

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

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

Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №6
«Шаблоны проектирования и модульное тестирование на Go»

Выполнил:
Студент группы ИУ5-31Б
Попов А.С.
Подпись и дата:

Проверил:
Преподаватель каф. ИУ5
Гапанюк Ю.Е.
Подпись и дата:

Задание

1. Необходимо для произвольной предметной области реализовать от одного до трех шаблонов проектирования: один порождающий, один структурный и один поведенческий. В качестве справочника шаблонов можно использовать каталог из википедии. Для сдачи лабораторной работы в минимальном варианте достаточно реализовать один паттерн.
2. В модульных тестах необходимо применить следующие технологии:
 - TDD - фреймворк.
 - BDD - фреймворк.
 - Создание Mock-объектов.

Текст программы

order_domain.go

```
package builder

import (
    "fmt"
    "time"
    "github.com/google/uuid"
)

type Currency string

const (
    // Fiat currencies
    EUR Currency = "EUR"
    USD Currency = "USD"
    RUB Currency = "RUB"
    // Crypto-currencies
    TRX Currency = "TRX"
    BTC Currency = "BTC"
    ETH Currency = "ETH"
)
type Order struct {
    ID uuid.UUID

    Items []Item

    Additional map[string]any

    CreatedAt     time.Time
    DeliveredAt  *time.Time
}

func (o Order) String() string {
    var items string
    for _, item := range o.Items {
        items += fmt.Sprintf("-%s", item)
    }

    return fmt.Sprintf(
        "ID: %s\nItems: [%n%s]\nAdditional: %v\nCreatedAt: %v\nDeliveredAt: %v\n",
        o.ID,
        items,
```

```

        o.Additional,
        o.CreatedAt,
        o.DeliveredAt,
    )
}

type Item struct {
    Name         string
    Description string
    Price        float64
    Currency     Currency
}

func (i Item) String() string {
    return fmt.Sprintf(" Name: %s\n Description: %s\n Price: %0.2f\n Currency: %s\n", i.Name, i.Description, i.Price, i.Currency)
}

```

order_builder.go

```

package builder

import (
    "time"
    "github.com/google/uuid"
)

type OrderBuilder struct {
    items []Item
}

func New() *OrderBuilder {
    return &OrderBuilder{}
}

// `Builder` implementation
func (o OrderBuilder) Build() *Order {
    return &Order{
        ID:          uuid.New(),
        Items:       o.items,
        CreatedAt:  time.Now(),
        Additional: map[string]any{
            "source": "builder",
        },
    }
}

```

```
func (o OrderBuilder) WithItems(items ...Item) OrderBuilder {
    o.items = append(o.items, items...)
    return o
}
```

order_builder_test.go

```
package builder_test

import (
    "patterns/internal/builder"
    "testing"
)

func TestOrderBuilder(t *testing.T) {
    order := builder.New().
        WithItems(
            builder.Item{
                Name:         "Cap",
                Description: "Cap for men",
                Price:        10,
                Currency:    builder.USD,
            },
            builder.Item{
                Name:         "Talon Knife",
                Description: "CS2 | Talon Knife | Boreal Forest",
                Price:        1500,
                Currency:    builder.TRX,
            },
        ).
        Build()

    t.Logf("Formed order:\n%s", order)
}
```

Скриншоты работы программы

```
PS D:\proga\lab_go_test\internal\builder> & "C:\Program Files\Go\bin\go.exe" test -v
● === RUN   TestOrderBuilder
order_builder_test.go:26: Formed order:
    ID: 0bdc7d79-c1f8-4b8e-adce-8467e27bc251
    Items: [
        - Name: Cap
          Description: Cap for men
          Price: 10.00
          Currency: USD
        - Name: Talon Knife
          Description: CS2 | Talon Knife | Boreal Forest
          Price: 1500.00
          Currency: TRX
    ]
    Additional: map[source:builder]
    CreatedAt: 2025-12-27 00:50:17.873624 +0300 MSK m=+0.002816101
    DeliveredAt: <nil>
--- PASS: TestOrderBuilder (0.03s)
PASS
ok      patterns/internal/builder      0.416s
○ PS D:\proga\lab_go_test\internal\builder>
```

Рис. 1 – Работа приложения