Министерство образования и науки РФ

ФГБОУ ВО «Тамбовский государственный технический университет»

Кафедра САПР

Отчёт по лабораторной работе №1

по дисциплине «Основы теории информации и криптографии»

Выполнил: студент группы БВТ211

Глазков С.И.

Проверила: Майстренко Н.В.

# Тамбов 2024

Задание было решено на языке программирования Go по причинам: изучение нового языка и получение нового опыта разработки, также его быстродействие и легкость изучения при имеющихся навыках разработки на других языках программирования.

**Задание**:

Вычислить кол-во информации в тексте используя формулу (1) с помощью программы с GUI. Интерфейс должен содержать выбор языка, по которому будет рассчитываться результат. Доступные языки: русский, английский и произвольный язык.

Формула (1):

**Решение:**

**Листинг программы**

**main.go**

package main

import (

    "fmt"

    "lab1/alphabet"

    "math"

    "strings"

    "fyne.io/fyne/v2"

    "fyne.io/fyne/v2/app"

    "fyne.io/fyne/v2/container"

    "fyne.io/fyne/v2/dialog"

    "fyne.io/fyne/v2/widget"

)

func main() {

    a := app.New()

    w := a.NewWindow("Lab1")

    w.Resize(fyne.NewSize(600, 400))

    languageSelector := widget.NewSelect([]string{alphabet.LangNames[alphabet.Ru], alphabet.LangNames[alphabet.En], alphabet.LangNames[alphabet.CUSTOM]}, func(*value* string) {

        fmt.Println("Выбранный язык", value)

    })

    messageEntry := widget.NewMultiLineEntry()

    customAlphabetEntry := widget.NewMultiLineEntry()

    calculateButton := widget.NewButton("Посчитать", func() {

        selectedLanguage := languageSelector.Selected

        messageText := messageEntry.Text

        customAlphabet := customAlphabetEntry.Text

        var lang alphabet.Langs

        switch selectedLanguage {

        case alphabet.LangNames[alphabet.Ru]:

            lang = alphabet.Ru

        case alphabet.LangNames[alphabet.En]:

            lang = alphabet.En

        case alphabet.LangNames[alphabet.CUSTOM]:

            lang = alphabet.CUSTOM

        }

        messageTextLower := strings.ToLower(messageText)

        if lang == alphabet.CUSTOM {

            alphabetLength, charCount := alphabet.GetAlphabet(lang, messageTextLower, customAlphabet)

            log2 := int(math.Ceil(math.Log2(float64(alphabetLength))))

            information := charCount \* log2

            resultText := fmt.Sprintf("Количество информации: %d бит\nВес символа: %d бит\nКоличество символов: %d", information, log2, charCount)

            dialog.ShowInformation("Результат", resultText, w)

        } else {

            alphabetLength, charCount := alphabet.GetAlphabet(lang, messageTextLower, "")

            log2 := int(math.Ceil(math.Log2(float64(alphabetLength))))

            information := charCount \* log2

            resultText := fmt.Sprintf("Количество информации: %d бит\nВес символа: %d бит\nКоличество символов: %d", information, log2, charCount)

            dialog.ShowInformation("Результат", resultText, w)

        }

    })

    languageSelector.OnChanged = func(*s* string) {}

    context := container.NewVBox(

        languageSelector,

        widget.NewLabel("Введите свое сообщение: "),

        messageEntry,

        calculateButton,

        customAlphabetEntry,

    )

    w.SetContent(context)

    w.ShowAndRun()

}

**alphabet.go**

package alphabet

import "strings"

type Langs int

const (

    Ru Langs = iota

    En

    CUSTOM

)

var LangNames = [...]string{"Русский", "Английский", "Произвольный"}

const (

    RUS\_ALPHABET = "абвгдеёжзийклмнопрстуфхцчшщъыьэюя"

    ENG\_ALPHABET = "abcdefghijklmnopqrstuvwxyz"

    CUSTOM\_ALPHABET = ""

)

func GetAlphabet(*language* Langs, *text* string, *customAlp* string) (int, int) {

    var currentAlphabet string

    switch language {

    case Ru:

        currentAlphabet = RUS\_ALPHABET

    case En:

        currentAlphabet = ENG\_ALPHABET

    case CUSTOM:

        currentAlphabet = customAlp

    }

    charCount := 0

    for \_, char := range text {

        if strings.ContainsRune(currentAlphabet, char) {

            charCount++

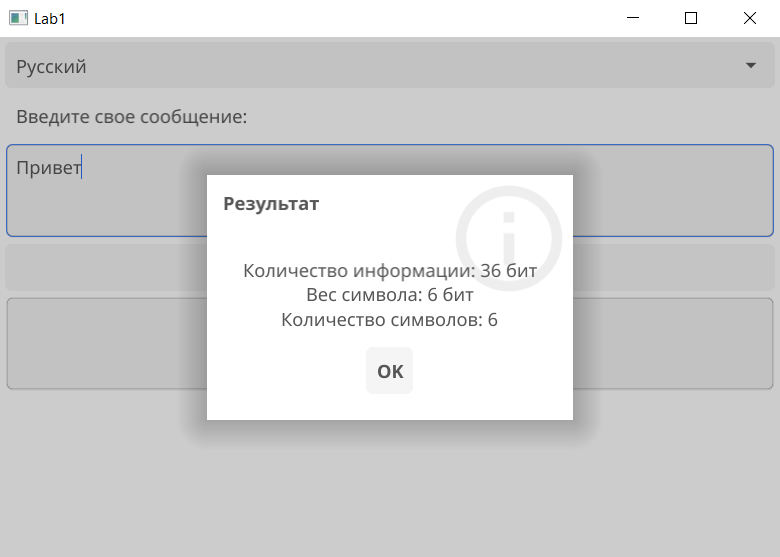
        }

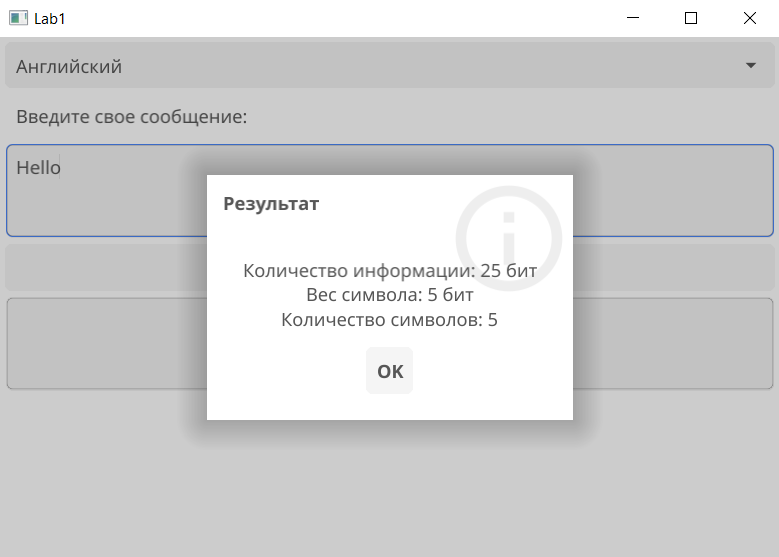
    }

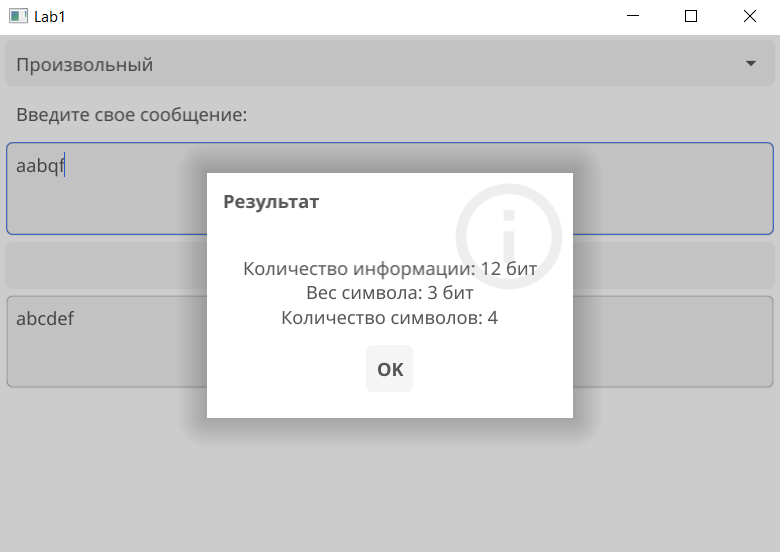
    return len([]rune(currentAlphabet)), charCount

}

**Результаты работы программы**







**Выводы**:

В результате решения поставленной задачи было вычислено количество информации в тексте, а также определен вес каждого алфавита. Задача выполнена с использованием формулы (1). Для взаимодействия с пользователем интерфейс написан с помощью фреймворка Fyne, который поддерживает кроссплатформенность и прост в изучении и взаимодействии для разработчиков, которые были знакомы ранее с другими редакторами GUI.