

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:

Варіант : непарний

Завдання 1: Оголосити структуру Product, яка представляє інформацію про один товар, який зберігається на складі. Має наступні поля:

- Name – назва товару;
- Price - вартість одиниці товару;
- Cost – грошова одиниця, у якій вимірюється вартість;
- Quantity – кількість наявних товарів на складі;
- Producer – назва компанії-виробника;
- Weight – вага одиниці товару.

Результат роботи:

```
type Product struct {
    name      string
    price     float64
    cost      Currency
    quantity  int64
    producer  string
    weight    float64
}
```

Оголосив структуру Currency, яка містить наступні поля:

- Name – назва валюти;
- ExRate – курс (дробове число - кількість гривень та копійок, що дають за одну одиницю валюти).

Результат роботи:

```
type Currency struct {
    name      string
    exRate    float64
}
```

Завдання 2: Для кожної структури реалізував конструктори:

					ДУ «Житомирська політехніка».22.121.4.000 – Лр5			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Васильєв А.К.			Звіт з лабораторної роботи 5		Лім.	Арк.
Перевір.		Петросян Р.В.						1
Керівник								6
Н. контр.							ФІКТ Гр. ІПЗк-22-1	
Зав. каф.								

```
func ProductConstructor(name string, price float64, cost Currency,
    quantity int64, producer string, weight float64) *Product {
    p := new(Product)

    p.name = name
    p.price = price
    p.cost = cost
    p.quantity = quantity
    p.producer = producer
    p.weight = weight

    return p
}
```

```
func ProductConstructorWithoutCost(name string, price float64,
    quantity int64, producer string, weight float64) *Product {
    p := new(Product)

    p.name = name
    p.price = price
    p.cost = Currency{ name: "USD", exRate: 36.8}
    p.quantity = quantity
    p.producer = producer
    p.weight = weight

    return p
}
```

```
func ProductDefaultConstructor() *Product {
    p := new(Product)

    p.name = "Tab M10 (3rd Gen) 4/64 Wi-Fi Storm Grey"
    p.price = 230.7
    p.cost = Currency{ name: "USD", exRate: 36.8}
    p.quantity = 54
    p.producer = "Lenovo"
    p.weight = 423.7

    return p
}
```

Завдання 3: Реалізував set- та get- методи для кожного поля Product.

Завдання 4: У структурі Product створив методи:

- *GetPriceIn()*, який повертає ціну одиниці товару в гривнях;
- *GetTotalPrice()*, що повертає загальну вартість усіх наявних на складі товарів даного виду;
- *GetTotalWeight()*, який повертає загальну вагу усіх товарів на складі даного виду.

У структурі *Currency* методи обрати самостійно.

					ДУ «Житомирська політехніка».22.121.4.000 – Лр5	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		2

Завдання 5: Реалізував ф-ї:

- *ReadProductsArray()* – читає з клавіатури дані і повертає множину об'єктів типу *Product* (*n* об'єктів з різною валютою);
- *PrintProduct()* – приймає тип *Product* і виводить його на екран;
- *PrintProducts()* – приймає множину типу *Product* і виводить його на екран;
- *GetProductsInfo()* – приймає множину типу *Product* і повертає через вихідні параметри найдешевший та найдорожчий товар.

Результати роботи програми :

```
Зчитування продуктів...
Введіть кількість елементів у масиві продуктів: 3

Заповнення інформації про продукт # 1
Оберіть тип конструктора структури продукту:
1) З усіма полями
2) Валюта за замочуванням
3) За замовчуванням
Введіть обраний варіант: 1
Введіть назву продукту: Honey
Введіть ціну продукту: 3
Оберіть тип конструктора структури валюти:
1) З назвою та курсом
2) Лише курсом
3) За замовчуванням
Введіть обраний варіант: 1
Введіть назву валюти: USD
Введіть курс: 39.3
Введіть кількість продукту: 10
Введіть назву виробника продукту: Gelly
Введіть вагу продукту: 1
```

					ДУ «Житомирська політехніка».22.121.4.000 – Лр5	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

Заповнення інформації про продукт # 2
 Оберіть тип конструктора структури продукту:
 1) З усіма полями
 2) Валюта за замочуванням
 3) За замовчуванням
 Введіть обраний варіант: 2
 Введіть назву продукту: Grimgo
 Введіть ціну продукту: 17
 Введіть кількість продукту: 5
 Введіть назву виробника продукту: Gagergag
 Введіть вагу продукту: 3

Заповнення інформації про продукт # 3
 Оберіть тип конструктора структури продукту:
 1) З усіма полями
 2) Валюта за замочуванням
 3) За замовчуванням
 Введіть обраний варіант: 3

Продукт # 1
 Name: Honey
 Price: 3.00
 Cost: 1 USD = 39.3 UAH
 Quantity: 10
 Producer: Gelly
 Weight: 1.00

Продукт # 2
 Name: Grimgo
 Price: 17.00
 Cost: 1 USD = 36.8 UAH
 Quantity: 5
 Producer: Gagergag
 Weight: 3.00

					ДУ «Житомирська політехніка».22.121.4.000 – Лр5	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

Продукт # 3

Name: Tab M10 (3rd Gen) 4/64 Wi-Fi Storm Grey

Price: 230.70

Cost: 1 USD = 36.8 UAH

Quantity: 54

Producer: Lenovo

Weight: 423.70

Визначення мінімального і максимального продуктів...

Найдешевший продукт: Honey 117.90 UAH

Найдорожчий продукт: Tab M10 (3rd Gen) 4/64 Wi-Fi Storm Grey 8489.76 UAH

Контрольні питання

1. Що таке структура і для чого призначена?

Структури – набір різних типів даних, що зберігаються як єдине ціле та передбачають доступ до окремих полів структури. Для доступу до полів структури використовується точкова нотація. Структури доцільно використовувати там, де необхідно об'єднати дані, що відносяться до одного об'єкту

Загальний синтаксис визначення шаблону структури має вигляд:

```
var ім'я struct {  
    ім'я_змінної1 тип1  
    ім'я_змінної2 тип2  
    // інші члени структури  
}
```

2. Як створюються користувацький тип?

Роль класів виконують користувацькі типи (в основному структури).

Структури можуть мати методи. Метод – це особливого роду функція, яка викликається відносно значення користувацького типу (передається методу, що викликається). Значення передається за вказівником або за значенням, в залежності від того, якого оголошений метод. Синтаксис методу аналогічний функції, тільки перед ім'ям методу вказується користувацький тип до якого він відноситься.

					ДУ «Житомирська політехніка».22.121.4.000 – Лр5	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

Наприклад:

```
type Count struct { c uint64 }
```

```
func (this *Count) Inc() { this.c++ }
```

```
func (this *Count) Dec() { this.c-- }
```

```
func (this *Count) Zero() { this.c = 0 }
```

```
func (this Count) IsZero() bool { return this.c == 0 }
```

3. Що таке ООП?

Об'єктоорієнтоване програмування — одна з парадигм програмування, яка розглядає програму як множину «об'єктів», що взаємодіють між собою.

Основу ООП складають чотири основні концепції: інкапсуляція, успадкування, поліморфізм та абстракція. Однією з переваг ООП є краща модульність програмного забезпечення.

4. Для чого використовується вбудова користувацьких типів?

Складові типи на основі структур можуть включати одна або більше типів у вигляді вбудованих полів. Головна зручність такого підходу полягає в можливості викликати методи вбудованого типу щодо значення користувацького типу, так якби вони були власними методами цього типу. Методи вбудованих полів можна перевизначати, просто створюючи, для структури в яку вбудовують, нові методи з тими ж іменами, що і методи вбудованого поля.

5. Що таке інтерфейси і для чого вони призначені?

Інтерфейс – це тип, який визначає сигнатури одного або більше методів. Інтерфейси є повністю абстрактними, тому немає ніякої можливості створювати їх екземпляри. Однак є можливість створювати змінні з типами інтерфейсів, яким потім можна присвоювати значення будь-якого конкретного типу, що володіє методами інтерфейсу.

Висновок: засвоїв принципи проектування та оголошення структура інтерфейсів; вивчив особливості реалізації структур та їх методів.

					ДУ «Житомирська політехніка».22.121.4.000 – Лр5	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		