**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 № 2

«Работа с массивами в Go»

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

Москва, 2024 г.

**Описание задания**

1. Ввод массива:

* Пользователь вводит элементы массива через пробел с клавиатуры.
* Программа сохраняет их в массив.

1. Определение максимального и минимального элементов массива:

* Реализовать функции *findMax* и *findMin* для поиска максимального и минимального значений.

1. Добавление элемента:

* Реализовать функцию *appendElement*, которая добавляет новый элемент в конец массива.

1. Удаление элемента:

* Реализовать функцию *removeElement*, которая удаляет элемент по заданному индексу.
* Обработать некорректный ввод индекса (например, отрицательные или слишком большие значения).

1. Меню действий:

* Организовать цикл, в котором пользователь может выполнять следующие действия:
  + Добавить элемент (*add*)
  + Удалить элемент (*remove*)
  + Завершить выполнение программы (*exit*)

1. Вывод результатов:

После каждого действия программа выводит обновлённый массив.

**Текст программы**

Файл *lab2.go*

package main

import (

    "bufio"

    "fmt"

    "os"

    "strconv"

    "strings"

)

func findMax(arr []int) int {

    if len(arr) == 0 {

        return 0

    }

    max := arr[0]

    for \_, v := range arr {

        if v > max {

            max = v

        }

    }

    return max

}

func findMin(arr []int) int {

    if len(arr) == 0 {

        return 0

    }

    min := arr[0]

    for \_, v := range arr {

        if v < min {

            min = v

        }

    }

    return min

}

func appendElement(arr []int, element int) []int {

    return append(arr, element)

}

func removeElement(arr []int, index int) []int {

    return append(arr[:index], arr[index+1:]...)

}

func main() {

    reader := bufio.NewReader(os.Stdin)

    fmt.Print("Введите элементы массива, разделенные пробелами: ")

    input, \_ := reader.ReadString('\n')

    input = strings.TrimSpace(input)

    elements := strings.Split(input, " ")

    arr := make([]int, len(elements))

    for i, element := range elements {

        num, \_ := strconv.Atoi(element)

        arr[i] = num

    }

    fmt.Println("Введенный массив:", arr)

    max := findMax(arr)

    min := findMin(arr)

    fmt.Println("Максимальный элемент:", max)

    fmt.Println("Минимальный элемент:", min)

    for {

        fmt.Print("Выберите действие (add - добавить элемент, remove - удалить элемент, exit - выход): ")

        action, \_ := reader.ReadString('\n')

        action = strings.TrimSpace(action)

        switch action {

        case "add":

            fmt.Print("Введите элемент для добавления: ")

            elementStr, \_ := reader.ReadString('\n')

            elementStr = strings.TrimSpace(elementStr)

            element, \_ := strconv.Atoi(elementStr)

            arr = appendElement(arr, element)

            fmt.Println("Массив после добавления:", arr)

        case "remove":

            fmt.Print("Введите индекс элемента для удаления: ")

            indexStr, \_ := reader.ReadString('\n')

            indexStr = strings.TrimSpace(indexStr)

            index, \_ := strconv.Atoi(indexStr)

            if index < 0 || index >= len(arr) {

                fmt.Println("Неверный индекс ")

                continue

            }

            arr = removeElement(arr, index)

            fmt.Println("Массив после удаления:", arr)

        case "exit":

            fmt.Println("Выход из программы.")

            return

        default:

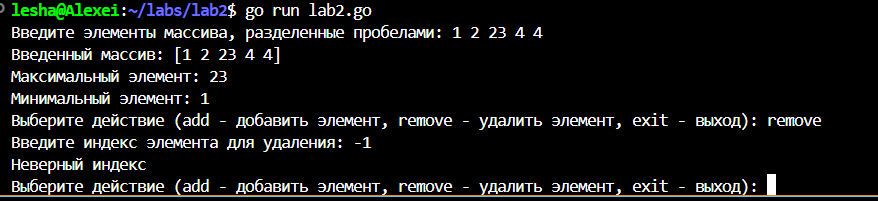
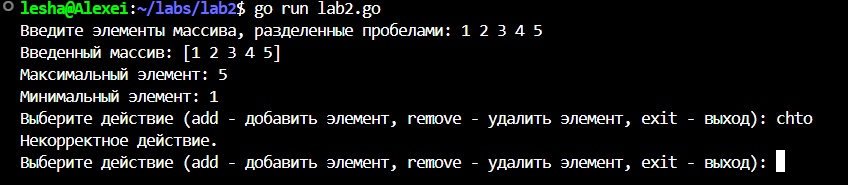
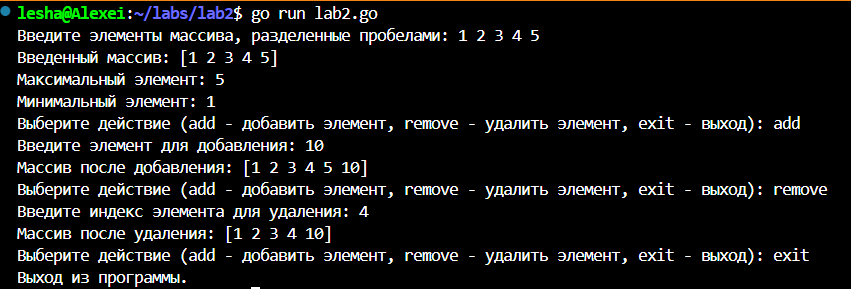
            fmt.Println("Некорректное действие.")

        }

    }

}

**Экранные формы с примерами выполнения программы**

****