**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 № 6

«Разработка телеграмм бота на Go»

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

Москва, 2024 г.

**Описание задания**

1. Написать программу на языке Go, которая реализует Telegram-бота с использованием библиотеки *github.com/go-telegram-bot-api/telegram-bot-api/v5*. Бот должен предоставлять пользователю выбор из двух функций:

* Найти корни квадратного уравнения: Пользователь вводит коэффициенты a, b и c квадратного уравнения, бот вычисляет и возвращает корни (действительные или комплексные).
* Найти площадь прямоугольника: Пользователь вводит длины сторон прямоугольника, бот вычисляет и возвращает площадь.

1. Бот должен корректно обрабатывать некорректный ввод пользователя и выводить соответствующие сообщения об ошибках.

**Текст программы**

Файл *main.go*

package main

import (

 "fmt"

 "github.com/go-telegram-bot-api/telegram-bot-api/v5"

 "log"

 "math"

 "strconv"

 "strings"

)

func main() {

 bot, err := tgbotapi.NewBotAPI("7557980296:AAHWaavcV85arPbn-erWPAuEy176wm7S4Gg")

 if err != nil {

  log.Panic(err)

 }

 bot.Debug = true

 log.Printf("Authorized on account %s", bot.Self.UserName)

 u := tgbotapi.NewUpdate(0)

 u.Timeout = 60

 updates := bot.GetUpdatesChan(u)

 rootButton := tgbotapi.NewKeyboardButton("Найти корни квадратного уравнения")

 areaButton := tgbotapi.NewKeyboardButton("Найти площадь прямоугольника")

 keyboard := tgbotapi.NewReplyKeyboard(

  tgbotapi.NewKeyboardButtonRow(rootButton),

  tgbotapi.NewKeyboardButtonRow(areaButton),

 )

 for update := range updates {

  if update.Message == nil {

   continue

  }

  chatID := update.Message.Chat.ID

  msg := tgbotapi.NewMessage(chatID, "")

  switch update.Message.Text {

  case "/start":

   msg.Text = "Выберите функцию:"

   msg.ReplyMarkup = keyboard

   bot.Send(msg)

  case "Найти корни квадратного уравнения":

   msg.Text = "Введите коэффициенты a, b и c через пробел (например: 1 -3 2):"

   bot.Send(msg)

  case "Найти площадь прямоугольника":

   msg.Text = "Введите длины сторон прямоугольника a и b через пробел (например: 3 4):"

   bot.Send(msg)

  default:

   switch {

   case strings.Contains(update.Message.Text, " "): // Проверяем наличие пробелов, предполагая ввод чисел

    input := strings.Fields(update.Message.Text)

    if len(input) == 3 { // Корни квадратного уравнения

     a, \_ := strconv.ParseFloat(input[0], 64)

     b, \_ := strconv.ParseFloat(input[1], 64)

     c, \_ := strconv.ParseFloat(input[2], 64)

     msg.Text = calculateRoots(a, b, c)

    } else if len(input) == 2 { // Площадь прямоугольника

     a, err1 := strconv.ParseFloat(input[0], 64)

     b, err2 := strconv.ParseFloat(input[1], 64)

     if err1 == nil && err2 == nil {

      msg.Text = calculateRectangleArea(a, b)

     } else {

      msg.Text = "Неверный формат. Пожалуйста, введите два числа через пробел."

     }

    } else {

     msg.Text = "Неверный формат ввода."

    }

   default:

    msg.Text = "Пожалуйста, выберите функцию из предложенных вариантов или введите данные в правильном формате."

   }

   bot.Send(msg)

  }

 }

}

func calculateRoots(a, b, c float64) string {

 d := b\*b - 4\*a\*c

 if a == 0 {

  return "Уравнение не является квадратным"

 }

 if d > 0 {

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

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

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

 } else if d == 0 {

  x := -b / (2 \* a)

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

 } else {

  realPart := -b / (2 \* a)

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

  x1 := complex(realPart, imagPart)

  x2 := complex(realPart, -imagPart)

  return fmt.Sprintf("Уравнение не имеет действительных корней: Корни уравнения: x1 = %.3v, x2 = %.3v\n", x1, x2)

 }

}

func calculateRectangleArea(a, b float64) string {

 if a <= 0 || b <= 0 {

  return "Длины сторон прямоугольника должны быть положительными числами."

 }

 area := a \* b

 return fmt.Sprintf("Площадь прямоугольника: %.2f", area)

}

**Экранные формы с примерами выполнения программы**



