**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**](https://github.com/ugapanyuk/BKIT_2022/wiki/lab_python_intro)**»**

|  |  |  |
| --- | --- | --- |
| **Выполнил:** |  | **Проверил:** |
| студент группы ИУ5-31Б |  | преподаватель каф. ИУ5 |
| Санников Н.А. |  |  |
|  |  |  |

Москва, 2024 г.

**Описание задания**

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.

**Текст программы**

**#решение задачи с применением объектно-ориентированной парадигмы (lab1(oop).py)**

import sys

import math

class BiquadraticEquation:

def \_\_init\_\_(self, a, b, c): #конструктор класса

self.a = a

self.b = b

self.c = c

def solve(self):

if self.a == 0:

raise ValueError("Коэффициент A не должен быть равен 0.")

print(f"Решаем уравнение {self.a}x^4 + {self.b}x^2 + {self.c} = 0")

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

print(f"Дискриминант: {discriminant}")

if discriminant > 0:

z1 = (-self.b + math.sqrt(discriminant)) / (2 \* self.a)

z2 = (-self.b - math.sqrt(discriminant)) / (2 \* self.a)

roots = []

if z1 >= 0:

roots.append(math.sqrt(z1))

roots.append(-math.sqrt(z1))

if z2 >= 0:

roots.append(math.sqrt(z2))

roots.append(-math.sqrt(z2))

if roots:

print(f"Действительные корни: {sorted(set(roots))}")

else:

print("Нет действительных корней.")

elif discriminant == 0:

z = -self.b / (2 \* self.a)

if z >= 0:

print(f"Действительные корни: {math.sqrt(z)}, {-math.sqrt(z)}")

else:

print("Нет действительных корней.")

else:

print("Нет действительных корней.")

def get\_coefficient(name):

while True:

try:

value = float(input(f"Введите коэффициент {name}: "))

return value

except ValueError:

print(f"Коэффициент {name} должен быть числом. Попробуйте снова.")

if len(sys.argv) == 4:

try:

a = float(sys.argv[1])

b = float(sys.argv[2])

c = float(sys.argv[3])

except ValueError:

print("Один или несколько параметров командной строки некорректны.")

a = get\_coefficient('A')

b = get\_coefficient('B')

c = get\_coefficient('C')

else:

a = get\_coefficient('A')

b = get\_coefficient('B')

c = get\_coefficient('C')

equation = BiquadraticEquation(a, b, c)

equation.solve()

**#решение задачи с применением процедурной парадигмы (lab1(procedural).py)**

import sys

import math

def get\_coefficient(name):

while True:

try:

value = float(input(f"Введите коэффициент {name}: "))

return value

except ValueError:

print(f"Коэффициент {name} должен быть числом. Попробуйте снова.")

def solve\_biquadratic\_equation(a, b, c):

if a == 0:

print("Коэффициент A не должен быть равен 0.")

return

print(f"Решаем уравнение {a}x^4 + {b}x^2 + {c} = 0")

# Приведем его к квадратному уравнению относительно z = x^2

# Получаем a\*z^2 + b\*z + c = 0

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

print(f"Дискриминант: {discriminant}")

if discriminant > 0:

z1 = (-b + math.sqrt(discriminant)) / (2 \* a)

z2 = (-b - math.sqrt(discriminant)) / (2 \* a)

roots = []

if z1 >= 0:

roots.append(math.sqrt(z1))

roots.append(-math.sqrt(z1))

if z2 >= 0:

roots.append(math.sqrt(z2))

roots.append(-math.sqrt(z2))

if roots:

print(f"Действительные корни: {sorted(set(roots))}")

else:

print("Нет действительных корней.")

elif discriminant == 0:

z = -b / (2 \* a)

if z >= 0:

print(f"Действительные корни: {math.sqrt(z)}, {-math.sqrt(z)}")

else:

print("Нет действительных корней.")

else:

print("Нет действительных корней.")

if len(sys.argv) == 4:

try:

a = float(sys.argv[1])

b = float(sys.argv[2])

c = float(sys.argv[3])

except ValueError:

print("Один или несколько параметров командной строки некорректны.")

a = get\_coefficient('A')

b = get\_coefficient('B')

c = get\_coefficient('C')

else:

a = get\_coefficient('A')

b = get\_coefficient('B')

c = get\_coefficient('C')

solve\_biquadratic\_equation(a, b, c)

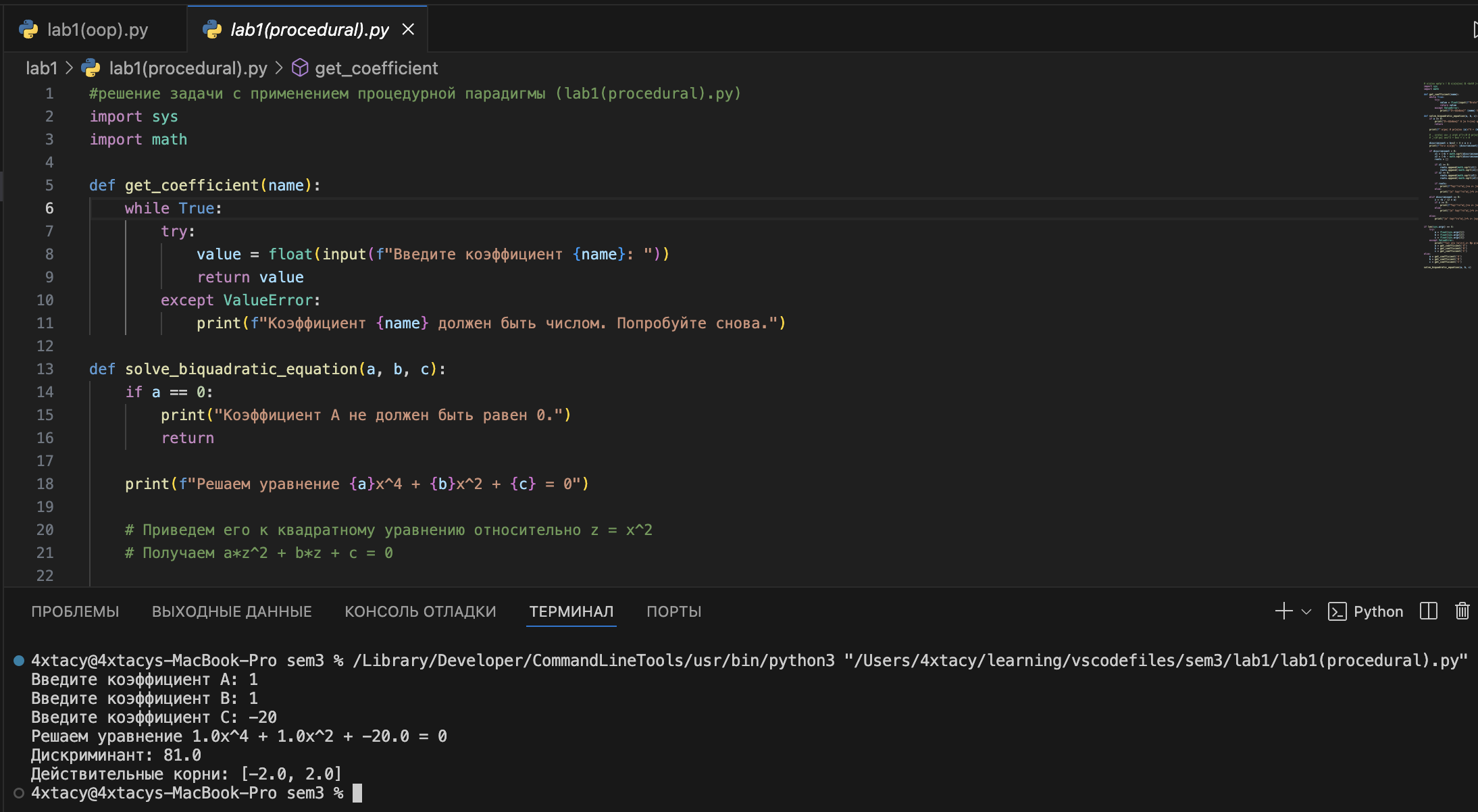
**Экранные формы**

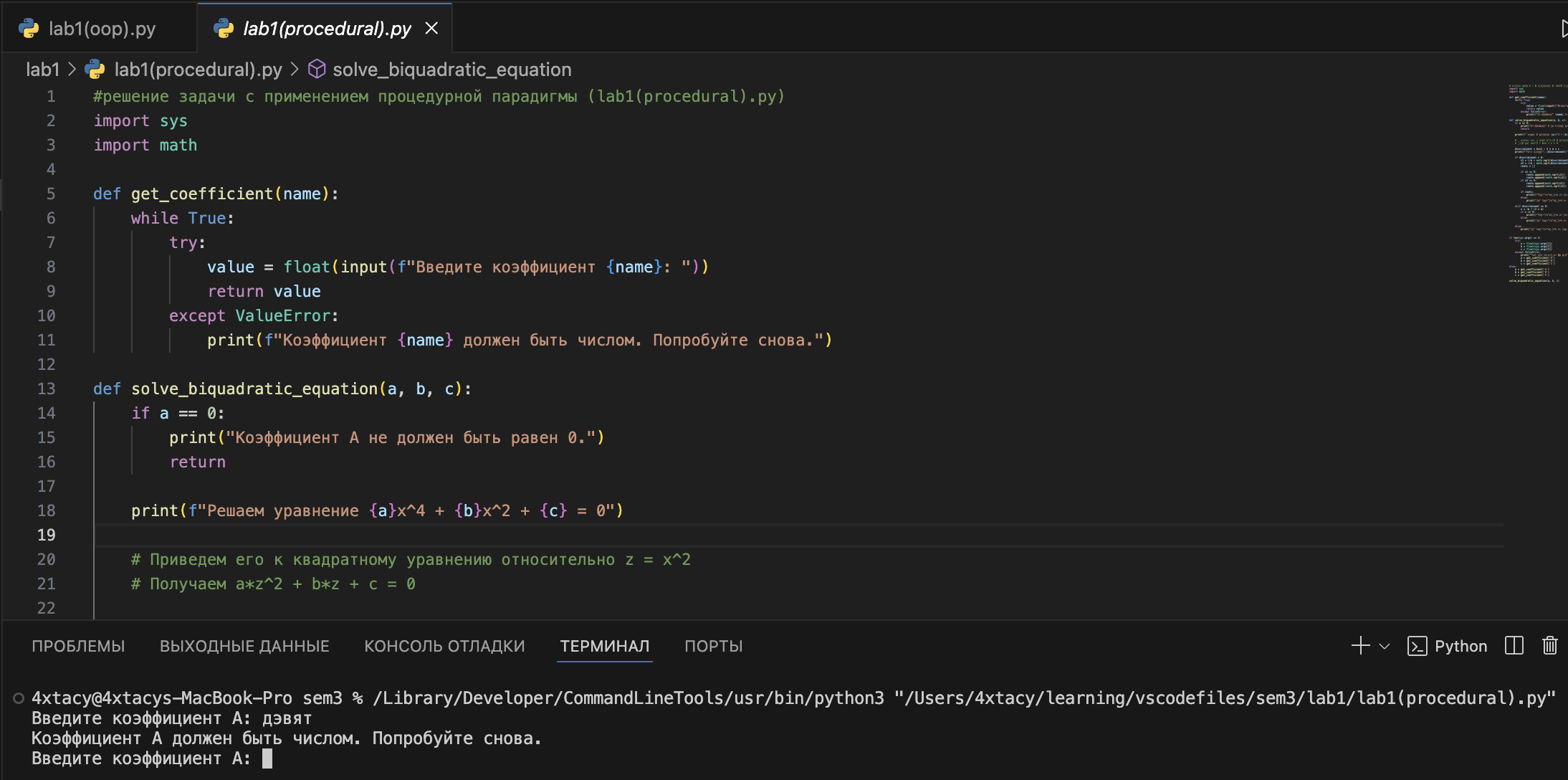
**с примерами выполнения программы**

*решение задачи с применением объектно-ориентированной парадигмы***Изображение выглядит как текст, программное обеспечение, Мультимедийное программное обеспечение, Графическое программное обеспечение

Автоматически созданное описание**

*решение задачи с применением процедурной парадигмы*

 Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание 

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание