

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. Н.Э. Баумана

Факультет “Информатика и системы управления”
Кафедра “Системы обработки информации и управления”



Дисциплина “Парадигмы и конструкции языков программирования”

Отчет по домашнему заданию
«Изучение ранее неизвестного языка
программирования Go на примере
проекта конвертера валют»

Выполнил:
Студент группы ИУ5-34Б
Солиженов У.И.
Преподаватель:
Нардид А.Н.

Москва 2025

Задание

1. Выберите язык программирования (который Вы ранее не изучали) и (1) напишите по нему реферат с примерами кода или (2) реализуйте на нем небольшой проект (с детальным текстовым описанием).
2. Реферат (проект) может быть посвященциальному аспекту (аспектам) языка или содержать решение какой-либо задачи на этом языке.
3. Необходимо установить на свой компьютер компилятор (интерпретатор, транспилятор) этого языка и произвольную среду разработки.
4. В случае написания реферата необходимо разработать и откомпилировать примеры кода (или модифицировать стандартные примеры).
5. В случае создания проекта необходимо детально комментировать код.
6. При написании реферата (создании проекта) необходимо изучить и корректно использовать особенности парадигмы языка и основных конструкций данного языка.
7. Приветствуется написание черновика статьи по результатам выполнения ДЗ. Черновик статьи может быть подготовлен группой студентов, которые исследовали один и тот же аспект в нескольких языках или решили одинаковую задачу на нескольких языках.

Текстовое описание

Проект: Консольный конвертер валют

Аннотация и технологический стек. Проект представляет собой прикладное консольное приложение, предназначенное для пересчета различных иностранных валют (USD, EUR, CNY, UZS) в российские рубли.

Технологическая база проекта включает версию языка Go 1.22.2 и среду разработки VS Code в связке с подсистемой WSL.

Архитектура и принципы работы. Приложение спроектировано с использованием стандартных библиотек Go. В качестве основного механизма хранения данных используется хеш-таблица (map), где ключом выступает код валюты (string), а значением — структура данных Currency.

Использование карт обеспечивает константное время доступа к курсам валют $O(1)$, что делает приложение производительным.

Алгоритмическая реализация. Программа запрашивает у пользователя ввод кода валюты, приводит его к верхнему регистру с помощью пакета strings и

производит поиск в оперативной памяти²⁴²⁴²⁴²⁴. После успешной валидации суммы выполняется расчет по формуле:

$\text{Result} = \text{Amount} \times \text{Rate}$

Результат выводится в форматированном виде с точностью до двух знаков после запятой.

Заключение. Разработанный продукт (MVP) демонстрирует ключевые возможности Go: работу со структурами, статическую типизацию и эффективную обработку пользовательского ввода без использования внешних зависимостей.

Структура программы

- **go.mod** — файл инициализации модуля, указывающий версию языка 1.22.2.
- **main.go** — основной файл исходного кода программы.

Листинг программы

main.go

```
package main
```

```
import (  
    "fmt"      // Пакет для ввода и вывода данных  
    "os"       // Пакет для работы с операционной системой (выход из программы)  
    "strings"  // Пакет для обработки строк (перевод в верхний регистр)  
)
```

```
// Currency — это структура (аналог класса), описывающая валюту.
```

```
// В Go мы используем структуры для группировки данных.
```

```
type Currency struct {
```

```
    Name string // Полное название валюты
```

```
    Rate float64 // Курс обмена: сколько рублей стоит 1 единица этой валюты
```

```
}
```

```
// Convert — это метод структуры Currency.
```

```
// Он принимает количество валюты и возвращает эквивалент в рублях.
```

```
func (c Currency) Convert(amount float64) float64 {
```

```
    return amount * c.Rate
```

```
}
```

```
func main() {
```

```
    // Инициализируем карту (map), где ключ — это код валюты (строка),
```

```
    // а значение — структура Currency с данными.
```

```

exchangeRates := map[string]Currency{
    "USD": {Name: "Доллар США", Rate: 79.01},
    "EUR": {Name: "Евро", Rate: 99.20},
    "CNY": {Name: "Китайский юань", Rate: 12.65},
    "UZS": {Name: "Узбекский сум", Rate: 0.0072}, // Обновленная валюта по
вашей просьбе
}

fmt.Println("====")
fmt.Println(" ДОМАШНЕЕ ЗАДАНИЕ: КОНВЕРТЕР ВАЛЮТ (GO)")
fmt.Println("====")

// Выводим список доступных валют из карты
fmt.Println("Доступные валюты для конвертации в рубли:")
for code, info := range exchangeRates {
    fmt.Printf("[%s] - %s (Курс: %.4f руб.)\n", code, info.Name, info.Rate)
}
fmt.Println("-----")

var code string
var amount float64

// Запрашиваем код валюты у пользователя
fmt.Print("Введите код валюты (например, UZS): ")
fmt.Scan(&code)
code = strings.ToUpper(code) // Превращаем в заглавные буквы для
соответствия ключам в map

// Проверяем, существует ли такая валюта в нашей базе данных
// В Go обращение к карте может возвращать второй аргумент (ok),
указывающий на наличие ключа
currency, exists := exchangeRates[code]
if !exists {
    fmt.Printf("Ошибка: Валюта с кодом '%s' не найдена.\n", code)
    os.Exit(1) // Завершаем программу, если код неверный
}

// Запрашиваем сумму для перевода
fmt.Printf("Введите сумму в %s: ", code)
_, err := fmt.Scan(&amount)

// Простая обработка ошибок ввода (например, если ввели буквы вместо
цифр)
if err != nil || amount < 0 {
    fmt.Println("Ошибка: Введите корректное положительное число.")
    return
}

// Вызываем метод структуры для выполнения расчета

```

```
rubResult := currency.Convert(amount)

// Вывод итогового результата с форматированием до 2 знаков после запятой
fmt.Println("-----")
fmt.Printf("РЕЗУЛЬТАТ: %.2f %s = %.2f Российских рублей\n", amount, code,
rubResult)
fmt.Println("=====")
}
```

Запуск программы в терминале

```
● umid@WIN-2QQU3U62OKST:~/course_pcpl/homework$ go run main.go
=====
ДОМАШНЕЕ ЗАДАНИЕ: КОНВЕРТЕР ВАЛЮТ (GO)
=====

Доступные валюты для конвертации в рубли:
[USD] - Доллар США (Курс: 79.0100 руб.)
[EUR] - Евро (Курс: 99.2000 руб.)
[CNY] - Китайский юань (Курс: 12.6500 руб.)
[UZS] - Узбекский сум (Курс: 0.0072 руб.)

-----
Введите код валюты (например, UZS): USD
Введите сумму в USD: 100
-----
РЕЗУЛЬТАТ: 100.00 USD = 7901.00 Российских рублей
=====
```

