Московский государственный технический университет

им. Н.Э. Баумана

(МГТУ им. Н.Э. Баумана)

Кафедра «Системы обработки информации и управления» (ИУ5)

Лабораторная работа № 1

По дисциплине: «Парадигмы и конструкции языков программирования»

«ПРОГРАММА ДЛЯ РЕШЕНИЯ БИКВАДРАТНОГО УРАВНЕНИЯ»

Выполнила: Вешторт Е. С.,

студентка группы ИУ5-32Б

Проверил: Нардид А. Н.

г. Москва 2023 г.

Задание: написать программу для решения биквадратного уравнения.

Python (Продедурная парадигма):

```
import sys
import math

def get_coef(index, prompt):
     try:
         coef_str = sys.argv[index]
     except:
         print(prompt)
         coef_str = input()
     while True:
         try:
             coef = float(coef_str)
             break
         except:
             print('Ошибка: введено не число. Повторите ввод коэффициента.')
             coef_str = input()

     return coef


def get_roots(a, b, c):
    pre_result = []
    result = []
    D = b*b - 4*a*c
    if a == 0.0:
        pre_result.append(-c/b)
    elif D == 0.0:
        pre_root = -b / (2.0*a)
        pre_result.append(pre_root)
    elif D > 0.0:
        sqD = math.sqrt(D)
        pre_root1 = (-b + sqD) / (2.0*a)
        pre_root2 = (-b - sqD) / (2.0*a)
        pre_result.append(pre_root1)
        pre_result.append(pre_root2)
    for pre_root in pre_result:
        if pre_root > 0:
            result.append(math.sqrt(pre_root))
            result.append(-math.sqrt(pre_root))
        elif pre_root == 0:
            result.append(pre_root)
    return result


def main():

    a = get_coef(1, 'Введите коэффициент A:')
    b = get_coef(2, 'Введите коэффициент B:')
    c = get_coef(3, 'Введите коэффициент C:')

    roots = get_roots(a,b,c)

    len_roots = len(roots)
    if len_roots == 0:
        print('Нет корней')
    elif len_roots == 1:
        print('Один корень: {}'.format(roots[0]))
    elif len_roots == 2:
        print('Два корня: {} и {}'.format(roots[0], roots[1]))
    elif len_roots == 3:
```

```
            print('Три корня: {}, {} и {}'.format(roots[0], roots[1], roots[2]))
        elif len_roots == 4:
            print('Четыре корня: {}, {}, {}, и {}'.format(roots[0], roots[1], roots[2],
    roots[3]))


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

```
Введите коэффициент A:
1
Введите коэффициент B:
-4
Введите коэффициент C:
1
Четыре корня: 1.9318516525781366, -1.9318516525781366, 0.5176380902050416, и -0.5176380902050416
```

Python (Объектно-ориентированная парадигма):

```
import math
import sys

class BiSquare:
    def __init__(self):
        self.coef_A = 0.0
        self.coef_B = 0.0
        self.coef_C = 0.0
        self.num_roots = 0
        self.roots_list = []

    def get_coef(self, index, prompt):
        try:
            coef_str = sys.argv[index]
        except:
            print(prompt)
            coef_str = input()
        while True:
            try:
                coef = float(coef_str)
                break
            except:
                print('Ошибка: введено не число. Повторите ввод коэффициента.')
                coef_str = input()

        return coef

    def get_coefs(self):
        self.coef_A = self.get_coef(1, 'Введите коэффициент A:')
        self.coef_B = self.get_coef(2, 'Введите коэффициент B:')
        self.coef_C = self.get_coef(3, 'Введите коэффициент C:')

    def calculate_roots(self):
        pre_roots = []
        a = self.coef_A
        b = self.coef_B
        c = self.coef_C
        D = b*b - 4*a*c
        if a == 0.0:
            pre_roots.append(-c/b)
        elif D == 0.0:
            pre_roots.append(-b / (2.0 * a))
        elif D > 0.0:
            sqD = math.sqrt(D)
            pre_roots.append((-b + sqD) / (2.0*a))
            pre_roots.append((-b - sqD) / (2.0*a))
```

```
            for pre_root in pre_roots:
                if pre_root == 0:
                    self.num_roots += 1
                    self.roots_list.append(pre_root)
                elif pre_root > 0:
                    self.num_roots += 2
                    self.roots_list.append(math.sqrt(pre_root))
                    self.roots_list.append(-math.sqrt(pre_root))

        def print_roots(self):
            if self.num_roots != len(self.roots_list):
                print(('Ошибка. Уравнение содержит {} действительных корней, ' +\
                'но было вычислено {} корней.').format(self.num_roots,
len(self.roots_list)))
            else:
                if self.num_roots == 0:
                    print('Нет корней')
                elif self.num_roots == 1:
                    print('Один корень: {}'.format(self.roots_list[0]))
                elif self.num_roots == 2:
                    print('Два корня: {} и {}'.format(self.roots_list[0],
self.roots_list[1]))
                elif self.num_roots == 3:
                    print('Три корня: {}, {} и {}'.format(roots[0], roots[1], roots[2]))
                elif self.num_roots == 4:
                    print('Четыре корня: {}, {}, {}, и {}'.format(roots[0], roots[1],
roots[2], roots[3]))

        def main():
            r = BiSquare()
            r.get_coefs()
            r.calculate_roots()
            r.print_roots()

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

```
Введите коэффициент A:
1
Введите коэффициент B:
-5
Введите коэффициент C:
2
Четыре корня: 2.135779205069857, -2.135779205069857, 0.6621534468619564, и -0.6621534468619564
```

F#:

```
        open System

        type BiSquareRootResult =
            | NoRoots
            | OneRoot of double
            | TwoRoots of double * double
            | ThreeRoots of double * double * double
            | FourRoots of double * double * double * double

        let CalculateRoots(a: double, b: double, c: double): BiSquareRootResult =
            let D = b * b - 4.0 * a * c

            if a = 0 then
                let pre_rt = -c / b
                match pre_rt with
                |r when r < 0 -> NoRoots
                |r when r = 0 -> OneRoot 0.0
```

```
            |r -> TwoRoots (Math.Sqrt(r), -Math.Sqrt(r))
        else if D < 0.0 then
            NoRoots
        else if D = 0.0 then
            let pre_rt = -b / (2.0 * a)
            match pre_rt with
            |r when r < 0 -> NoRoots
            |r when r = 0 -> OneRoot 0.0
            |r -> TwoRoots (Math.Sqrt(r), -Math.Sqrt(r))
        else
            let sqrtD = Math.Sqrt(D)
            let pre_rt1 = (-b + sqrtD) / (2.0 * a)
            let pre_rt2 = (-b - sqrtD) / (2.0 * a)

            match pre_rt1, pre_rt2 with
            |r1, r2 when r1 = 0.0 && r2 < 0.0 -> OneRoot 0.0
            |r1, r2 when r1 < 0.0 && r2 = 0.0 -> OneRoot 0.0
            |r1, r2 when r1 < 0.0 && r2 < 0.0 -> NoRoots
            |r1, r2 when r1 < 0.0 && r2 > 0.0 -> TwoRoots (Math.Sqrt(r2), -
    Math.Sqrt(r2))
            |r1, r2 when r2 < 0.0 && r1 > 0.0 -> TwoRoots (Math.Sqrt(r1), -
    Math.Sqrt(r1))
            |r1, r2 when r1 = 0.0 && r2 > 0.0 -> ThreeRoots (0.0, Math.Sqrt(r2), -
    Math.Sqrt(r2))
            |r1, r2 when r2 = 0.0 && r1 > 0.0 -> ThreeRoots (0.0, Math.Sqrt(r1), -
    Math.Sqrt(r1))
            |r1, r2 -> FourRoots (Math.Sqrt(r1), -Math.Sqrt(r1), Math.Sqrt(r2), -
    Math.Sqrt(r2))

    let PrintRoots(a: double, b: double, c: double): unit =
        printf "Коэффициенты: a=%f, b=%f, c=%f. " a b c
        match CalculateRoots(a, b, c) with
        | NoRoots -> printfn "Нет корней"
        | OneRoot(rt) -> printfn "Один корень: %f" rt
        | TwoRoots(rt1, rt2) -> printfn "Два корня: %f и %f" rt1 rt2
        | ThreeRoots(rt1, rt2, rt3) -> printfn "Три корня: %f, %f и %f" rt1 rt2 rt3
        | FourRoots(rt1, rt2, rt3, rt4) -> printfn "Четыре корня: %f, %f, %f и %f" rt1
    rt2 rt3 rt4

    let rec readFloat() =
        printfn "Введите коэффициент:"
        match System.Double.TryParse(System.Console.ReadLine()) with
        | false, _ ->
            printf "Ошибка: введено не число. Повторите ввод коэффициента"
            readFloat()
        | true, x -> x

    [<EntryPoint>]
    let main argv =
        let a = readFloat()
        let b = readFloat()
        let c = readFloat()

        PrintRoots(a, b, c)

        Console.ReadLine() |> ignore
        0
```

```
Введите коэффициент:
1
Введите коэффициент:
-5
Введите коэффициент:
2
Коэффициенты: a=1.000000, b=-5.000000, c=2.000000. Четыре корня: 2.135779, -2.135779, 0.662153 и -0.662153
```