**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 № 5

«Тесты на языке Go»

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

Москва, 2024 г.

**Описание задания**

**Описание задания:**

Протестировать функцию для вычисления корней квадратного уравнения вида

на языке Go. Основное внимание уделить обработке различных случаев на основе значения дискриминанта.

**Функциональные требования:**

1. **Разработка функции**:
   * Функция CalculateRoots принимает коэффициенты a, b, c и возвращает корни уравнения.
   * Обработка трёх возможных случаев:
     + D > 0: два действительных корня.
     + D = 0: один действительный корень.
     + D < 0: нет действительных корней, возвращается ошибка.
   * Использовать формулу дискриминанта: и формулы корней: .
2. **Разработка тестов**:
   * Использовать пакет testing для автоматического тестирования функции.
   * Реализовать таблицу тестов с различными входными значениями:
     + Два действительных корня (D > 0).
     + Один действительный корень (D = 0).
     + Отсутствие действительных корней (D < 0).
   * Проверить корректность возвращаемых значений с использованием допусков при сравнении чисел с плавающей точкой.
3. **Структура программы**:
   * Основная функция тестирования должна организовывать тесты с использованием t.Run() для каждого сценария.
   * Обеспечить понятный вывод ошибок при несоответствии ожидаемых и полученных значений.

**Дополнительные требования**:

* Обеспечить читаемость и структурированность кода.
* Реализовать обработку ошибок корректно.
* Использовать пакет math для вычисления квадратного корня и сравнения чисел с плавающей точкой.
* Намеренно создадим провальный тест (последний)

**Текст программы**

Файл *main\_table\_test.go*

package main

import (

    "math"

    "testing"

)

func TestCalculateRootsTable(t \*testing.T) {

    tests := []struct {

        name           string

        a, b, c        float64

        wantX1, wantX2 float64

        expectError    bool

    }{

        {"TwoRealRoots", 1, -3, 2, 2, 1, false},

        {"OneRealRoot", 1, 2, 1, -1, -1, false},

        {"NoRealRoots", 1, 0, 1, 0, 0, true},

    }

    for \_, tt := range tests {

        t.Run(tt.name, func(t \*testing.T) {

            x1, x2, err := CalculateRoots(tt.a, tt.b, tt.c)

            if (err != nil) != tt.expectError {

                t.Errorf("expected error %v, got %v", tt.expectError, err)

                return

            }

            if !tt.expectError {

                if math.Abs(x1-tt.wantX1) > 0.001 || math.Abs(x2-tt.wantX2) > 0.001 {

                    t.Errorf("CalculateRoots(%v, %v, %v) = (%v, %v), want (%v, %v)", tt.a, tt.b, tt.c, x1, x2, tt.wantX1, tt.wantX2)

                }

            }

        })

    }

}

Файл *main\_unit\_test.go*

package main

import (

    "math"

    "testing"

)

func TestCalculateRoots(t \*testing.T) {

    tests := []struct {

        a, b, c        float64

        wantX1, wantX2 float64

        expectError    bool

    }{

        {1, -3, 2, 2, 1, false},  // Два действительных корня

        {1, 2, 1, -1, -1, false}, // Один действительный корень (D = 0)

        {1, 0, 1, 0, 0, true},    // Нет действительных корней (D < 0)

        {1, 2, 3, 0, 0, false},   // Провленный тест

    }

    for \_, tt := range tests {

        x1, x2, err := CalculateRoots(tt.a, tt.b, tt.c)

        if (err != nil) != tt.expectError {

            t.Errorf("CalculateRoots(%v, %v, %v) ожидал ошибку %v, но получил %v", tt.a, tt.b, tt.c, tt.expectError, err != nil)

            continue

        }

        if !tt.expectError {

            if math.Abs(x1-tt.wantX1) > 0.001 || math.Abs(x2-tt.wantX2) > 0.001 {

                t.Errorf("CalculateRoots(%v, %v, %v) = (%v, %v), ожидалось (%v, %v)", tt.a, tt.b, tt.c, x1, x2, tt.wantX1, tt.wantX2)

            }

        }

    }

}

Файл *main.go*

package main

import (

    "fmt"

    "math"

)

func CalculateRoots(a, b, c float64) (float64, float64, error) {

    D := b\*b - 4\*a\*c

    if D < 0 {

        return 0, 0, fmt.Errorf("уравнение не имеет действительных корней")

    }

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

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

    return x1, x2, nil

}

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)

    x1, x2, err := CalculateRoots(a, b, c)

    if err != nil {

        fmt.Println(err)

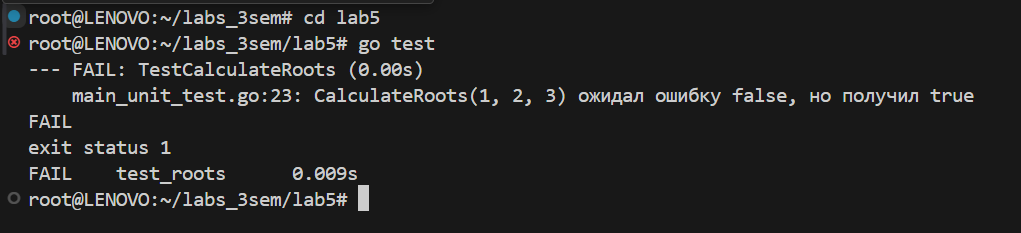
    } else {

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

    }

}

**Экранные формы с примерами выполнения программы**

****