

# Лабораторна робота №5

## ООП. Структури та інтерфейси

### Мета роботи:

Засвоїти принципи проектування та оголошення структур та інтерфейсів; вивчити особливості реалізації структур та їх методів.

### Хід роботи:

#### 1. Оголошення структур та конструкторів

```
package main

import "fmt"

// Оголошення структури Product
type Product struct {
    Name      string
    Price     float64
    Cost      string
    Quantity  int
    Producer  string
    Weight    float64
}

// Конструктор для Product
func NewProduct(name string, price float64, cost string, quantity int,
    producer string, weight float64) Product {
    return Product{Name: name, Price: price, Cost: cost, Quantity:
    quantity, Producer: producer, Weight: weight}
}

// Оголошення структури Currency
type Currency struct {
    Name      string
    ExRate    float64
}

// Конструктор для Currency
func NewCurrency(name string, exRate float64) Currency {
    return Currency{Name: name, ExRate: exRate}
}
```

					ДУ «Житомирська політехніка»..24.121.19.000 – Лр5			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Стецюк Б.Ю.			Звіт з лабораторної роботи №6		Літ.	Арк.
Перевір.		Петросян Р.В.						1
Керівник							ФІКТ Гр. ІПЗ-22-2	
Н. контр.								
Зав. каф								

## 2. Методи для Product

```
func (p Product) GetPriceIn(grnRate float64) float64 {  
    return p.Price * grnRate  
}
```

```
func (p Product) GetTotalPrice() float64 {  
    return p.Price * float64(p.Quantity)  
}
```

```
func (p Product) GetTotalWeight() float64 {  
    return p.Weight * float64(p.Quantity)  
}
```

## 3. Функції для роботи з масивами

```
func ReadProductsArray() []Product {  
    var n int  
    fmt.Print("Enter number of products: ")  
    fmt.Scan(&n)  
  
    products := make([]Product, n)  
    for i := 0; i < n; i++ {  
        fmt.Printf("Enter details for product %d\n", i+1)  
        var name, cost, producer string  
        var price, weight float64  
        var quantity int  
  
        fmt.Print("Name: ")  
        fmt.Scan(&name)  
        fmt.Print("Price: ")  
        fmt.Scan(&price)  
        fmt.Print("Cost: ")  
        fmt.Scan(&cost)  
        fmt.Print("Quantity: ")  
        fmt.Scan(&quantity)  
        fmt.Print("Producer: ")  
        fmt.Scan(&producer)  
        fmt.Print("Weight: ")  
        fmt.Scan(&weight)  
  
        products[i] = NewProduct(name, price, cost, quantity, producer,  
weight)  
    }  
  
    return products  
}  
  
func PrintProduct(p Product) {
```

		Стецюк Б.Ю.			ДУ «Житомирська політехніка»..24.121.19.000 – – .	Арк.
		Петросян Р.В				
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    fmt.Printf("Name: %s, Price: %.2f, Cost: %s, Quantity: %d, Producer:
%s, Weight: %.2f\n",
        p.Name, p.Price, p.Cost, p.Quantity, p.Producer, p.Weight)
}

func PrintProducts(products []Product) {
    for _, p := range products {
        PrintProduct(p)
    }
}

func GetProductsInfo(products []Product) (Product, Product) {
    if len(products) == 0 {
        panic("No products available")
    }

    min, max := products[0], products[0]
    for _, p := range products {
        if p.Price < min.Price {
            min = p
        }
        if p.Price > max.Price {
            max = p
        }
    }
    return min, max
}

```

#### 4. Тестування програми

```

func main() {
    products := ReadProductsArray()
    PrintProducts(products)

    min, max := GetProductsInfo(products)
    fmt.Println("Cheapest Product:")
    PrintProduct(min)
    fmt.Println("Most Expensive Product:")
    PrintProduct(max)
}

```

#### Висновки:

Під час лабораторної роботи було вивчено концепції ООП у мові Go, зокрема реалізацію структур, методів та інтерфейсів. Отримано практичні навички роботи з користувацькими типами даних, написання конструкторів, геттерів, сеттерів та методів для обробки масивів структур.

		Стецюк Б.Ю.			ДУ «Житомирська політехніка»..24.121.19.000 –	Арк.
		Петросян Р.В				
Змн.	Арк.	№ докум.	Підпис	Дата		