|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **Компьютерные системы и сети (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.01 Информатика и вычислительная техника**

**Отчет**

**по лабораторной работе № 3**

**Название:** Основы Golang

**Дисциплина:** Языки интернет-программирования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-31БВ |  |  | В.О. Бокова |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | В.Д. Шульман |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2024

**Цель работы** - знакомство с Go, компилируемым многопоточным языком

программирования

Задание:

1. Ознакомьтесь с первыми 2-я разделами курса по Golang.

2. Сделайте форк данного репозитория в GitHub, склонируйте получившуюся

копию локально, создайте от мастера ветку дев и переключитесь на нее

3. Выполните задания. Ссылки на задания содержатся в README-файлах в

директории projects

4. Сделайте отчёт и поместите его в директорию docs

5. Зафиксируйте изменения, сделайте коммит и отправьте полученное

состояние ветки дев в удаленный репозиторий GitHub

6. Через интерфейс GitHub создайте Pull Request dev --> master

Ход работы:

3. Выполнить задания в projects.

A) Первое задание «Triangle»: На вход подаются a и b - катеты прямоугольного

треугольника. Нужно найти длину гипотенузы.

Решение:

package main

import "fmt"

import "math"

func main() {

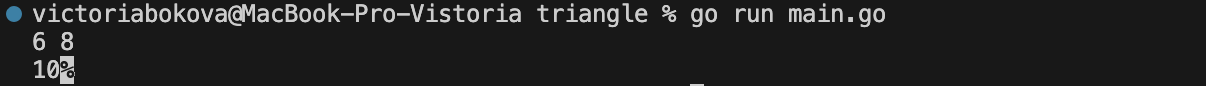
var a, b int

fmt.Scan(&a, &b)

fmt.Print(math.Sqrt(float64(a \* a + b \* b)))

}

**Тестирование:**

****

B) Второе задание «Star»: Дана строка, содержащая только английские буквы

(большие и маленькие). Добавить символ ‘\*’ (звездочка) между буквами (перед

первой буквой и после последней символ ‘\*’ добавлять не нужно).

Решение:

package main

import "fmt"

func main() {

var myString string

fmt.Scan(&myString)

for i, v := range myString {

f i != 0 {

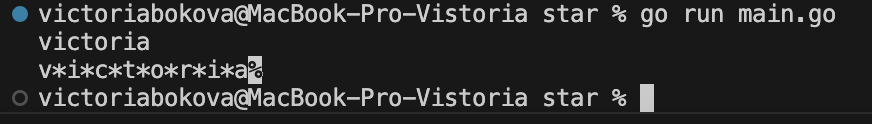
fmt.Print("\*")

}

fmt.Printf("%c", v)

}

**Тестирование:**

****

C) Третье задание «Digits»: Дана строка, содержащая только арабские цифры. Найти и

вывести наибольшую цифру.

Решение:

package main

import "fmt"

func main() {

var myString string

fmt.Scan(&myString)

var mx rune

for \_, v := range myString {

if v > mx {

mx = v

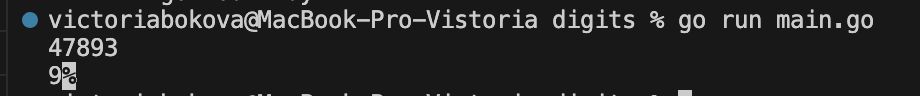
}

}

fmt.Printf("%c", mx)

}

**Тестирование:**



D) Четвертое задание «Atoi»: На вход подается целое число. Необходимо возвести в квадрат каждую цифру числа и вывести получившееся число.

Решение:

package main

import "fmt"

func main() {

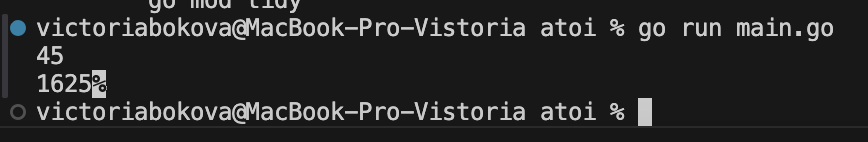
var n string

fmt.Scan(&n)

for i := 0; i < len(n); i++ {

fmt.Print((n[i] - 48) \* (n[i] - 48))

}



E) Пятое задание: Требуется вычислить период колебаний математического

маятника, для этого нужно найти циклическую частоту колебания пружинного

маятника, в формуле w встречается масса которую также нужно найти, все нужные

формулы приведены в задание на Stepik.

Решение:

package main

import (

"fmt"

"math"

"strconv"

)

// Определение функций вне main

func M(p, v float64) float64 {

return p \* v

}

func W(k, m float64) float64 {

return math.Sqrt(k / m)

}

func T(w float64) float64 {

return 6 / w

}

func main() {

var k, p, v float64

// Ввод значений с консоли

fmt.Print("Введите значение k: ")

var inputK string

fmt.Scanln(&inputK)

k, \_ = strconv.ParseFloat(inputK, 64)

fmt.Print("Введите значение p: ")

var inputP string

fmt.Scanln(&inputP)

p, \_ = strconv.ParseFloat(inputP, 64)

fmt.Print("Введите значение v: ")

var inputV string

fmt.Scanln(&inputV)

v, \_ = strconv.ParseFloat(inputV, 64)

// Вычисления и вывод результата

m := M(p, v)

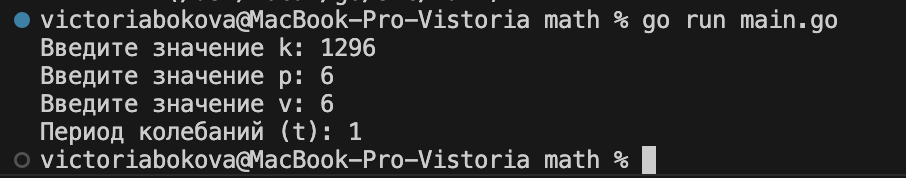
w := W(k, m)

t := T(w)

fmt.Println("Период колебаний (t):", t)

}

**Тестирование:**



4. Отчет сделан и помещен в docs.

5. Командой git add . и git commit -m были зафиксированы и проиндексированы

изменения. И командой git push origin dev запушили локальную ветку dev в

удаленный репозиторий.

6. Интерфейсом GitHub создан Pull request из dev в master.

Заключение: Познакомились с многопоточным компилируемым языком

программирования GoLang. Выполнили 5 заданий на базовые навыки работы с

этим языком

Список источников:

• Сайт: https://stepik.org/

• Cайт: https://go.dev/