then

begin

q := [0];

end;

Result := q;

end

else

begin

SetLength(q, 0);

Result := q;

end;

end;

procedure Main;

var

a, b, c: Real;

solutions: array of Real;

i: Integer;

begin

a := get\_coef(1, 'Введите коэффициент A: ');

while a = 0 do

begin

a := get\_coef(1, 'Это не квадратное уравнение! \nВведите еще раз коэффициент А:');

end;

b := get\_coef(2, 'Введите коэффициент B: ');

c := get\_coef(3, 'Введите коэффициент C: ');

solutions := solve\_quadratic(a, b, c);

if Length(solutions) = 0 then

begin

WriteLn('Корней нет');

end

else

begin

Write('Корни: ');

for i := 0 to Length(solutions) - 1 do

begin

Write(solutions[i], ' ');

end;

WriteLn;

end;

end;

begin

Main;

end.