



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

О Т Ч Е Т

по лабораторной работе № 3

Название: Основы Golang

Дисциплина: Языки интернет-программирования

Студент

ИУ6-31Б

(Группа)

(Подпись, дата)

Д.Ю. Воронин

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

В.Д. Шульман

(И.О. Фамилия)

Москва, 2024

Цель работы – знакомство с Go, компилируемым многопоточным языком программирования.

Задание:

1. Ознакомьтесь с первыми 2-я разделами курса
<https://stepik.org/course/54403/info>
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
7. Защитите лабораторную работу...

Ход работы

1. Задача triangle

Условие:

На вход подаются a и b - катеты прямоугольного треугольника. Нужно найти длину гипотенузы

Sample Input:

6 8

Sample Output:

10

Рисунок 1 – Условие задачи triangle

Решение:

```
package main
import (
    "fmt"
```

```

    "math"
)

func main() {
    var a, b float64
    fmt.Scan(&a, &b)
    fmt.Println(math.Sqrt(a*a + b*b))
}

```

Тестирование

```

admin@MBP-admin math % go run "/Users/admin/Desktop/WEB/web-3/projects/triangle/main.go"
6 8
10

```

Рисунок 2 – Тестирование задачи triangle

2. Задача star

Условие:

Дана строка, содержащая только английские буквы (большие и маленькие). Добавить символ '*' (звездочка) между буквами (перед первой буквой и после последней символ '*' добавлять не нужно).

Входные данные

Вводится строка ненулевой длины. Известно также, что длина строки не превышает 1000 знаков.

Выходные данные

Вывести строку, которая получится после добавления символов '*'.

Sample Input:

LItBeoFLcSGB0FQxMHoIuDDWcqcVgkcRoAeocX0

Sample Output:

L*I*t*B*e*o*F*L*c*S*G*B*0*F*Q*x*M*H*o*I*u*D*W*c*q*c*V*g*k*c*R*o*A*e*o*c*X*0

Рисунок 3 – Условие задачи star

Решение:

```

package main

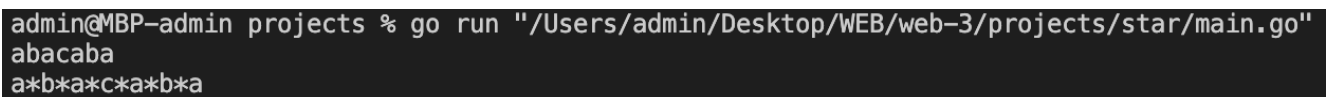
import (
    "fmt"
    "unicode/utf8" // 1 вариант
    //"strings" // 2 вариант

```

)

```
func main() {  
    var str, res string  
    fmt.Scan(&str)  
    for index, symbol := range str { // 1 вариант с циклом  
        res += string(symbol)  
        if index != utf8.RuneCountInString(str)-1 {  
            res += string('*')  
        }  
    }  
    //res = strings.Join(strings.Split(str, ""), string('*')) // 2 вариант  
    fmt.Println(res)  
}
```

Тестирование



```
admin@MBP-admin projects % go run "/Users/admin/Desktop/WEB/web-3/projects/star/main.go"  
abacaba  
a*b*a*c*a*b*a
```

Рисунок 4 – Тестирование задачи star

3. Задача digits

Условие:

Дана строка, содержащая только арабские цифры. Найти и вывести наибольшую цифру.

Входные данные

Вводится строка ненулевой длины. Известно также, что длина строки не превышает 1000 знаков и строка содержит только арабские цифры.

Выходные данные

Выведите максимальную цифру, которая встречается во введенной строке.

Sample Input:

1112221112

Sample Output:

2

Рисунок 5 – Условие задачи digits

Решение:

```

package main

import (
    "fmt"
    "strconv"
)

func main() {
    var str string
    var maxNumber int
    maxNumber = 0
    fmt.Scan(&str)
    for _, symbol := range str {
        i, err := strconv.Atoi(string(symbol))
        if err != nil {
            panic(err)
        }
        maxNumber = max(maxNumber, i)
    }
    fmt.Println(maxNumber)
}

```

Тестирование

```

admin@MBP-admin projects % go run "/Users/admin/Desktop/WEB/web-3/projects/digits/main.go"
123454321132
5

```

Рисунок 6 – Тестирование задачи digits

4. Задача atoi

Условие:

На вход подается целое число. Необходимо возвести в квадрат каждую цифру числа и вывести получившееся число.

Например, у нас есть число 9119. Первая цифра - 9. 9 в квадрате - 81. Дальше 1. Единица в квадрате - 1. В итоге получаем 811181

Sample Input:

9119

Sample Output:

811181

Рисунок 7 – Условие задачи atoi

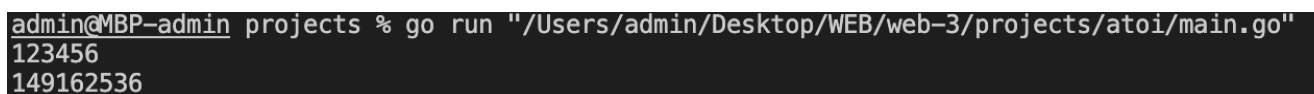
Решение:

```
package main

import (
    "fmt"
    "strconv"
)

func main() {
    var str, res string
    fmt.Scan(&str)
    for _, symbol := range str {
        i, err := strconv.Atoi(string(symbol))
        if err != nil {
            panic(err)
        }
        res += strconv.Itoa(i * i)
    }
    fmt.Println(res)
}
```

Тестирование



```
admin@MBP-admin projects % go run "/Users/admin/Desktop/WEB/web-3/projects/atoi/main.go"
123456
149162536
```

Рисунок 8 – Тестирование задачи atoi

5. Задача math

Условие:

Требуется вычислить период колебаний (t) математического маятника (мы округлили некоторые значения для удобства проверки), для этого нужно найти циклическую частоту колебания пружинного маятника (w), в формуле w встречается масса которую также нужно найти, все нужные формулы приведены ниже:

Напишите три функции, каждая из которых будет выполнять конкретную формулу. Название функций обязательно должны соответствовать букве формулы: $T()$, $W()$ и $M()$. Для того чтобы найти t - необходимо сначала найти w , и т.д. Так что используйте результат функции $W()$ в формуле функции $T()$ - то-есть вызывайте функцию $W()$ в $T()$. Аналогично и с $W()$, $M()$.

$$t = \frac{6}{w}, w = \sqrt{\frac{k}{m}}, m = p * v$$

ВАЖНО! Считайте, что пакет `main` уже объявлен, а также функция `main()` с вызовом ВАШЕЙ будущей функции $T()$ уже есть. Несмотря на то, что тестирование будет через ввод-вывод, вам НЕ требуется вводить и выводить что-либо. Для подсчета используйте УЖЕ ВВЕДЕННЫЕ ГЛОБАЛЬНЫЕ переменные k, p, v ТИПА `float64` !!!

Пакет `math` уже импортирован! Напоминаю: корень (`sqrt`) можно найти с помощью пакета `"math"`, например:

```
fmt.Println(math.Sqrt(9))  
// результат: 3
```

Sample Input:

1296 6 6

Sample Output:

1

Рисунок 9 – Условие задачи math

Решение:

```
package main
```

```
import (  
    "fmt"  
    "math"  
)
```

```
var k, p, v float64
```

```
// Начало решения задачи
```

```
func T() float64 {  
    return 6 / W()  
}
```

```
func W() float64 {  
    return math.Sqrt(k / M())  
}
```

```
func M() float64 {  
    return p * v  
}
```

// Конец решения

```
func main() {  
    fmt.Scan(&k, &p, &v)  
    fmt.Println(T())  
}
```

Тестирование

```
admin@MBP-admin projects % go run "/Users/admin/Desktop/WEB/web-3/projects/math/main.go"  
1296 6 6  
1
```

Рисунок 10 – Тестирование задачи math

6. Загрузим решения на GitHub и сделаем Pull request из dev в master с помощью интерфейса GitHub.

Заключение:

Язык программирования Golang предоставляет возможность работы с консолью, строками и математическими операциями для решения соответствующих задач.

Список использованных источников:

<https://github.com/ValeryBMSTU/web-3>

<https://stepik.org/course/54403/info>