

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ
ФЕДЕРАЦИИ**
**Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

**Отчет по лабораторной работе № 1
по дисциплине «Программная инженерия»**

Выполнила студентка группы ПИЖ-б-о-
21-1, 3 курса, очной формы обучения
Коновалова Виктория Николаевна
« _ » _____ 2024г.

Ставрополь 2024 г.

ЛАБОРАТОРНАЯ РАБОТА 1

Цель: исследование назначения и способов установки Go, исследование типов данных, констант и арифметических операции языка программирования Go.

Ход работы

```
package main

import "fmt"

func main() {
    fmt.Println("Hello, Go!")
}
```

Рисунок – Код программы

```
PS C:\Users\vika1\OneDrive\Desktop\univer\sem6\Проги\1\prim\1> go run 1.go
Hello, Go!
```

Рисунок – Результат программы

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     fmt.Println(string("Hello Go"[0])) // вывод: H
7 }
8
```

Рисунок – Код программы

```
PS C:\Users\vika1\OneDrive\Desktop\univer\sem6\Проги\1\prim\2> go run 2.go
H
```

Рисунок – Результат программы

```

package main

import "fmt"

func main() {
    var hello string
    hello = "Hello Go!"
    var a int = 2019
    fmt.Println(hello)
    fmt.Println(a)
}

```

Рисунок – Код программы

```

PS C:\Users\vika1\OneDrive\Desktop\univer\sem6\ПроИн\1\prim\3> go run 3.go
Hello Go!
2019

```

Рисунок – Результат программы

```

package main

import "fmt"

func main() {
    var (
        name string = "Dima"
        age  int     = 23
    )
    fmt.Println(name)
    fmt.Println(age)
}

```

Рисунок – Код программы

```
PS C:\Users\vika1\OneDrive\Desktop\univer\sem6\ПроИн\1\prim\4> go run 4.go
Dima
23
```

Рисунок – Результат программы

```
package main

import "fmt"


func main() {
    var name string
    var age int
    fmt.Print("Введите имя: ")
    fmt.Scan(&name)
     fmt.Print("Введите возраст: ")
    fmt.Scan(&age)
    fmt.Println(name, age)
}
```

Рисунок – Код программы

```
PS C:\Users\vika1\OneDrive\Desktop\univer\sem6\ПроИн\1\prim\5> go run 5.go
Введите имя: Вика
Введите возраст: 21
Вика 21
```

Рисунок – Результат программы

```
package main

import "fmt"

func main() {
    name := "Ivan"
    age := 27
    fmt.Println("My name is", name, "and I am", age, "years old.")
}
```

Рисунок – Код программы

```
PS C:\Users\vika1\OneDrive\Desktop\univer\sem6\ПроИн\1\prim\6> go run 6.go
My name is Ivan and I am 27 years old.
```

Рисунок – Результат программы

```
/*
Первая программа
на языке Go
*/
package main // определение пакета для текущего файла
import "fmt" // подключение пакета fmt
// определение функции main
func main() {
    fmt.Println("Hello Go!") // вывод строки на консоль
}
```

Рисунок – Код программы

```
PS C:\Users\vika1\OneDrive\Desktop\univer\sem6\ПроИн\1\prim\7> go run 7.go
Hello Go!
```

Рисунок – Результат программы

```
package main

import (
    "fmt"
)

const (
    A int = 45
    B
    C float32 = 3.3
    D
)

func main() {
    fmt.Println(A, B, C, D) // Вывод: 45 45 3.3 3.3
}
```

Рисунок –

Код программы

```
PS C:\Users\vika1\OneDrive\Desktop\univer\sem6\ПроИн\1\prim\8> go run 8.go
45 45 3.3 3.3
```

Рисунок – Результат программы

```
package main

import "fmt"

func main() {
    fmt.Println("I like Go!")
}
```

Рисунок – Код программы

```
PS C:\Users\vika1\OneDrive\Desktop\univer\sem6\ПроИн\1\com\1> go run 1.go
I like Go!
```

Рисунок – Результат программы

```
package main

import "fmt"

func main() {
    fmt.Println("I like Go!")
    fmt.Println("I like Go!")
    fmt.Print("I like Go!")
}
```

Рисунок – Код программы

```
PS C:\Users\vika1\OneDrive\Desktop\univer\sem6\ПроИн\1\com\2> go run 2.go
I like Go!
I like Go!
I like Go!
```

Рисунок – Результат программы

```

package main

import (
    "fmt"
)

func main() {
    var input int
    fmt.Print("Введите число: ")
    fmt.Scan(&input)

    result := input * 2
    result += 100

    fmt.Print(result)
}

```

Рисунок – Код программы

```

PS C:\Users\vika1\OneDrive\Desktop\univer\sem6\Проги\1\com\3> go run 3.go
Введите число: 1
102

```

Рисунок – Результат программы

```

package main

import "fmt"

func main() {
    var a, b, c int
    fmt.Println("Введите два числа:")
    fmt.Scan(&a) // считываем значение первого числа
    fmt.Scan(&b) // считываем значение второго числа

    a = a * a // находим квадрат первого числа
    b = b * b // находим квадрат второго числа
    c = a + b // находим сумму квадратов двух чисел

    fmt.Println("Результат:", c)
}

```

Рисунок –

Код программы

```
PS C:\Users\vika1\OneDrive\Desktop\univer\sem6\ПроИн\1\com\4> go run 4.go
Введите два числа:
2
2
Результат: 8
```

Рисунок – Результат программы

```
package main

import "fmt"

func main() {
    var num int
    fmt.Println("Введите целое число:")
    fmt.Scan(&num)

    square := num * num

    fmt.Println("Квадрат числа:", square)
}
```

Рисунок – Код программы

```
PS C:\Users\vika1\OneDrive\Desktop\univer\sem6\ПроИн\1\com\5> go run 5.go
Введите целое число:
3
Квадрат числа: 9
```

Рисунок – Результат программы


```

package main

import "fmt"

func main() {
    var num int
    fmt.Println("Введите натуральное число:")
    fmt.Scan(&num)

    lastDigit := num % 10

    fmt.Println("Последняя цифра:", lastDigit)
}

```

Рисунок – Код программы

```

PS C:\Users\vika1\OneDrive\Desktop\univer\sem6\Программы\1\com\6> go run 6.go
Введите натуральное число:
123
Последняя цифра: 3

```

Рисунок – Результат программы

```

package main

import "fmt"

func main() {
    var num int
    fmt.Println("Введите неотрицательное натуральное число:")
    fmt.Scan(&num)

    // Убираем последнюю цифру (единицы)
    num = num / 10

    // Находим новую последнюю цифру (десятки)
    tens := num % 10

    fmt.Println("Число десятков:", tens)
}

```

Рисунок – Код программы

```
PS C:\Users\vika1\OneDrive\Desktop\univer\sem6\ПрогиИн\1\com\7> go run 7.go
Введите неотрицательное натуральное число:
2010
Число десятков: 1
```

Рисунок – Результат программы

```
package main

import "fmt"

func main() {
    var d int
    fmt.Println("Введите угол поворота часовой стрелки (0 < d < 360):")
    fmt.Scan(&d)

    // Находим количество часов
    hours := d / 30

    // Находим количество минут
    minutes := (d % 30) * 2

    fmt.Printf("It is %d hours %d minutes.\n", hours, minutes)
}
```

Рисунок – Код программы

```
PS C:\Users\vika1\OneDrive\Desktop\univer\sem6\ПрогиИн\1\com\8> go run 8.go
Введите угол поворота часовой стрелки (0 < d < 360):
90
It is 3 hours 0 minutes.
```

Рисунок – Результат программы

```
package main

import "fmt"

func main() {
    var a2 int = 10
    a2 = a2 * 10
    fmt.Println(a2)
}
```

Рисунок – Код программы

```
PS C:\Users\vika1\OneDrive\Desktop\univer\sem6\ПроИн\1\com\9> go run 9.go
100
```

Рисунок – Результат программы

```
package main

import "fmt"

func main() {
    var a int = 8
    const b int = 10
    a = a + b
    fmt.Println(a)
}
```

Рисунок – Код программы

```
PS C:\Users\vika1\OneDrive\Desktop\univer\sem6\ПроИн\1\com\10> go run 10.go
18
```

Рисунок – Результат программы

```

package main

import (
    "fmt"
    "math"
)

func main() {
    var a, b float64
    fmt.Println("Введите значения параметров а и b:")
    fmt.Scan(&a, &b)

    // Площадь поверхности
    surfaceArea := 2*math.Pi*b*b + 2*math.Pi*a*b

    // Объем тела
    volume := (4.0 / 3.0) * math.Pi * a * b * b

    fmt.Printf("Площадь поверхности: %.2f\n", surfaceArea)
    fmt.Printf("Объем тела: %.2f\n", volume)
}

```

Рисунок – Код программы

```

PS C:\Users\vika1\OneDrive\Desktop\univer\sem6\ПроИн\1\com\11> go run 11.go
Введите значения параметров а и b:
5
6
Площадь поверхности: 414.69
Объем тела: 753.98

```

Рисунок – Результат программы

```

/* Объем и площадь поверхности куба: Задайте переменную для длины
ребра куба. Рассчитайте и выведите объем и площадь его поверхности.
*/

package main

import (
    "fmt"
    "math"
)

func main() {
    var sideLength float64
    fmt.Println("Введите длину ребра куба:")
    fmt.Scan(&sideLength)

    // Вычисляем объем куба
    volume := math.Pow(sideLength, 3)

    // Вычисляем площадь его поверхности
    surfaceArea := 6 * math.Pow(sideLength, 2)

    // Выводим результаты
    fmt.Printf("Объем куба: %.2f\n", volume)
    fmt.Printf("Площадь поверхности куба: %.2f\n", surfaceArea)
}

```

Рисунок – Код для первого индивидуально

```

PS C:\Users\vika1\OneDrive\Desktop\univer\sem6\Проги\1\ind\1> go run 1.go
Введите длину ребра куба:
4
Объем куба: 64.00
Площадь поверхности куба: 96.00

```

Рисунок – Результат программы

Для вычисления периметра и площади треугольника, заданного координатами своих вершин, мы можем воспользоваться формулами для нахождения расстояния между точками и площади треугольника по формуле Герона.

Начнем с вычисления расстояния между двумя точками. Для этого мы можем использовать формулу расстояния между двумя точками на плоскости:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Где (x_1, y_1) и (x_2, y_2) - координаты точек

После того, как мы найдем длины всех трех сторон треугольника, мы можем вычислить его периметр, сложив длины всех сторон.

Для вычисления площади треугольника по формуле Герона, нам сначала нужно вычислить полупериметр

$$P = (a + b + c) / 2$$

где a, b, c - длины сторон треугольника. После этого мы можем использовать формулу Герона для нахождения площади треугольника:

$$S = \sqrt{p \cdot (p - a) \cdot (p - b) \cdot (p - c)}$$

```
/* Треугольник задан координатами своих вершин. Найти периметр и площадь
треугольника.
*/

package main

import (
    "fmt"
    "math"
)

func main() {
    var x1, y1, x2, y2, x3, y3 float64
    fmt.Println("Введите координаты вершин треугольника (x1 y1 x2 y2 x3 y3):")
    fmt.Scan(&x1, &y1, &x2, &y2, &x3, &y3)

    // Вычисляем длины сторон треугольника
    a := distance(x1, y1, x2, y2)
    b := distance(x2, y2, x3, y3)
    c := distance(x3, y3, x1, y1)

    // Вычисляем периметр
    perimeter := a + b + c

    // Вычисляем полупериметр для формулы Герона
    p := perimeter / 2

    // Вычисляем площадь по формуле Герона
    area := math.Sqrt(p * (p - a) * (p - b) * (p - c))

    // Выводим результаты
    fmt.Printf("Периметр треугольника: %.2f\n", perimeter)
    fmt.Printf("Площадь треугольника: %.2f\n", area)
}

// Функция для вычисления расстояния между двумя точками
func distance(x1, y1, x2, y2 float64) float64 {
    return math.Sqrt(math.Pow(x2-x1, 2) + math.Pow(y2-y1, 2))
}
```

Рисунок – Код для второго индивидуально

```
PS C:\Users\vika1\OneDrive\Desktop\univer\sem6\Прогин\1\ind\2> go run 2.go
Введите координаты вершин треугольника (x1 y1 x2 y2 x3 y3):
1 1 4 5 7 2
Периметр треугольника: 15.33
Площадь треугольника: 10.50
```

Рисунок – Результат программы

Вывод: В ходе выполнения лабораторной работы были изучены основные аспекты языка программирования Go, включая назначение и способы установки Go, типы данных, константы и арифметические операции.

Вопросы:

1) Для объявления переменной типа `int` в Go используется следующий синтаксис:

```
var variableName int
```

2) Переменной типа `int` в Go по умолчанию присваивается нулевое значение, которое равно 0.

3) Для изменения значения существующей переменной в Go используется присваивание нового значения:

```
variableName = newValue
```

4) Множественное объявление переменных в Go позволяет объявить несколько переменных одновременно, указав их имена и типы:

```
var (
    variable1 int
    variable2 string
)
```

5) Для объявления константы в Go используется ключевое слово `const`:

```
const constantName = value
```

6) Нет, значение константы нельзя изменить после ее объявления в Go.

7) В Go поддерживаются арифметические операторы: `+`, `-`, `*`, `/`, `%` (остаток от деления), `++` (увеличение на 1), `--` (уменьшение на 1).

8) Оператор % используется для выполнения операции остатка от деления.

9) Результат выражения $5 / 2$ в Go будет равен 2, так как это целочисленное деление, и в результате будет отброшена дробная часть.

10) Для считывания строки с консоли в Go используется пакет `fmt` и функция `Scanln`:

```
var input string
fmt.Scanln(&input)
```

Для считывания целого числа с консоли в Go также используется пакет `fmt` и функция `Scanln`:

```
var number int
fmt.Scanln(&number)
```

Ошибки при считывании данных с консоли в Go обычно обрабатываются проверкой возвращаемого значения функции `Scanln`.

Для вывода строки в консоль в Go также используется пакет `fmt` и функция `Println`:

```
fmt.Println("Hello, World!")
```

15) Для вывода значения переменной типа `int` в консоль в Go также используется функция `Println`:

```
fmt.Println(variableName)
```

16) Для форматированного вывода числа с плавающей точкой в Go можно использовать функцию `Printf`:

```
fmt.Printf("%.2f\n", floatValue)
```

17) Для объявления переменной типа `byte` и присвоения ей значения 65 в Go можно использовать следующий синтаксис:

```
var b byte = 65
```

Оператор `:=` используется для объявления и инициализации новой переменной в Go, в то время как оператор `=` используется для присваивания значения существующей переменной.

18) Для объявления и использования нескольких переменных в Go можно использовать следующий синтаксис:

```
var variable1 int
```

```
var variable2 string
```