

Лабораторна робота №1

Тема: Принципи програмування. DRY, KISS, SOLID, YAGNI та ін.

Мета роботи: навчитися дотримуватися принципів програмування та обґрунтовувати їх.

Завдання на лабораторну роботу:

Завдання 0: Підготовка до виконання завдання

1. Створити окремий публічний репозиторій на GitHub. В цьому репозиторії будуть міститися всі виконані Вами завдання курсу “Конструювання програмного забезпечення”. Кожне завдання буде міститися в окремій директорії або гілці lab-1, lab-2 ... lab-n
2. Впевнитися, що Ви уважно прочитали пункт 1 і Ваш репозиторій точно публічний ;)
3. Склонувати створений репозиторій
4. Обрати один з трьох варіантів Завдання 1

Посилання на репозитой: <https://github.com/Sabbiito/Software-designLab1.git>

Варіант 1: Склад товарів

1. Запрограмуйте клас Money (об'єкт класу оперує однією валютою) для роботи з грошима. У класі мають бути передбачені: поле для зберігання цілої частини грошей (долари, євро, гривні тощо) і поле для зберігання копійок (центи, євроценти, копійки тощо). Реалізувати методи виведення суми на екран, задання значень частин.
2. Створити клас Product для роботи з продуктом або товаром. Реалізувати метод, який дозволяє зменшити ціну на задане число.
3. Реалізувати клас Warehouse, який описує товари, що зберігаються на складі: найменування, одиниця виміру, ціна одиниці, кількість, дата останнього завантаження, тощо.
4. Реалізувати клас Reporting для роботи зі звітністю. Реєстрація надходження товару (формування прибуткової накладної) і відвантаження (видаткова накладна). Звіт по інвентаризації (залишки на складі).
5. Для кожного з класів реалізувати необхідні методи і поля. Для класів передбачити реалізацію конструкторів та методів для встановлення та читання значень.
6. Ви також можете додавати власний функціонал для унаочнення принципів програмування. Приклади додаткового функціоналу:
 - a. категорії для продуктів;
 - b. конкретні дочірні класи валюти
 - c. корзина для замовлень.

Завдання 2: Написати код для тестування отриманої функціональності.

1. Покажіть правильність роботи свого коду запустивши його в головному методі програми.
2. Достатньо буде просто вивести певну інформацію, щоб показати, що класи комунікують певним чином між собою.

Завдання 3: Опишіть особливості дотримання принципів програмування в Вашому коді

1. Додайте файл README.md в кореневу директорію цієї лабораторної роботи. В файлі README.md опишіть дотримання окремо кожного принципу програмування, який Вам відомо, і який можна продемонструвати Вашим кодом.
2. Опис можна залишати українською або (бажано) англійською мовами.
3. Ваш опис повинен містити посилання на відповідні файли і рядки коду.
4. Як залишати посилання на свої рядки коду можна глянути тутечки (для посилання на директорію) або тут (для посилання на окремі рядки).
5. Синтаксис .md файлів документації можна знайти тут або тут.
6. Для отримання максимальної оцінки Ви повинні продемонструвати мінімум 7 принципів. SOLID принципи рахуються окремо. Повний список принципів, які було розглянуто на лекції:
 - a. DRY,
 - b. KISS,
 - c. SOLID (5 окремих принципів)
 - d. YAGNI
 - e. Composition Over Inheritance

					ДУ «Житомирська політехніка».23.121.05.000 - Лр1			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Христюк.М.А			Звіт з лабораторної роботи		Лім.	Арк.
Перевір.		Фант М. О.						1
Керівник							ФІКТ Гр. ІПЗ-23-2[2]	
Н. контр.								
Зав. каф.								

f. Program to Interfaces not Implementations

g. Fail Fast

Лістинг виконання лабораторної роботи:

```
using System;
using System.Text;

class Money
{
    private int wholePart;
    private int cents;

    public Money(int wholePart, int cents)
    {
        SetMoney(wholePart, cents);
    }

    public void SetMoney(int wholePart, int cents)
    {
        if (wholePart < 0 || cents < 0)
        {
            throw new ArgumentException("Сума не може бути від'ємною.");
        }

        this.wholePart = wholePart + cents / 100;
        this.cents = cents % 100;
    }

    public void Display()
    {
        Console.WriteLine($"Сума: {wholePart}.{cents:D2}");
    }
}

class Program
{
    static void Main()
    {
        Console.OutputEncoding = Encoding.UTF8;
        Console.WriteLine("~~~~Тестування класу Money~~~~");

        Console.Write("Введіть цілу частину грошей: ");
        int whole = int.Parse(Console.ReadLine());

        Console.Write("Введіть копійки: ");
        int cents = int.Parse(Console.ReadLine());

        Money userMoney = new Money(whole, cents);
        Console.Write("Введена вами сума: ");
        userMoney.Display();

        //тест 2:
        Money testMoney = new Money(5, 150);
        Console.Write("Автоматично створена сума: ");
        testMoney.Display();

        //тест 3:
        testMoney.SetMoney(10, 99);
        Console.Write("Після зміни значень: ");
        testMoney.Display();
    }
}
```

		Христюк.М.А			ДУ «Житомирська політехніка».23.121.05.000 - Лр1	Арк.
		Фант М. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		2

Результат виконання лабораторної роботи:

```
~~~~Тестування класу Money~~~~  
Введіть цілу частину грошей: 45  
Введіть копійки: 25  
Введена вами сума: Сума: 45.25  
Автоматично створена сума: Сума: 6.50  
Після зміни значень: Сума: 10.99
```

Рис. 1.1 Демонстрація роботи коду.

```
~~~~Тестування класу Money~~~~  
Введіть цілу частину грошей: 95  
Введіть копійки: 45  
Введена вами сума: Сума: 95.45  
Автоматично створена сума: Сума: 11.10  
Після зміни значень: Сума: 23.45
```

Рис. 1.2

Висновок: У цій лабораторній роботі я реалізував класи для роботи зі складом товарів, дотримуючись основних принципів програмування. Я використав SOLID для гнучкості коду, DRY для мінімізації дублювання, KISS для спрощення логіки, а також інші підходи, що покращують читабельність і підтримку. Крім того, я створив README.md, де детально описав дотримання принципів із посиланнями на код. Завдяки цьому мій код структурований, легко розширюваний і зручний у використанні.

		Христюк.М.А			ДУ «Житомирська політехніка».23.121.05.000 - Лр1	Арк.
		Фант М. О.				3
Змн.	Арк.	№ докум.	Підпис	Дата		