МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
им. Н.Э. Баумана

Факультет “Информатика и системы управления”  
Кафедра “Системы обработки информации и управления”

A logo with a gold star and blue ribbon

Description automatically generated

Дисциплина “Парадигмы и конструкции языков программирования”

Отчет по лабораторной работе №1  
“Основные конструкции языка Python”

**Выполнил:**Студент группы ИУ5-35Б  
Ким А. А.  
**Преподаватель:**Нардид А.Н.

Москва 2025

**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++).

**Листинг кода**

**main\_oop.py:**

import sys  
  
class EquationSolver:  
  
 def \_\_init\_\_(self, a=0, b=0, c=0):  
 self.a = a  
 self.b = b  
 self.c = c  
 self.roots = []  
  
 @property  
 def a(self):  
 return self.\_a  
  
 @a.setter  
 def a(self, value):  
 if not isinstance(value, (int, float)):  
 raise ValueError("Коэффициент A должен быть числом")  
 self.\_a = value  
  
 @property  
 def b(self):  
 return self.\_b  
  
 @b.setter  
 def b(self, value):  
 if not isinstance(value, (int, float)):  
 raise ValueError("Коэффициент B должен быть числом")  
 self.\_b = value  
  
 @property  
 def c(self):  
 return self.\_c  
  
 @c.setter  
 def c(self, value):  
 if not isinstance(value, (int, float)):  
 raise ValueError("Коэффициент C должен быть числом")  
 self.\_c = value  
  
 def solve\_linear\_case(self):  
 if self.c == 0:  
 return ["Все действительные числа"]  
 else:  
 return []  
  
 def solve\_quadratic\_case(self):  
 roots = []  
 x\_squared = -self.c / self.b  
  
 if x\_squared > 0:  
 x1 = x\_squared \*\* 0.5  
 x2 = -x1  
 roots.extend([x1, x2])  
 elif x\_squared == 0:  
 roots.append(0.0)  
  
 return roots  
  
 def solve\_biquadratic\_case(self):  
 roots = []  
 discriminant = self.b \*\* 2 - 4 \* self.a \* self.c  
  
 if discriminant < 0:  
 return []  
 elif discriminant == 0:  
 y = -self.b / (2 \* self.a)  
 if y > 0:  
 x1 = y \*\* 0.5  
 x2 = -x1  
 roots.extend([x1, x2])  
 elif y == 0:  
 roots.append(0.0)  
 else:  
 y1 = (-self.b + discriminant \*\* 0.5) / (2 \* self.a)  
 y2 = (-self.b - discriminant \*\* 0.5) / (2 \* self.a)  
  
 for y in [y1, y2]:  
 if y > 0:  
 x1 = y \*\* 0.5  
 x2 = -x1  
 roots.extend([x1, x2])  
 elif y == 0:  
 roots.append(0.0)  
  
 return sorted(list(set(roots)))  
  
 def solve(self):  
 self.roots = []  
  
 if self.a == 0:  
 if self.b == 0:  
 self.roots = self.solve\_linear\_case()  
 else:  
 self.roots = self.solve\_quadratic\_case()  
 else:  
 self.roots = self.solve\_biquadratic\_case()  
  
 return self.roots  
  
 def display\_solution(self):  
 print(f"\nУравнение: {self.a}x⁴ + {self.b}x² + {self.c} = 0")  
  
 if self.a == 0:  
 if self.b == 0:  
 print("Уравнение линейное")  
 else:  
 print("Уравнение квадратное (A = 0)")  
 else:  
 discriminant = self.b \*\* 2 - 4 \* self.a \* self.c  
 print(f"Дискриминант: {discriminant}")  
  
 if not self.roots:  
 print("Действительных корней нет")  
 elif self.roots == ["Все действительные числа"]:  
 print("Уравнение превращается в тождество: 0 = 0")  
 print("Решение: все действительные числа")  
 else:  
 roots\_str = ", ".join([f"x{i + 1} = {root:.4f}" for i, root in enumerate(self.roots)])  
 print(f"Корни: {roots\_str}")  
  
  
class InputHandler:  
  
 @staticmethod  
 def is\_valid\_float(value):  
 try:  
 float(value)  
 return True  
 except (ValueError, TypeError):  
 return False  
  
 @staticmethod  
 def get\_coefficient(prompt, param\_value=None):  
 while True:  
 if param\_value is not None and InputHandler.is\_valid\_float(param\_value):  
 value = float(param\_value)  
 print(f"{prompt} = {value}")  
 return value  
 else:  
 user\_input = input(prompt)  
 if InputHandler.is\_valid\_float(user\_input):  
 return float(user\_input)  
 print("Ошибка: введите действительное число")  
  
def main():  
 print("Решение уравнения Ax⁴ + Bx² + C = 0")  
  
 # Получаем коэффициенты из параметров командной строки или с клавиатуры  
 a\_param = sys.argv[1] if len(sys.argv) > 1 else None  
 b\_param = sys.argv[2] if len(sys.argv) > 2 else None  
 c\_param = sys.argv[3] if len(sys.argv) > 3 else None  
  
 a = InputHandler.get\_coefficient("Введите коэффициент A: ", a\_param)  
 b = InputHandler.get\_coefficient("Введите коэффициент B: ", b\_param)  
 c = InputHandler.get\_coefficient("Введите коэффициент C: ", c\_param)  
  
 try:  
 equation = EquationSolver(a, b, c)  
 equation.solve()  
 equation.display\_solution()  
 except ValueError as e:  
 print(f"Ошибка: {e}")  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

**main\_procedur.py:**

import sys  
  
def is\_valid\_float(value):  
 try:  
 float(value)  
 return True  
 except (ValueError, TypeError):  
 return False  
  
def get\_coefficient(prompt, param\_value=None):  
 while True:  
 if param\_value is not None and is\_valid\_float(param\_value):  
 value = float(param\_value)  
 print(f"{prompt} = {value}")  
 return value  
 else:  
 user\_input = input(prompt)  
 if is\_valid\_float(user\_input):  
 return float(user\_input)  
 print("Ошибка: введите действительное число")  
  
def solve\_equation(a, b, c):  
 print(f"\nУравнение: {a}x⁴ + {b}x² + {c} = 0")  
  
 roots = []  
  
 if a == 0:  
 if b == 0:  
 if c == 0:  
 print("Уравнение превращается в тождество: 0 = 0")  
 return ["Все действительные числа"]  
 else:  
 print(f"Уравнение не имеет решений: {c} = 0")  
 return []  
 else:  
 print("Уравнение квадратное (A = 0)")  
 x\_squared = -c / b  
  
 if x\_squared > 0:  
 x1 = x\_squared \*\* 0.5  
 x2 = -x1  
 roots.extend([x1, x2])  
 print(f"Корни: x₁ = {x1:.4f}, x₂ = {x2:.4f}")  
 elif x\_squared == 0:  
 roots.append(0.0)  
 print(f"Корень: x = 0.0")  
 else:  
 print("Действительных корней нет")  
 return roots  
 else:  
 discriminant = b \*\* 2 - 4 \* a \* c  
 print(f"Дискриминант: {discriminant}")  
  
 if discriminant < 0:  
 print("Действительных корней нет")  
 elif discriminant == 0:  
 y = -b / (2 \* a)  
 if y > 0:  
 x1 = y \*\* 0.5  
 x2 = -x1  
 roots.extend([x1, x2])  
 print(f"Корни: x₁ = {x1:.4f}, x₂ = {x2:.4f}")  
 elif y == 0:  
 roots.append(0.0)  
 print(f"Корень: x = 0.0")  
 else:  
 print("Действительных корней нет")  
 else: # y=x^2  
 y1 = (-b + discriminant \*\* 0.5) / (2 \* a)  
 y2 = (-b - discriminant \*\* 0.5) / (2 \* a)  
  
 for y in [y1, y2]:  
 if y > 0:  
 x1 = y \*\* 0.5  
 x2 = -x1  
 roots.extend([x1, x2])  
 elif y == 0:  
 roots.append(0.0)  
  
 roots = sorted(list(set(roots)))  
  
 if roots:  
 roots\_str = ", ".join([f"x{i + 1} = {root:.4f}" for i, root in enumerate(roots)])  
 print(f"Корни: {roots\_str}")  
 else:  
 print("Действительных корней нет")  
  
 return roots  
  
def main():  
 print("Решение уравнения Ax⁴ + Bx² + C = 0")  
  
 a\_param = sys.argv[1] if len(sys.argv) > 1 else None  
 b\_param = sys.argv[2] if len(sys.argv) > 2 else None  
 c\_param = sys.argv[3] if len(sys.argv) > 3 else None  
  
 a = get\_coefficient("Введите коэффициент A: ", a\_param)  
 b = get\_coefficient("Введите коэффициент B: ", b\_param)  
 c = get\_coefficient("Введите коэффициент C: ", c\_param)  
  
 solve\_equation(a, b, c)  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

**Скриншоты работы программ**

 

 