



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

ИУ6-32Б

(Группа)

29.09.2024

(Подпись, дата)

Л.И. Заушников

(И.О. Фамилия)

Преподаватель

29.09.2024

(Подпись, дата)

В.Д. Шульман

(И.О. Фамилия)

Москва, 2024

Цель работы: познакомиться с компилируемым многопоточным языком программирования Golang.

Ход работы

1. Первым делом было проведено ознакомление с первыми 2-я разделами курса <https://stepik.org/course/54403/info> .

1)"Введение в основы"

2) "Функции, структуры, указатели и другое"

2. Далее был сделан форк данного репозитория в GitHub (рисунок 1), клонирована получившаяся копия локально (рисунок 2), создана от мастера ветку dev и переключение на нее (рисунок 3):

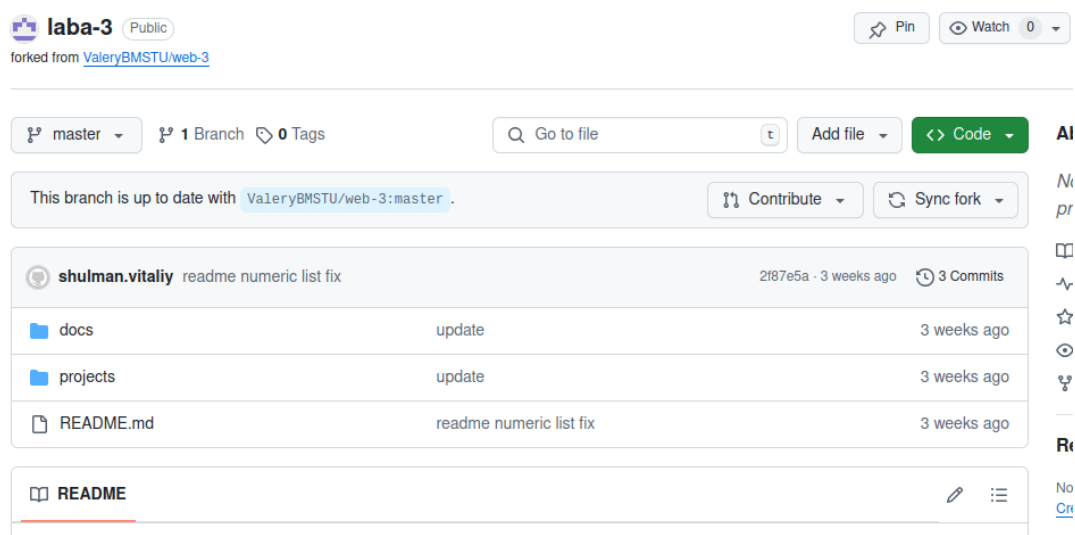
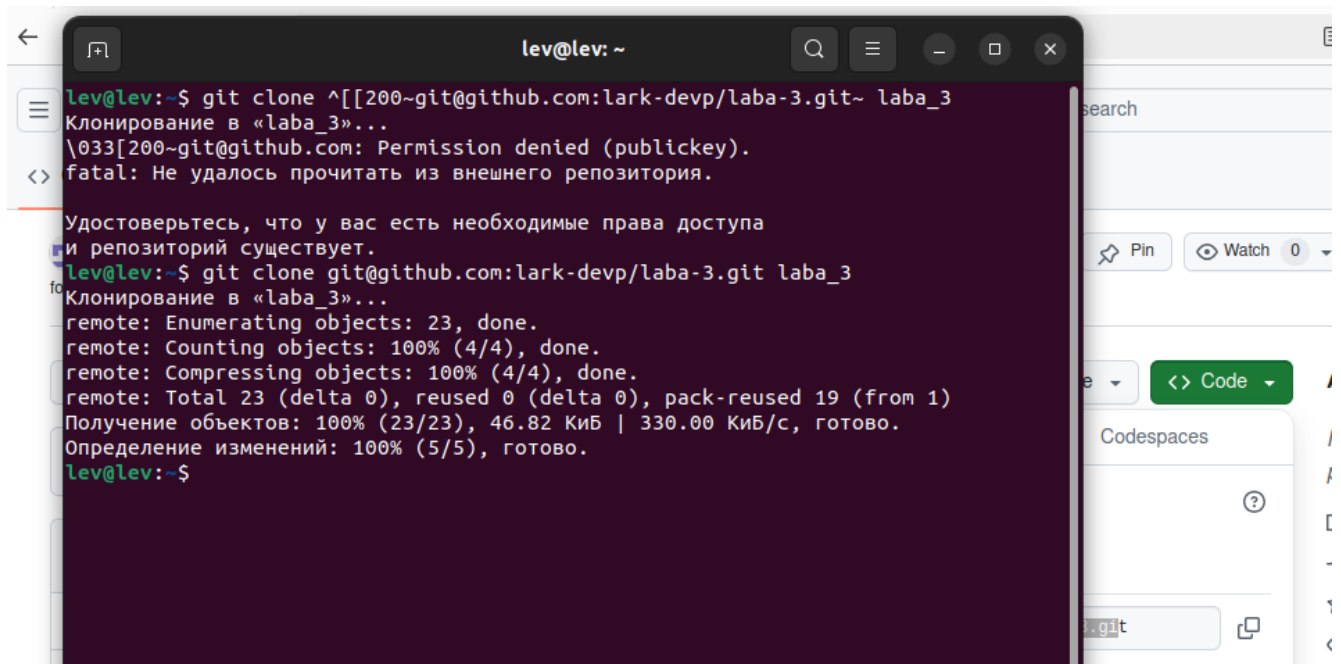
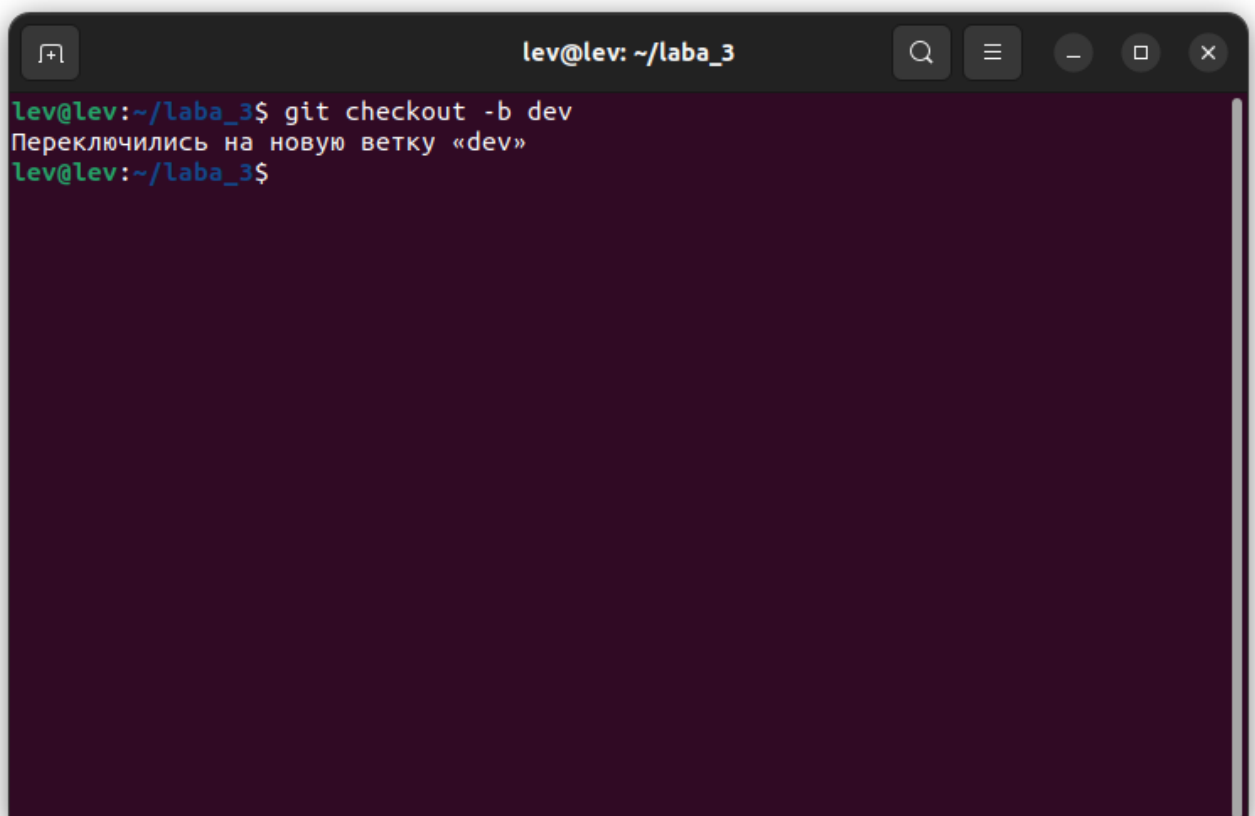


Рисунок 1 – Fork репозитория



```
lev@lev: ~  
lev@lev:~$ git clone ^[[200~git@github.com:lark-devp/laba-3.git~ laba_3  
Клонирование в «laba_3»...  
\\033[200~git@github.com: Permission denied (publickey).  
<> fatal: Не удалось прочитать из внешнего репозитория.  
  
Удостоверьтесь, что у вас есть необходимые права доступа  
и репозиторий существует.  
lev@lev:~$ git clone git@github.com:lark-devp/laba-3.git laba_3  
Клонирование в «laba_3»...  
remote: Enumerating objects: 23, done.  
remote: Counting objects: 100% (4/4), done.  
remote: Compressing objects: 100% (4/4), done.  
remote: Total 23 (delta 0), reused 0 (delta 0), pack-reused 19 (from 1)  
Получение объектов: 100% (23/23), 46.82 КиБ | 330.00 КиБ/с, готово.  
Определение изменений: 100% (5/5), готово.  
lev@lev:~$
```

Рисунок 2 - Локальное клонирование копии



```
lev@lev: ~/laba_3  
lev@lev:~/laba_3$ git checkout -b dev  
Переключились на новую ветку «dev»  
lev@lev:~/laba_3$
```

Рисунок 3 - Переключение на ветку dev

Далее были решены 5 задач.

Задача 1

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

Код на Golang для решения задачи представлен ниже.

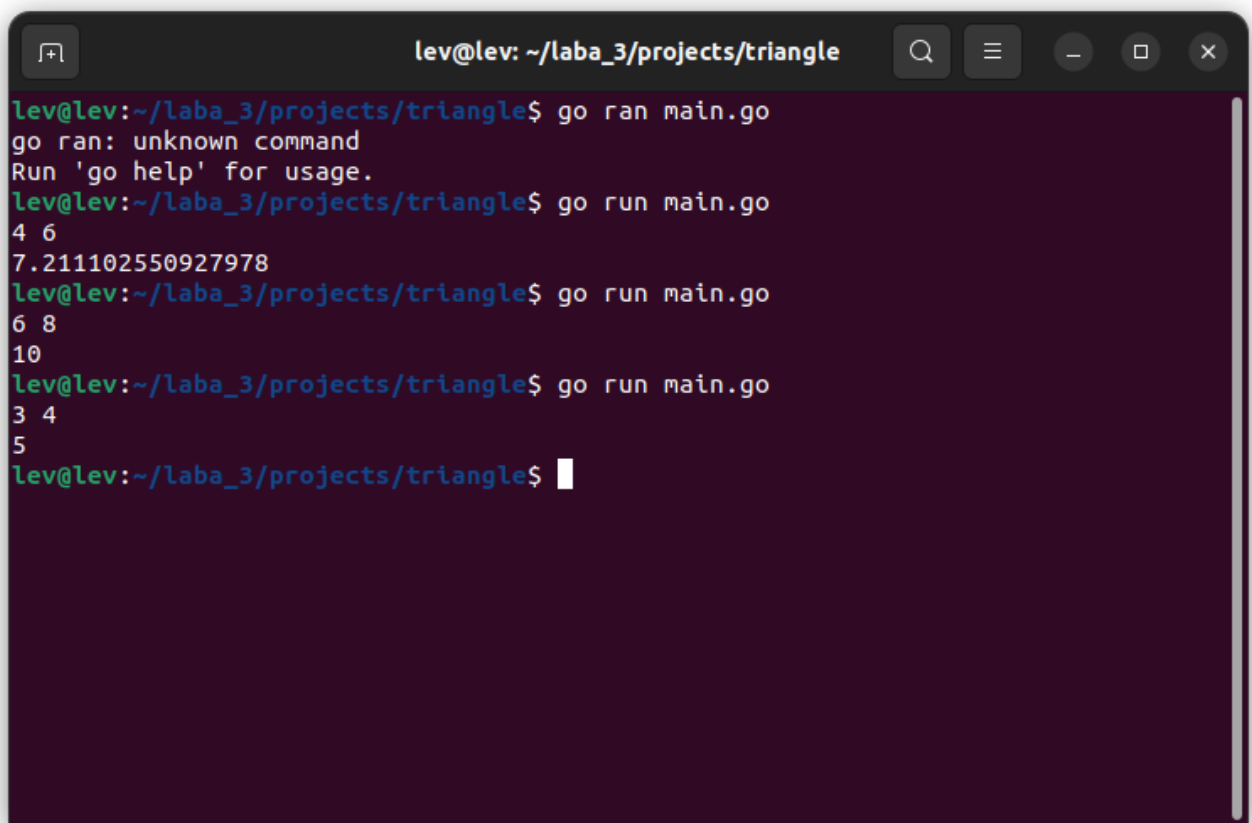
```
package main
```

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

```
func Gepot(a float64, b float64) float64 {  
    var c float64 = math.Sqrt(a*a+b*b)  
    return c  
}
```

```
func main() {  
    var a, b float64  
    fmt.Scanf("%f %f", &a, &b)  
    fmt.Println(Gepot(a,b))  
}
```

Результаты тестирования представлены на рисунке 4.

A terminal window with a dark background and light-colored text. The title bar at the top reads 'lev@lev: ~/laba_3/projects/triangle'. The terminal shows the following sequence of commands and outputs:

```
lev@lev:~/laba_3/projects/triangle$ go ran main.go
go ran: unknown command
Run 'go help' for usage.
lev@lev:~/laba_3/projects/triangle$ go run main.go
4 6
7.211102550927978
lev@lev:~/laba_3/projects/triangle$ go run main.go
6 8
10
lev@lev:~/laba_3/projects/triangle$ go run main.go
3 4
5
lev@lev:~/laba_3/projects/triangle$
```

Рисунок 4 - Тестирование программы для решения задачи 1

Задача 2

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

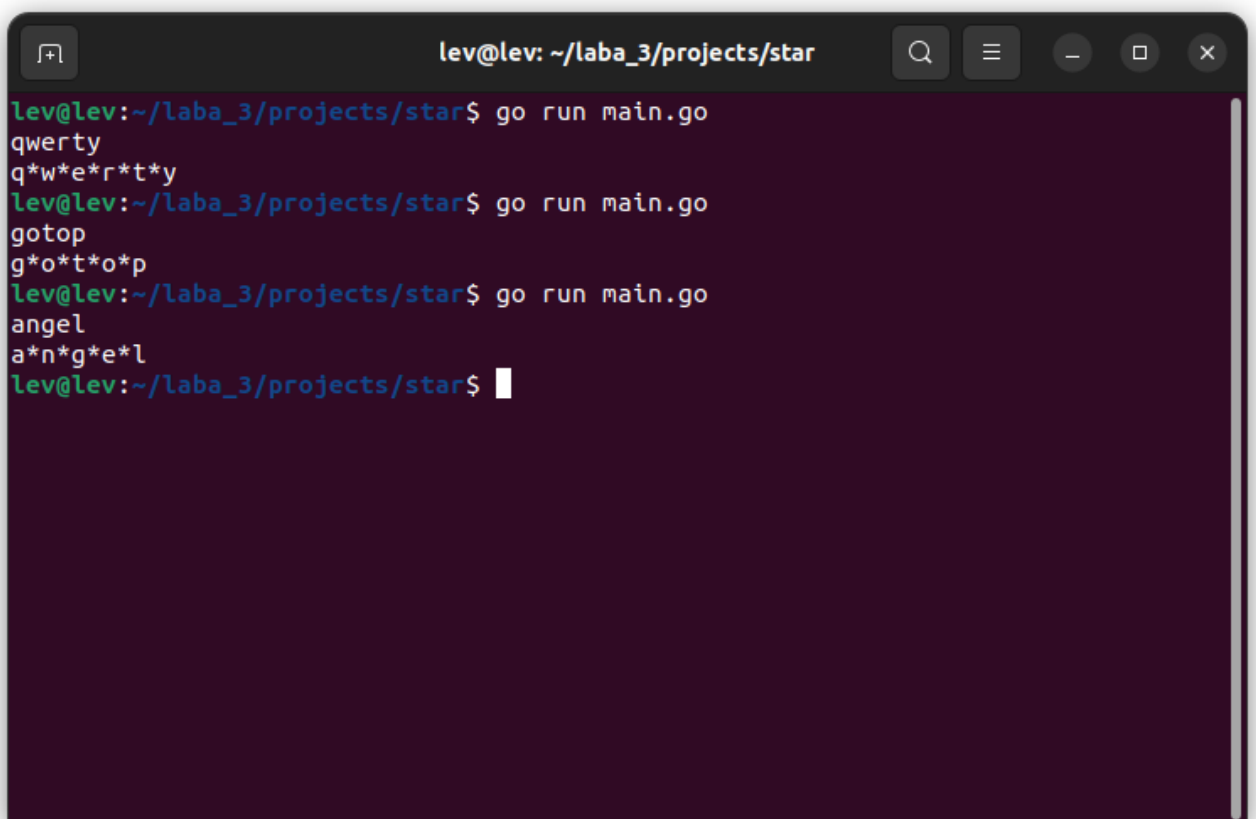
Код на Golang для решения задачи представлен ниже

```
package main
```

```
import (  
    "fmt"  
    "strings"  
)
```

```
func main() {  
    var str string  
    fmt.Scan(&str)  
  
    res:= strings.Join(strings.Split(str,""),"*")  
  
    fmt.Println(res)  
}
```

Результаты тестирования представлены на рисунке 5.



```
lev@lev:~/laba_3/projects/star$ go run main.go  
qwerty  
q*w*e*r*t*y  
lev@lev:~/laba_3/projects/star$ go run main.go  
g*o*t*o*p  
lev@lev:~/laba_3/projects/star$ go run main.go  
angel  
a*n*g*e*l  
lev@lev:~/laba_3/projects/star$
```

Рисунок 5 - Тестирование программы для решения задачи 2

Задача 3

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

Код на Golang для решения задачи представлен ниже

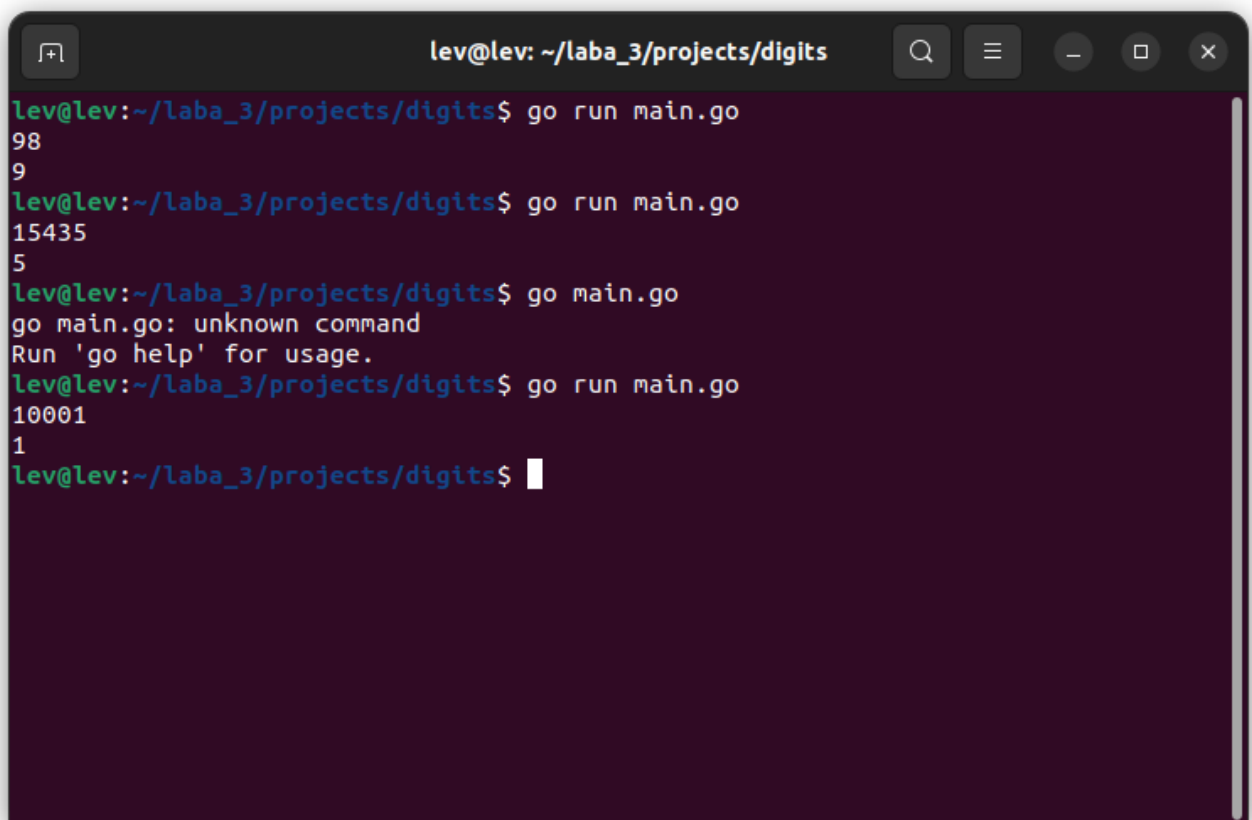
```
package main

import "fmt"

func getMaxDigit(str string) int{
    maxD := 0
    for _, elem := range str{
        d := int(elem - '0')
        if d > maxD{
            maxD = d
        }
    }
    return maxD
}

func main() {
    var s string
    _, _ = fmt.Scan(&s)
    fmt.Println(getMaxDigit(s))
}
```

Результаты тестирования представлены на рисунке 6.

A terminal window with a dark background and light-colored text. The window title is 'lev@lev: ~/laba_3/projects/digits'. The terminal shows the following sequence of commands and outputs:

```
lev@lev:~/laba_3/projects/digits$ go run main.go
98
9
lev@lev:~/laba_3/projects/digits$ go run main.go
15435
5
lev@lev:~/laba_3/projects/digits$ go main.go
go main.go: unknown command
Run 'go help' for usage.
lev@lev:~/laba_3/projects/digits$ go run main.go
10001
1
lev@lev:~/laba_3/projects/digits$
```

Рисунок 6 - Тестирование программы для решения задачи 3

Задача 4

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.

Код на Golang для решения задачи представлен ниже.

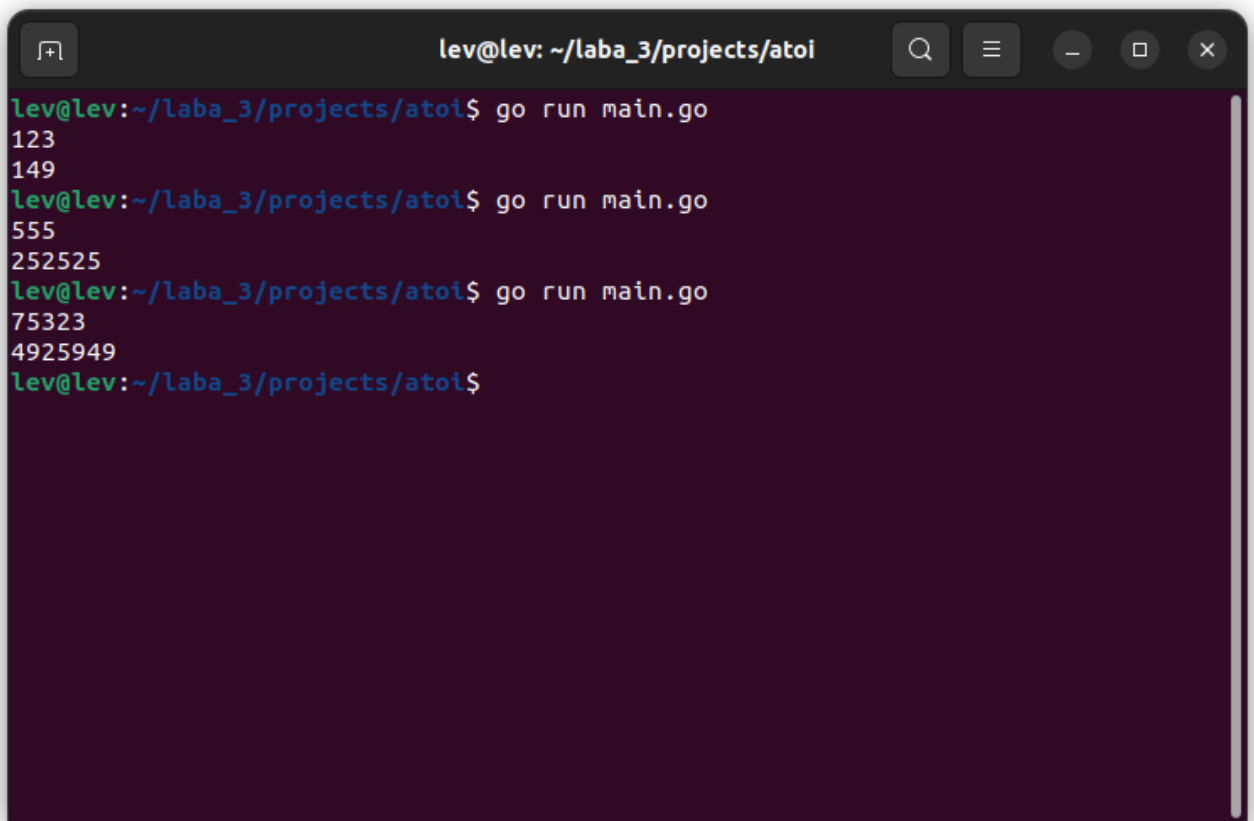
```
package main
```

```
import "fmt"
```



```
func main() {  
  
    var str string  
  
    _, _ = fmt.Scan(&str)  
  
    allToSquare(str)  
  
    fmt.Println()  
  
}  
  
func allToSquare(s string) {  
  
    for _, elem := range s {  
  
        num := int(elem-'0')  
  
        res := num*num  
  
        fmt.Print(res)  
  
    }  
  
}
```

Результаты тестирования представлены на рисунке 7.

A terminal window with a dark background and light-colored text. The window title is 'lev@lev: ~/laba_3/projects/atoi'. The terminal shows three successful runs of a Go program, each printing a number. The first run prints '123', the second '149', and the third '555'. The fourth run prints '252525', the fifth '75323', and the sixth '4925949'. The prompt 'lev@lev:~/laba_3/projects/atoi\$' is visible at the end of each line.

```
lev@lev:~/laba_3/projects/atoi$ go run main.go
123
149
lev@lev:~/laba_3/projects/atoi$ go run main.go
555
252525
lev@lev:~/laba_3/projects/atoi$ go run main.go
75323
4925949
lev@lev:~/laba_3/projects/atoi$
```

Рисунок 7 - Тестирование программы для решения задачи 4

Задача 5

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().

Код на Golang для решения задачи представлен ниже.

```
package main

import (
    "fmt"
    "math"
)

func T( k float64, p float64, v float64) float64 {
    var t float64 = 6/W(k,p,v)
    return t
}

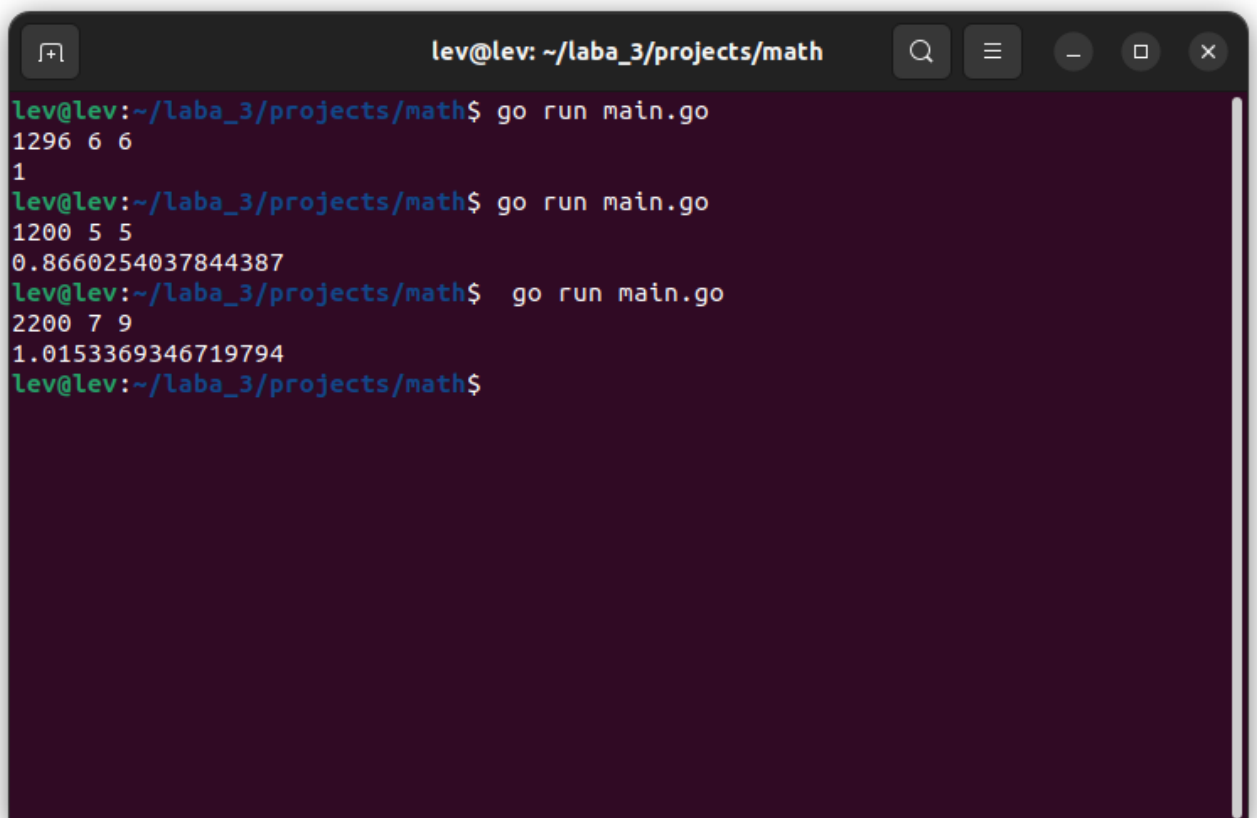
func M(p float64, v float64) float64 {
    var m float64 = p*v
    return m
}

func W (k float64, p float64, v float64) float64 {
    var w float64 = math.Sqrt(k / M(p,v))
    return w
}

func main() {
    var res, n1, n2, n3 float64
```

```
fmt.Scanf("%f %f %f", &n1, &n2, &n3)
res = T(n1,n2,n3)
fmt.Println(res)
}
```

Результаты тестирования представлены на рисунке 8.

A screenshot of a terminal window with a dark purple background. The window title is 'lev@lev: ~/laba_3/projects/math'. The terminal shows three runs of 'go run main.go'. Each run outputs three numbers: a large integer, a small integer, and a floating-point number. The first run outputs '1296 6 6' and '1'. The second run outputs '1200 5 5' and '0.8660254037844387'. The third run outputs '2200 7 9' and '1.0153369346719794'.

```
lev@lev:~/laba_3/projects/math$ go run main.go
1296 6 6
1
lev@lev:~/laba_3/projects/math$ go run main.go
1200 5 5
0.8660254037844387
lev@lev:~/laba_3/projects/math$ go run main.go
2200 7 9
1.0153369346719794
lev@lev:~/laba_3/projects/math$
```

Рисунок 8 - Тестирование программы для решения задачи 5

Заключение

В ходе выполнения лабораторной работы были изучены основы Golang и решены задачи для отработки полученных знаний.

Список источников

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