Кам’янець-Подільський національний університет

 імені Івана Огієнка

КАФЕДРА КОМП’ЮТЕРНИХ НАУК

Навчальна дисципліна «Об’єктно-орієнтоване програмування»

ЛАБОРАТОРНА РОБОТА #0105

Тема: Інтерфейси

Варіант №3

Виконав:

студент 1-го курсу

групи KNms1-B21

Волошин В.Ю.

Прийняла:

старший викладач,

Мястковська М.О.

Кам’янець-Подільський – 2023

1. **Хід роботи.**

Інтерфейси в мові програмування C# є ключовою концепцією об'єктно-орієнтованого програмування, яка дозволяє структурувати код та визначати контракти для класів. Інтерфейс визначає набір методів, властивостей та подій, але сам не містить конкретної реалізації. Клас може реалізовувати один чи декілька інтерфейсів, забезпечуючи конкретні реалізації визначених у них елементів.

Основна перевага використання інтерфейсів полягає в гнучкості та можливості використання множинного спадкування. Клас, який реалізує інтерфейс, отримує здатність взаємодіяти з іншими класами, які також реалізують цей інтерфейс, не залежно від їхньої конкретної реалізації. Це сприяє створенню загальних стандартів для взаємодії між класами.

Інтерфейси також дозволяють використовувати поліморфізм, забезпечуючи можливість використання об'єктів, які реалізують один і той же інтерфейс, в одному контексті, що робить код більш гнучким та легко розширюваним.

Важливою рисою інтерфейсів є відсутність реалізації в самому інтерфейсі, що забезпечує чистоту коду та визначає лише контракти, які повинні бути виконані класами-реалізаторами.

Отже, використання інтерфейсів у C# є важливим елементом об'єктно-орієнтованого програмування, який сприяє створенню гнучких, структурованих та розширюваних програм.

1. **Завдання до виконання.**

### Завдання 1.

Описати інтерфейс **IShape** з членами:

1) метод, що виводить на екран тип фігури;

2) метод, що виводить на екран площу фігури;

3) властивість, що відповідає за перший лінійний розмір фігури.

Описати інтерфейс IColoredShape, що успадковується від інтерфейсу IShape та містить властивість, що відповідає за колір фігури, та метод, що виводить на екран колір фігури.

Описати клас Circle, та ColoredCircle що реалізовують інтерфейси IShape та IColoredShape відповідно.

Створити не менше трьох екземплярів кожного класу, сформувати з них один масив.

Створити метод, який для кожного елемента масиву виконує його методи в залежності від інтерфейсу, що реалізується класом об’єкта.

### Завдання 2.

Визначити інтерфейс **Printable**, який містить метод void print().

Визначити клас Book, який реалізує інтерфейс Printable.

Визначити клас Magazine, який реалізує інтерфейс Printable.

Створити масив типу Printable, який міститиме книги та журнали.

У циклі пройти масивом і викликати метод print() кожного об'єкта.

Створити статичний метод printMagazines(Printable[] printable) у класі Magazine, який виводить на консоль назви лише журналів.

Створити статичний метод printBooks(Printable[] printable) у класі Book, який виводить на консоль назви лише книг. Використовуємо оператор instanceof.

### Завдання 3.

Створити клас **Shop**, що містить масив із розмірами одягу (XXS, XS, S, M, L). Клас містить метод getDescription, який повертає рядок "Дорослий розмір".

Перевизначити метод getDescription - для константи XXS метод має повертати рядок "Дитячий розмір".

Також клас має містити числове значення euroSize (32, 34, 36, 38, 40), що відповідає кожному розміру. Створити конструктор, який приймає на вхід euroSize.

Створити інтерфейси "Чоловічий Одяг" з методом "одягти Чоловіка" та "Жіночий Одяг" з методом "одягти Жінку".

Створити абстрактний клас Одяг, що містить змінні – розмір одягу, вартість, колір.

Створити класи спадкоємці Одягу - Футболка (реалізує інтерфейси "Чоловічий Одяг" та "Жіночий Одяг"), Штани (реалізує інтерфейси "Чоловічий Одяг" та "Жіночий Одяг"), Спідниця (реалізує інтерфейси "Жіночий Одяг"), Краватка (реалізує інтерфейси " Чоловічий одяг").

Створити масив, що містить усі типи одягу.

Створити клас Ательє, що містить методи одягнути Жінку, одягнути Чоловіка, на вхід яких надходитиме масив, що містить всі типи одягу.

Метод одягу Жінку виводить на консоль всю інформацію про жіночий одяг. Те ж саме для методу одягнути Чоловіка.

1. **Лістинг програми.**

// Інтерфейс IShape

public interface IShape

{

    void DisplayType();

    void DisplayArea();

    double FirstDimension { get; set; }

}

// Інтерфейс IColoredShape успадковується від IShape

public interface IColoredShape : IShape

{

    string Color { get; set; }

    void DisplayColor();

}

// Клас Circle реалізує інтерфейс IShape

public class Circle : IShape

{

    public double FirstDimension { get; set; }

    public void DisplayType()

    {

        Console.WriteLine("Figure Type: Circle");

    }

    public void DisplayArea()

    {

        double area = Math.PI \* Math.Pow(FirstDimension, 2);

        Console.WriteLine($"Area: {area}");

    }

}

// Клас ColoredCircle реалізує інтерфейс IColoredShape

public class ColoredCircle : IColoredShape

{

    public double FirstDimension { get; set; }

    public string Color { get; set; }

    public void DisplayType()

    {

        Console.WriteLine("Figure Type: Colored Circle");

    }

    public void DisplayArea()

    {

        double area = Math.PI \* Math.Pow(FirstDimension, 2);

        Console.WriteLine($"Area: {area}");

    }

    public void DisplayColor()

    {

        Console.WriteLine($"Color: {Color}");

    }

}

// Інтерфейс Printable

interface Printable

{

    void Print();

}

// Клас Book реалізує інтерфейс Printable

class Book : Printable

{

    private string title;

    public Book(string title)

    {

        this.title = title;

    }

    public void Print()

    {

        Console.WriteLine("Book: " + title);

    }

}

// Клас Magazine реалізує інтерфейс Printable

class Magazine : Printable

{

    private string title;

    public Magazine(string title)

    {

        this.title = title;

    }

    public void Print()

    {

        Console.WriteLine("Magazine: " + title);

    }

    // Статичний метод, що виводить на консоль назви лише журналів

    public static void PrintMagazines(Printable[] printables)

    {

        Console.WriteLine("\nMagazines Only:");

        foreach (var printable in printables)

        {

            if (printable is Magazine)

            {

                printable.Print();

            }

        }

    }

}

// Інтерфейс "Чоловічий Одяг"

interface IMensClothing

{

    void DressMan();

}

// Інтерфейс "Жіночий Одяг"

interface IWomensClothing

{

    void DressWoman();

}

// Абстрактний клас Одяг

abstract class Clothing

{

    protected string size;

    protected double cost;

    protected string color;

    public Clothing(string size, double cost, string color)

    {

        this.size = size;

        this.cost = cost;

        this.color = color;

    }

    public void DisplayInfo()

    {

        Console.WriteLine($"Size: {size}, Cost: {cost}, Color: {color}");

    }

}

// Клас Футболка, який реалізує інтерфейси "Чоловічий Одяг" та "Жіночий Одяг"

class TShirt : Clothing, IMensClothing, IWomensClothing

{

    public TShirt(string size, double cost, string color) : base(size, cost, color) { }

    public void DressMan()

    {

        Console.WriteLine("Man is dressed in T-shirt. ");

        DisplayInfo();

    }

    public void DressWoman()

    {

        Console.WriteLine("Woman is dressed in T-shirt. ");

        DisplayInfo();

    }

}

// Клас Штани, який реалізує інтерфейси "Чоловічий Одяг" та "Жіночий Одяг"

class Pants : Clothing, IMensClothing, IWomensClothing

{

    public Pants(string size, double cost, string color) : base(size, cost, color) { }

    public void DressMan()

    {

        Console.WriteLine("Man is dressed in Pants. ");

        DisplayInfo();

    }

    public void DressWoman()

    {

        Console.WriteLine("Woman is dressed in Pants. ");

        DisplayInfo();

    }

}

// Клас Спідниця, який реалізує інтерфейс "Жіночий Одяг"

class Skirt : Clothing, IWomensClothing

{

    public Skirt(string size, double cost, string color) : base(size, cost, color) { }

    public void DressWoman()

    {

        Console.WriteLine("Woman is dressed in Skirt. ");

        DisplayInfo();

    }

}

// Клас Краватка, який реалізує інтерфейс "Чоловічий Одяг"

class Tie : Clothing, IMensClothing

{

    public Tie(string size, double cost, string color) : base(size, cost, color) { }

    public void DressMan()

    {

        Console.WriteLine("Man is dressed in Tie. ");

        DisplayInfo();

    }

}

// Клас Ательє

class Atelier

{

    public void DressWoman(IWomensClothing[] womensClothing)

    {

        Console.WriteLine("Dressing Woman:");

        foreach (var clothing in womensClothing)

        {

            clothing.DressWoman();

            Console.WriteLine();

        }

    }

    public void DressMan(IMensClothing[] mensClothing)

    {

        Console.WriteLine("Dressing Man:");

        foreach (var clothing in mensClothing)

        {

            clothing.DressMan();

            Console.WriteLine();

        }

    }

}

class Program

{

    static void Main()

    {

        // task 1

        // Створення екземплярів класів

        IShape circle1 = new Circle { FirstDimension = 5.0 };

        IShape circle2 = new Circle { FirstDimension = 7.5 };

        IShape circle3 = new Circle { FirstDimension = 10.0 };

        IColoredShape coloredCircle1 = new ColoredCircle { FirstDimension = 3.0, Color = "Red" };

        IColoredShape coloredCircle2 = new ColoredCircle { FirstDimension = 6.0, Color = "Blue" };

        IColoredShape coloredCircle3 = new ColoredCircle { FirstDimension = 8.0, Color = "Green" };

        // Створення масиву та заповнення його екземплярами

        IShape[] shapesArray = { circle1, circle2, circle3, coloredCircle1, coloredCircle2, coloredCircle3 };

        // Метод, що викликає методи для кожного елемента масиву в залежності від інтерфейсу

        DisplayShapes(shapesArray);

        Console.WriteLine();

        //task 2

        // Створення масиву типу Printable

        Printable[] printables = {

            new Book("C# Programming"),

            new Magazine("C# Weekly"),

            new Book("Object-Oriented Design"),

            new Magazine("Tech Insights")

        };

        // Виклик методу Print() для кожного об'єкта в масиві

        foreach (var printable in printables)

        {

            printable.Print();

        }

        // Виклик статичного методу PrintMagazines

        Magazine.PrintMagazines(printables);

        Console.WriteLine();

        // task3

        // Створення масиву з усіх типів одягу

        Clothing[] clothingArray = {

            new TShirt("M", 20.0, "Red"),

            new Pants("L", 40.0, "Blue"),

            new Skirt("S", 30.0, "Black"),

            new Tie("L", 15.0, "Striped")

        };

        // Створення об'єкта Ательє та виклик методів одягу

        Atelier atelier = new Atelier();

        atelier.DressWoman(Array.FindAll(clothingArray, item => item is IWomensClothing).Cast<IWomensClothing>().ToArray());

        atelier.DressMan(Array.FindAll(clothingArray, item => item is IMensClothing).Cast<IMensClothing>().ToArray());

    }

    static void DisplayShapes(IShape[] shapes)

    {

        foreach (var shape in shapes)

        {

            shape.DisplayType();

            shape.DisplayArea();

            if (shape is IColoredShape coloredShape)

            {

                coloredShape.DisplayColor();

            }

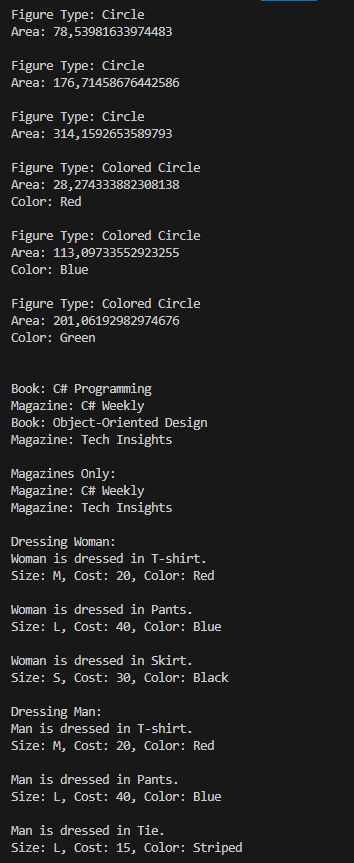
            Console.WriteLine();

        }

    }

}

1. **Результат роботи.**



1. **Посилання на проект.**