**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет ИУ

Кафедра ИУ5

Курс «Основы информатики»

Отчет по лабораторной работе №1

“Основные конструкции языка Python”

Разработана на языке Golang

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы 36б:  Азиев А. А. |  | преподаватель каф. \_\_\_\_ |
|  |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2024г.

**Задача:**

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.

**Код программы:**

package main

import (

    "fmt"

    "math"

    "os"

    "strconv"

)

func getCoefficient(name string) float64 {

    for {

        fmt.Printf("Введите коэффициент %s: ", name)

        var input string

        fmt.Scanln(&input)

        value, err := strconv.ParseFloat(input, 64)

        if err == nil {

            return value

        }

        fmt.Println("Некорректное значение, попробуйте еще раз.")

    }

}

func solveBiquadratic(a, b, c float64) ([]float64, error) {

    if a == 0 {

        return nil, fmt.Errorf("коэффициент A не может быть равен 0")

    }

    d := b\*b - 4\*a\*c

    if d < 0 {

        return nil, nil

    }

    var roots []float64

    root1 := (-b + math.Sqrt(d)) / (2 \* a)

    root2 := (-b - math.Sqrt(d)) / (2 \* a)

    if root1 >= 0 {

        roots = append(roots, math.Sqrt(root1), -math.Sqrt(root1))

    }

    if root2 >= 0 && root2 != root1 {

        roots = append(roots, math.Sqrt(root2), -math.Sqrt(root2))

    }

    return roots, nil

}

func main() {

    var a, b, c float64

    var err error

    if len(os.Args) == 4 {

        a, err = strconv.ParseFloat(os.Args[1], 64)

        if err != nil {

            a = getCoefficient("A")

        }

        b, err = strconv.ParseFloat(os.Args[2], 64)

        if err != nil {

            b = getCoefficient("B")

        }

        c, err = strconv.ParseFloat(os.Args[3], 64)

        if err != nil {

            c = getCoefficient("C")

        }

    } else {

        a = getCoefficient("A")

        b = getCoefficient("B")

        c = getCoefficient("C")

    }

    roots, solveErr := solveBiquadratic(a, b, c)

    if solveErr != nil {

        fmt.Println("Ошибка:", solveErr)

        return

    }

    if roots == nil {

        fmt.Println("Действительных корней нет.")

    } else {

        fmt.Println("Действительные корни уравнения:")

        for \_, root := range roots {

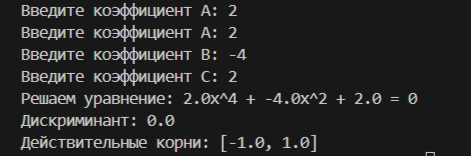
            fmt.Printf("%.4f\n", root)

        }

    }

}

**Результат выполнения:**

****