

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.01 Информатика и вычислительная техника**

по лабораторной работе № 3

**Дисциплина: Языки Интернет-программирования**

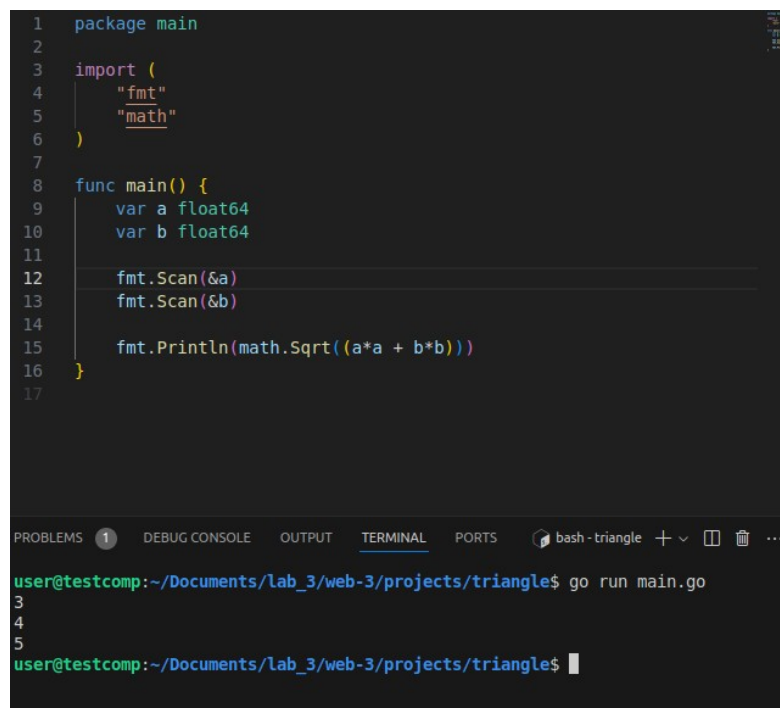
Москва, 2024

**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 dev 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 dev в удаленный репозиторий GitHub
6. Через интерфейс GitHub создайте Pull Request dev --> master
7. Защитите лабораторную работу...

Ход работы



```
1 package main
2
3 import (
4     "fmt"
5     "math"
6 )
7
8 func main() {
9     var a float64
10    var b float64
11
12    fmt.Scan(&a)
13    fmt.Scan(&b)
14
15    fmt.Println(math.Sqrt((a*a + b*b)))
16 }
17
```

PROBLEMS 1 DEBUG CONSOLE OUTPUT TERMINAL PORTS bash - triangle + - [ ] [ ] ...

```
user@testcomp:~/Documents/lab_3/web-3/projects/triangle$ go run main.go
3
4
5
user@testcomp:~/Documents/lab_3/web-3/projects/triangle$
```

Рисунок 1 – Результат выполнения программы triangle

```
star / main.go / main
1 package main
2
3 import (
4     "fmt"
5     "strings"
6 )
7
8 func main() {
9     var a string
10
11     fmt.Scan(&a)
12
13     res := strings.Join(strings.Split(a, ""), "*")
14
15     fmt.Println(res)
16 }
17
```

PROBLEMS 1 DEBUG CONSOLE OUTPUT TERMINAL PORTS bash - star + v [ ] [ ] ... ^ x

- user@testcomp:~/Documents/lab\_3/web-3/projects/triangle\$ cd ~/Documents/lab\_3/web-3/projects/star
- user@testcomp:~/Documents/lab\_3/web-3/projects/star\$ go run main.go  
fdfdfefewfwefwfwef  
f\*d\*f\*d\*f\*e\*f\*e\*w\*f\*w\*f\*e\*f\*w\*f\*w\*e\*f
- user@testcomp:~/Documents/lab\_3/web-3/projects/star\$

Рисунок 2 – Результат выполнения программы star

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var a string
7     fmt.Scan(&a)
8
9     max := '0'
10
11     for _, sym := range a {
12         if sym > max {
13             max = sym
14         }
15     }
16
17     fmt.Println(string(max))
18 }
19
```

PROBLEMS 2 DEBUG CONSOLE OUTPUT TERMINAL PORTS bash - digits + ▢ 🗑️ ...

● user@testcomp:~/Documents/lab\_3/web-3/projects/digits\$ go run main.go  
12748369645  
9

○ user@testcomp:~/Documents/lab\_3/web-3/projects/digits\$

Рисунок 3 – Результат выполнения программы digits

```
1 package main
2
3 import "fmt"
4
5 func main() {
6
7     var a string
8     fmt.Scan(&a)
9
10    for i := 0; i < len(a); i++ {
11        fmt.Print((a[i] - '0') * (a[i] - '0'))
12    }
13
14 }
15
```

PROBLEMS 2 DEBUG CONSOLE OUTPUT TERMINAL PORTS bash - atoi + - [ ] [ ] ... ^ x

• user@testcomp:~/Documents/lab\_3/web-3/projects/atoi\$ go run main.go  
1234  
○ 14916user@testcomp:~/Documents/lab\_3/web-3/projects/atoi\$

Рисунок 4 – Результат выполнения программы atoi

```
math > main.go
1 func T() float64 {
2     return 6/W()
3 }
4
5 func W() float64 {
6     return math.Sqrt(k/M())
7 }
8
9 func M() float64 {
10    return p * v
11 }
```

Рисунок 5 – Программа math

**Заключение** — были изучены базовые механизмы языка программирования Go, при помощи них изучение дальнейшего материала пойдет легче.