**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 Go»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-31Б |  | преподаватель каф. ИУ5 |
| Князев А.М. |  | Нардид А.Н. |
|  |  |  |

Москва, 2024 г.

**Описание задания**

1. Организовать ввод коэффициентов a, b и c.
2. Проверить, является ли уравнение квадратным (если a равно нулю, выводится сообщение об ошибке).
3. Рассчитать дискриминант.
4. Реализовать ветвление для обработки трёх случаев:

* Если дискриминант больше нуля — найти два действительных корня и вывести их.
* Если дискриминант равен нулю — найти один действительный корень и вывести его.
* Если дискриминант меньше нуля — предложить пользователю рассчитать комплексные корни.

1. Если пользователь соглашается на вычисление комплексных корней:

* Найти и вывести комплексные корни в виде пар с реальной и мнимой частью.

1. Обеспечить корректную обработку ввода и взаимодействие программы с пользователем

**Текст программы**

Файл *lab1.go*

package main

import (

    "fmt"

    "math"

)

func main() {

    var a, b, c float64

    fmt.Println("Введите коэффициенты квадратного уравнения ax^2 + bx + c = 0:")

    fmt.Print("a = ")

    fmt.Scanln(&a)

    fmt.Print("b = ")

    fmt.Scanln(&b)

    fmt.Print("c = ")

    fmt.Scanln(&c)

    if a == 0 {

        fmt.Println("Уравнение не является квадратным")

        return

    }

    d := b\*b - 4\*a\*c

    if d > 0 {

        x1 := (-b + math.Sqrt(d)) / (2 \* a)

        x2 := (-b - math.Sqrt(d)) / (2 \* a)

        fmt.Printf("Корни уравнения: x1 = %.2f, x2 = %.2f\n", x1, x2)

    } else if d == 0 {

        x := -b / (2 \* a)

        fmt.Printf("Корни уравнения: x = %.2f\n", x)

    } else {

        fmt.Println("Уравнение не имеет действительных корней, хотите посчитать комплексные корни?(y/n)")

        var e byte

        fmt.Scanf("%c", &e)

        if e == 'n' {

            return

        } else {

            realPart := -b / (2 \* a)

            imagPart := math.Sqrt(math.Abs(d)) / (2 \* a)

            x1 := complex(realPart, imagPart)

            x2 := complex(realPart, -imagPart)

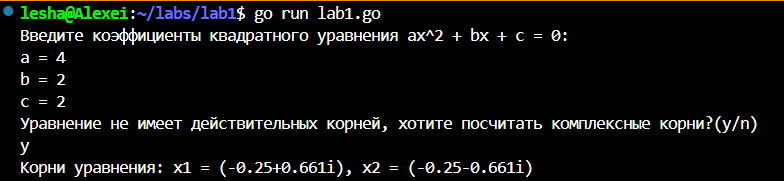
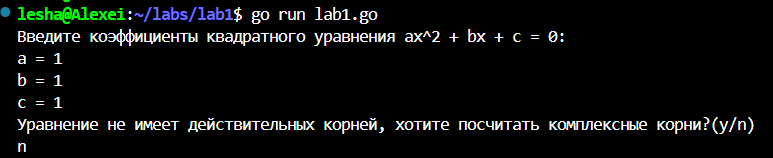
            fmt.Printf("Корни уравнения: x1 = %.3v, x2 = %.3v\n", x1, x2)

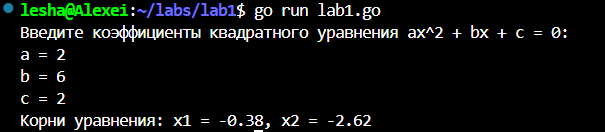
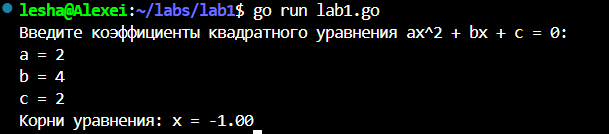
        }

    }

}

**Экранные формы с примерами выполнения программы**

****

****