**Лабораторна робота №1**

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

**Мета роботи:** навчитися дотримуватися принципів програмування та обґрунтовувати їх.

**Хід Роботи**

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.

public class Money

{

private int dollars;

private int cents;

public Money(int dollars, int cents)

{

this.dollars = dollars;

this.cents = cents;

}

public void Print()

{

Console.WriteLine($"Грошi: {dollars} долар{i(dollars)} {cents} цент{i(cents)}");

}

private string i(int num)

{

return num != 1 ? "и" : "";

}

}

2. Створити клас Product для роботи з продуктом або товаром. Реалізувати метод, який дозволяє зменшити ціну на задане число.

public class Product

{

private double price;

public double Price

{

get { return price; }

set { price = value; }

}

public Product(double price)

{

this.price = price;

}

public void DecreasePrice(double amount)

{

price -= amount;

}

}

3. Реалізувати клас Warehouse, який описує товари, що зберігаються на складі: найменування, одиниця виміру, ціна одиниці, кількість, дата останнього завозу, тощо.

public class Warehouse

{

private string name;

private string unit;

private double unitPrice;

private int quantity;

private DateTime lastArrivalDate;

public int Quantity

{

get { return quantity; }

set { quantity = value; }

}

public Warehouse(string name, string unit, double unitPrice, int quantity, DateTime lastArrivalDate)

{

this.name = name;

this.unit = unit;

this.unitPrice = unitPrice;

this.quantity = quantity;

this.lastArrivalDate = lastArrivalDate;

}

public void PrintInfo()

{

Console.WriteLine($"Продукт: {name}, Одиниця вимiру: {unit}, Цiна за одиницю: {unitPrice}, Кiлькiсть: {quantity}, Останнiй прихiд: {lastArrivalDate}");

}

}

4. Реалізувати клас Reporting для роботи зі звітністю. Реєстрація надходження товару (формування прибуткової накладної) і відвантаження (видаткова накладна). Звіт по інвентаризації (залишки на складі).

public class Reporting

{

private List<string> reports;

private List<string> arrivalReports;

private List<string> shipmentReports;

public Reporting()

{

reports = new List<string>();

arrivalReports = new List<string>();

shipmentReports = new List<string>();

}

public void AddReport(string report)

{

reports.Add(report);

}

public void AddArrivalReport(string report)

{

arrivalReports.Add(report);

}

public void AddShipmentReport(string report)

{

shipmentReports.Add(report);

}

public void PrintReports()

{

foreach (var report in reports)

{

Console.WriteLine(report);

}

}

public void PrintArrivalReports()

{

foreach (var report in arrivalReports)

{

Console.WriteLine(report);

}

}

public void PrintShipmentReports()

{

foreach (var report in shipmentReports)

{

Console.WriteLine(report);

}

}

public void InventoryReport(Warehouse warehouse)

{

warehouse.PrintInfo();

string report = $"Звiт про iнвентаризацiю: Залишена кiлькiсть: {warehouse.Quantity}";

reports.Add(report);

Console.WriteLine(report);

}

public void RegisterArrival(Warehouse warehouse, string productName, int quantity, DateTime arrivalDate)

{

warehouse.PrintInfo();

string report = $"Продукт {productName} прибув. Кiлькiсть: {quantity}, Дата: {arrivalDate}";

reports.Add(report);

arrivalReports.Add(report);

Console.WriteLine(report);

}

public void ShipGoods(Warehouse warehouse, string productName, int quantity)

{

warehouse.PrintInfo();

string report = $"Продукт {productName} вiдвантажено. Кiлькiсть: {quantity}";

reports.Add(report);

shipmentReports.Add(report);

Console.WriteLine(report);

}

}

5. Для кожного з класів реалізувати необхідні методи і поля. Для класів передбачити реалізацію конструкторів та методів для встановлення та читання значень.

Реалізовано.

**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.















