**Московский государственный технический университет им. Н.Э. Баумана**

**Факультет ИУ Кафедра ИУ5**

**Курс «Основы информатики» Отчет по домашнему заданию**

Выполнил студент группы ИУ5-36Б: Турланов В. Е.

Подпись и дата:

Проверил преподаватель каф.: Гапанюк Ю. Е.

Подпись и дата:

Москва, 2024

**Описание задания**

В рамках домашнего задания была поставлена цель изучить новый язык программирования, ранее не изучавшийся, и написать реферат с примерами кода. Для этого был выбран **язык** **программирования Go (или Golang)**, который является современным, высокопроизводительным языком, разработанным для решения задач, связанных с масштабированием приложений и эффективной обработкой данных.

В реферате подробно рассматриваются ключевые особенности языка Go, его основные конструкции и синтаксис, а также приведены примеры кода, разработанные в ходе изучения. Кроме того, была произведена установка компилятора и среды разработки для языка Go. Примеры кода компилировались и тестировались для проверки их работоспособности.

**Язык программирования Go: Обзор**

**Go (или Golang)** — это компилируемый язык программирования, разработанный компанией Google в 2007 году и официально выпущенный в 2009 году. Создателями языка являются Роберт Гризмер, Роб Пайк и Кен Томпсон. Go был разработан для улучшения производительности и упрощения разработки больших программных систем.

**Основные особенности Go:**

1. Простота и лаконичность: Язык имеет минималистичный синтаксис, который легко освоить, особенно если ранее был опыт работы с C-подобными языками.

2. Встроенная поддержка многопоточности: Использование \*goroutines\* и \*channels\* упрощает создание многопоточных программ.

3. Скорость выполнения: Благодаря компиляции в машинный код, программы на Go отличаются высокой производительностью.

4. Автоматическое управление памятью: Go использует сборщик мусора (garbage collector), что избавляет разработчика от необходимости вручную освобождать память.

5. Кроссплатформенность: С помощью инструмента go build можно компилировать приложения под разные платформы.

6. Встроенная система тестирования: Go предоставляет инструмент для написания и выполнения тестов (`go test`).

**Основные конструкции языка Go**

**Переменные и типы данных**

Go является строго типизированным языком. Объявление переменных осуществляется с использованием **ключевого слова var**. Также поддерживается короткая форма объявления с использованием **оператора :=**

Пример:

package main

import "fmt"

func main() {

var a int = 10 // Явное указание типа

b := 20 // Короткая форма (тип выводится автоматически)

var c string = "Hello, Go!"

fmt.Println(a, b, c)

}

#### Условные операторы

Go поддерживает стандартные условные конструкции if-else и switch.

Пример:

package main

import "fmt"

func main() {

x := 15

if x%2 == 0 {

fmt.Println("x - четное число")

} else {

fmt.Println("x - нечетное число")

}

// Switch-конструкция

switch x {

case 10:

fmt.Println("x равен 10")

case 15:

fmt.Println("x равен 15")

default:

fmt.Println("Неизвестное значение")

}

}

#### Циклы

В языке Go имеется только один цикл for, который может использоваться для итерации и как аналог while.

Пример:

package main

import "fmt"

func main() {

for i := 0; i < 5; i++ {

fmt.Println("Итерация:", i)

}

// Использование как while

n := 1

for n < 10 {

fmt.Println(n)

n \*= 2

}

}

#### Функции

Функции в Go объявляются с помощью ключевого слова func. Язык поддерживает возвращение нескольких значений.

Пример:

package main

import "fmt"

// Функция с двумя возвращаемыми значениями

func addAndMultiply(a, b int) (int, int) {

return a + b, a \* b

}

func main() {

sum, product := addAndMultiply(3, 4)

fmt.Println("Сумма:", sum, "Произведение:", product)

}

#### Goroutines и Channels

Одна из ключевых особенностей Go — встроенная поддержка конкурентности через \*goroutines\* и \*channels\*.

Пример:

package main

import (

"fmt"

"time"

)

func printNumbers() {

for i := 1; i <= 5; i++ {

fmt.Println(i)

time.Sleep(100 \* time.Millisecond)

}

}

func main() {

go printNumbers() // Запуск в goroutine

fmt.Println("Горутина запущена!")

time.Sleep(1 \* time.Second) // Ожидание завершения

}

**Примеры программ на Go**

#### Пример 1: Сортировка массива

package main

import (

"fmt"

"sort"

)

func main() {

nums := []int{5, 3, 8, 6, 2}

sort.Ints(nums) // Встроенная сортировка

fmt.Println("Отсортированный массив:", nums)

}

#### Пример 2: Простое веб-приложение

package main

import (

"fmt"

"net/http"

)

func handler(w http.ResponseWriter, r \*http.Request) {

fmt.Fprintf(w, "Привет, это мой первый веб-сервер на Go!")

}

func main() {

http.HandleFunc("/", handler)

fmt.Println("Сервер запущен на порту 8080")

http.ListenAndServe(":8080", nil)

}