

**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:
студент группы ИУ5-33Б
Буров Р. Ю.

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю. Е.

Москва, 2025 г.

OOP.py

```
import sys
import math

class Equation:
    def __init__(self, a=1.0, b=1.0, c=1.0):
        self.a = a
        self.b = b
        self.c = c
        self.roots = []

    def coef_info(self):
        print(self.a, self.b, self.c)

    def roots_info(self):
        len_roots = len(self.roots)
        if len_roots == 0:
            print("Нет корней")
        elif len_roots == 1:
            print("Один корень: " + str(self.roots[0]))
        elif len_roots == 2:
            print("Два корня: " + str(self.roots[0]) + " и " + str(self.roots[1]))
        elif len_roots == 3:
            print("Три корня: " + str(self.roots[0]) + ", " + str(self.roots[1])
                  + ", " + str(self.roots[2]))
        elif len_roots == 4:
            print("Четыре корня: " + str(self.roots[0]) + ", " + str(self.roots[1]) + ", "
                  + str(self.roots[2]) + ", " + str(self.roots[3]))

    def get_coef(self):
        if len(sys.argv) == 4:
```

```
try:
    self.a = float(sys.argv[1])
    self.b = float(sys.argv[2])
    self.c = float(sys.argv[3])
except ValueError:
    print("Ошибка ввода командной строки")
    sys.exit(0)

elif len(sys.argv) == 1:
    while True:
        try:
            print("Укажите коэффициенты уравнения")
            self.a = float(input("Коэффициент A: "))
            self.b = float(input("Коэффициент B: "))
            self.c = float(input("Коэффициент C: "))
            break
        except ValueError:
            print("Ошибка ввода. Попробуйте еще раз.")

elif (len(sys.argv) != 1 and len(sys.argv) != 4):
    print("Ошибка ввода командной строки")
    sys.exit(0)

def get_roots(self):
    D = self.b**2 - 4*self.a*self.c
    if (D == 0):
        t = -self.b / (2.0*self.a)
        if (t > 0):
            root1 = math.sqrt(t)
            root2 = -math.sqrt(t)
            self.roots.append(round(root1, 2))
```

```
    self.roots.append(round(root2, 2))

    elif(t == 0):
        self.roots.append(0)

    elif (D > 0):
        sqD = math.sqrt(D)
        t1 = (-self.b + sqD) / (2.0*self.a)
        if (t1 > 0):
            root1 = math.sqrt(t1)
            root2 = -math.sqrt(t1)
            self.roots.append(round(root1, 2))
            self.roots.append(round(root2, 2))

        elif(t1 == 0):
            self.roots.append(0)
            t2 = (-self.b - sqD) / (2.0*self.a)
            if (t2 > 0):
                root3 = math.sqrt(t2)
                root4 = -math.sqrt(t2)
                self.roots.append(round(root3, 2))
                self.roots.append(round(root4, 2))

            elif(t2 == 0):
                self.roots.append(0)

def main():
    eq = Equation()

    eq.get_coef()
    eq.coef_info()
    eq.get_roots()
    eq.roots_info()

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

PP.py

```
import math

import sys

def get_coef():
    list = []
    if (len(sys.argv) == 1):
        print ("Ввод коэффициентов биквадратного уравнения")
    for i in range(3):
        while True:
            try:
                coef = float(input("Введите " + str(i + 1) + "й коэффициент: "))
                break
            except ValueError:
                print("Ошибка ввода. Попробуйте снова.")
        list.append(coef)

    elif (len(sys.argv) == 4):
        for i in range(3):
            try:
                coef = float(sys.argv[i + 1])
            except ValueError:
                print("Ошибка ввода командной строки")
                sys.exit(0)
            list.append(coef)

    elif (len(sys.argv) != 1 and len(sys.argv) != 4):
        print("Ошибка ввода командной строки")
        sys.exit(0)

    return list

def get_roots(list):
```

```
a = list[0]
b = list[1]
c = list[2]
result = []
D = b**2 - 4*a*c
if (D == 0):
    t = -b / (2.0*a)
    if (t > 0):
        root1 = math.sqrt(t)
        root2 = -math.sqrt(t)
        result.append(round(root1, 2))
        result.append(round(root2, 2))
    elif(t == 0):
        result.append(0)
    elif (D > 0):
        sqD = math.sqrt(D)
        t1 = (-b + sqD) / (2.0*a)
        if (t1 > 0):
            root1 = math.sqrt(t1)
            root2 = -math.sqrt(t1)
            result.append(round(root1, 2))
            result.append(round(root2, 2))
        elif(t1 == 0):
            result.append(0)
        t2 = (-b - sqD) / (2.0*a)
        if (t2 > 0):
            root3 = math.sqrt(t2)
            root4 = -math.sqrt(t2)
            result.append(round(root3, 2))
            result.append(round(root4, 2))
        elif(t2 == 0):
```

```
result.append(0)

return result

def print_roots(roots):

len_roots = len(roots)

if len_roots == 0:
    print("Нет корней")

elif len_roots == 1:
    print("Один корень: " + str(roots[0]))

elif len_roots == 2:
    print("Два корня: " + str(roots[0]) + " и " + str(roots[1]))

elif len_roots == 3:
    print("Три корня: " + str(roots[0]) + ", " + str(roots[1]) + ", " + str(roots[2]))

elif len_roots == 4:
    print("Четыре корня: " + str(roots[0]) + ", " + str(roots[1]) + ", " + str(roots[2]) + ", " + str(roots[3]))


def main():

list_coef = get_coef()

roots = get_roots(list_coef)

print_roots(roots)

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

PP.go

```
package main

import (
    "fmt"
    "math"
    "os"
    "strconv"
)

func Round(number float64) float64{
    rounded := math.Round(number * 100) / 100
    return rounded
}

func get_coef() []float64 {
    var slice []float64
    switch len(os.Args) {
    case 1:
        fmt.Println("Ввод коэффициентов уравнения")
        i := 0
        for i < 3 {
            fmt.Print("Введите ", i+1, "й коэффициент: ")
            var str string
            fmt.Scan(&str)
            coef, err := strconv.ParseFloat(str, 64)
            if err != nil {
                fmt.Println("Ошибка ввода. Попробуйте снова.")
            } else {
                slice = append(slice, coef)
                i++
            }
        }
    }
}
```

```
        }

    }

case 4:

for i := range 3 {

    coef, err := strconv.ParseFloat(os.Args[i+1], 64)

    if err == nil {

        slice = append(slice, coef)

        i++

    } else {

        fmt.Println("Ошибка ввода командной строки")

        os.Exit(0)

    }

}

default:

fmt.Println("Ошибка ввода командной строки")

os.Exit(0)

}

return slice

}

func get_roots(slice []float64) []float64 {

    a := slice[0]

    b := slice[1]

    c := slice[2]

    var result []float64

    D := math.Pow(b, 2) - 4*a*c

    if D == 0 {

        t := -b / (2.0*a)

        if t > 0 {

            root1 := Round(math.Sqrt(t))

            root2 := -Round(math.Sqrt(t))

        }

    }

}
```

```

        result = append(result, root1)
        result = append(result, root2)

    } else if t == 0 {
        result = append(result, 0)
    }

} else if D > 0 {

    sqD := math.Sqrt(D)
    t1 := (-b + sqD) / (2.0*a)
    if t1 > 0 {
        root1 := Round(math.Sqrt(t1))
        root2 := -Round(math.Sqrt(t1))
        result = append(result, root1)
        result = append(result, root2)
    } else if(t1 == 0) {
        result = append(result, 0)
    }

    t2 := (-b - sqD) / (2.0*a)
    if t2 > 0 {
        root3 := Round(math.Sqrt(t2))
        root4 := -Round(math.Sqrt(t2))
        result = append(result, root3)
        result = append(result, root4)
    } else if t2 == 0 {
        result = append(result, 0)
    }

}

return result
}

```

```
func print_roots(slice []float64) {  
    len_roots := len(slice)  
    switch len_roots {  
        case 0:  
            fmt.Println("Нет корней")  
        case 1:  
            fmt.Println("Один корень:", slice[0])  
        case 2:  
            fmt.Println("Два корня:", slice[0], "и", slice[1])  
        case 3:  
            fmt.Println("Три корня:", slice[0], ", ", slice[1], ", ", slice[2])  
        case 4:  
            fmt.Println("Четыре корня:", slice[0], ", ", slice[1], ", ", slice[2], ", ", slice[3])  
    }  
}  
  
func main() {  
    coef_list := get_coef()  
    roots := get_roots(coef_list)  
    print_roots(roots)  
}
```

Скриншоты работы приложения

- `artorias@LAPTOP-2ESPT302:~/IU5/PCPL/Labs/lab-1$ go run PP.go`
Ввод коэффициентов уравнения
Введите 1й коэффициент: 2
Введите 2й коэффициент: 4
Введите 3й коэффициент: -1
Два корня: 0.47 и -0.47
- `artorias@LAPTOP-2ESPT302:~/IU5/PCPL/Labs/lab-1$ python3 OOP.py 4 2 2`
4.0 2.0 2.0
Нет корней
- `artorias@LAPTOP-2ESPT302:~/IU5/PCPL/Labs/lab-1$ python3 PP.py 4 2 2`
Нет корней
- `artorias@LAPTOP-2ESPT302:~/IU5/PCPL/Labs/lab-1$ python3 PP.py s 45`
Ошибка ввода командной строки
- `artorias@LAPTOP-2ESPT302:~/IU5/PCPL/Labs/lab-1$ python3 PP.py`
Ввод коэффициентов биквадратного уравнения
Введите 1й коэффициент: 5
Введите 2й коэффициент: _
Ошибка ввода. Попробуйте снова.
Введите 2й коэффициент: 5
Введите 3й коэффициент: 5
Нет корней
- `artorias@LAPTOP-2ESPT302:~/IU5/PCPL/Labs/lab-1$ python3 PP.py`
Ввод коэффициентов биквадратного уравнения
Введите 1й коэффициент: 10 12 3
Ошибка ввода. Попробуйте снова.
Введите 1й коэффициент: 10
Введите 2й коэффициент: 12
Введите 3й коэффициент: 2
Нет корней