



ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Системы обработки информации и управления» (ИУ-5)

## ЛАБОРАТОРНАЯ РАБОТА

№ 1

Основные конструкции языка Python

Группа ИУ5-35Б

Студент Мешков 16.12.2024 /Д.Е. Мушкарин /

(Подпись, дата)

(И.О.Фамилия)

Преподаватель \_\_\_\_\_ /Ю. Е. Гапанюк/

(Подпись, дата)

(И.О.Фамилия)

2024

## Задание:

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  $A$ ,  $B$ ,  $C$ , вычисляет дискриминант и **ДЕЙСТВИТЕЛЬНЫЕ** корни уравнения (в зависимости от дискриминанта).
3. Коэффициенты  $A$ ,  $B$ ,  $C$  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.
4. Если коэффициент  $A$ ,  $B$ ,  $C$  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 C++).

## Текст программы (Python):

### def.py

```
import math
from typing import List
import sys

def SolveQuadratic(Equation: List) -> List:

    a, b, c = map(float, Equation)
    D = math.pow(b, 2) - 4 * a * c

    if (D < 0):
        solutions = ()
    if (D == 0):
        solutions = -b / (2*a)
    if (D > 0):
        sq = math.sqrt(D)
        solutions = (-b + sq) / (2*a), (-b - sq) / (2*a)
    return solutions

def initialization():
    coef = []
    arg = sys.argv
    if (len(arg) > 1):
        coef = arg[1:4]
    else:
        for i in ["A", "B", "C"]:
            while (True):
                try:
                    coef.append(float(input(f"Введите коэффициент {i}: ")))
                    break
                except ValueError: print("Неверный коэффициент. Попробуйте снова.\n")
            except: return -1
    return coef

def main(*args, **kwargs):

    coef = initialization()
    quadratic_solutions = SolveQuadratic(coef)
    solutions = []
    for i in quadratic_solutions:
        solutions.extend(SolveQuadratic([1, 0, -i]))
    print(*solutions)
    return 0
```

```
if (__name__ == "__main__"): main()
```

## Дополнительное задание 1

### class.py

```
import math
from typing import List
import sys
```

```
class Quadratic:
```

```
    def __init__(self, *args, **kwargs) -> List[float]:
```

```
        self.coef = []
```

```
        if (len(args) == 4):
```

```
            self.coef = args[1:4]
```

```
        else:
```

```
            for i in ["A", "B", "C"]:
```

```
                while (True):
```

```
                    try:
```

```
                        self.coef.append(float(input(f"Введите коэффициент {i}: ")))
```

```
                        break
```

```
                    except ValueError: print("Неверный коэффициент. Попробуйте снова.\n")
```

```
    def get(self):
```

```
        return self.coef[0], self.coef[1], self.coef[2]
```

```
    def solve(self) -> List:
```

```
        a, b, c = map(float, self.get())
```

```
        D = math.pow(b, 2) - 4 * a * c
```

```
        if (D < 0):
```

```
            solutions = ()
```

```
        if (D == 0):
```

```
            solutions = -b / (2*a)
```

```
        if (D > 0):
```

```
            sq = math.sqrt(D)
```

```
            solutions = (-b + sq) / (2*a), (-b - sq) / (2*a)
```

```
        return solutions
```

```
def main(*args, **kwargs):
```

```
    initial = Quadratic(sys.argv).solve()
```

```
    solutions = []
```

```
    for i in initial:
```

```
        solutions.extend(Quadratic("", 1, 0, -i).solve())
```

```
    print(*solutions)
```

```
if (__name__ == "__main__"): main()
```

```
[user@nobara-pc] - [~/Documents/py_sem3/lab_python_intro] - [Вт дек 17, 16:13]
• [[$] <git:(main*)> /usr/bin/python /home/user/Documents/py_sem3/lab_python_intro/class.py
Введите коэффициент A: 1
Введите коэффициент B: -5
Введите коэффициент C: 4
2.0 -2.0 1.0 -1.0

[user@nobara-pc] - [~/Documents/py_sem3/lab_python_intro] - [Вт дек 17, 16:13]
• [[$] <git:(main*)> /usr/bin/python /home/user/Documents/py_sem3/lab_python_intro/def.py
Введите коэффициент A: 1
Введите коэффициент B: -5
Введите коэффициент C: 6
1.7320508075688772 -1.7320508075688772 1.4142135623730951 -1.4142135623730951

[user@nobara-pc] - [~/Documents/py_sem3/lab_python_intro] - [Вт дек 17, 16:20]
• [[$] <git:(main*)> /usr/bin/python /home/user/Documents/py_sem3/lab_python_intro/class.py
Введите коэффициент A: 1
Введите коэффициент B: DFFF
Неверный коэффициент. Попробуйте снова.

Введите коэффициент B: -5
Введите коэффициент C: 6
1.7320508075688772 -1.7320508075688772 1.4142135623730951 -1.4142135623730951

[user@nobara-pc] - [~/Documents/py_sem3/lab_python_intro] - [Вт дек 17, 16:20]
• [[$] <git:(main*)> /usr/bin/python /home/user/Documents/py_sem3/lab_python_intro/def.py
Введите коэффициент A: aaa
Неверный коэффициент. Попробуйте снова.

Введите коэффициент A: 1
Введите коэффициент B: -5
Введите коэффициент C: 4
2.0 -2.0 1.0 -1.0
```