Note: All codes can be found at https://github.com/mgmarques/CSharp.

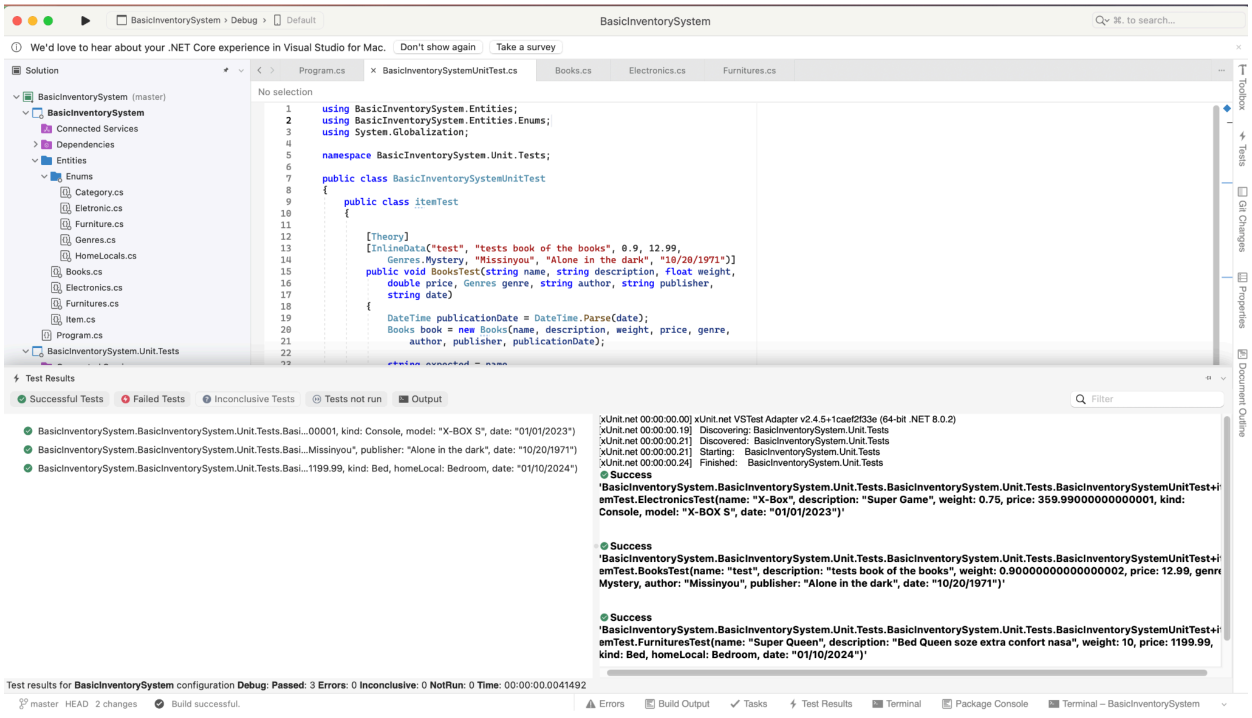
Exercise 1: Building a Simple Class Hierarchy for a Basic Inventory System

Statement:

Develop a basic inventory system that utilizes class hierarchies to manage different types of items in a store. Your system should include a base class named Item with common properties and specific derived classes for different item categories such as Book, Furniture, and Electronics. Demonstrate the use of constructors, properties, and method overriding.

Topics to Practice:

- Defining classes and objects
- Constructors and properties
- Inheritance and base classes
- Method overriding (Polymorphism)

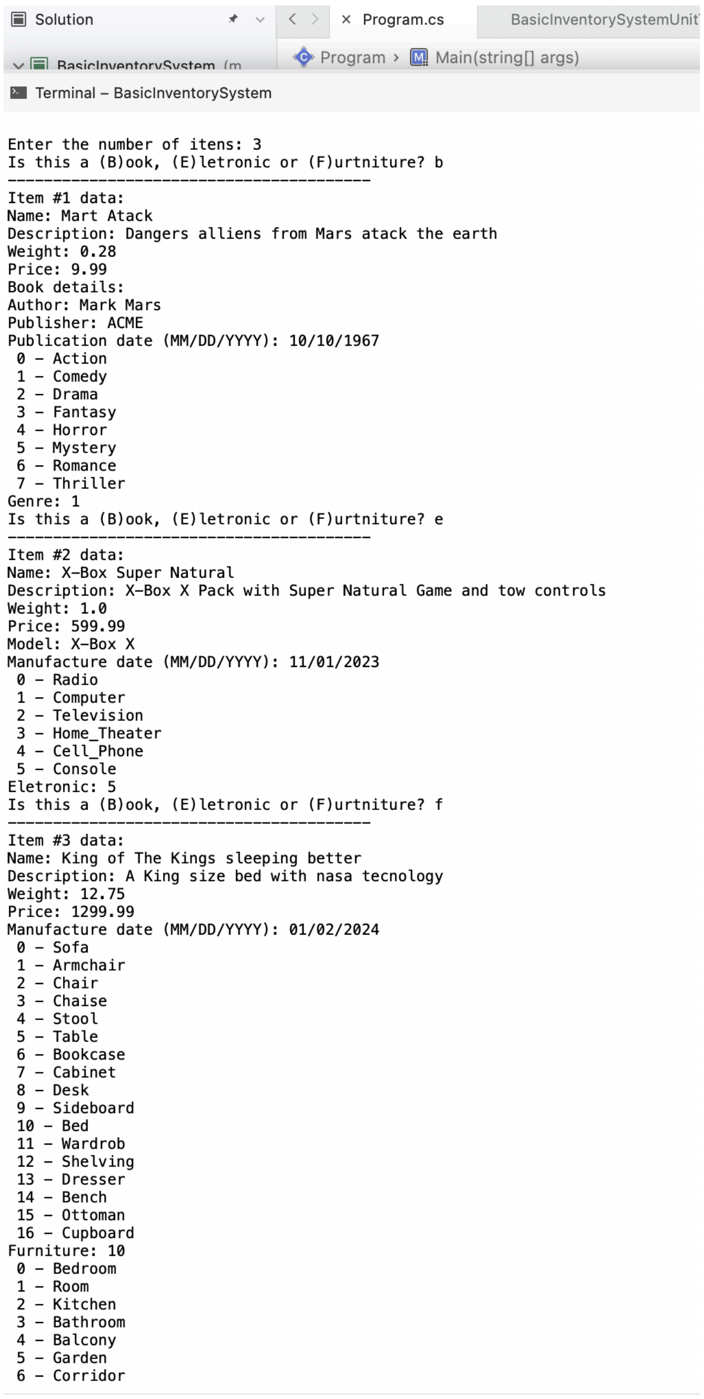


```
using BasicInventorySystem.Entities;
using BasicInventorySystem.Entities.Enums;
using System.Globalization;

namespace BasicInventorySystem.Unit.Tests;

public class BasicInventorySystemUnitTest
{
    public class itemTest
    {

        [Theory]
        [InlineData("test", "tests book of the books", 0.9, 12.99,
            Genres.Mystery, "Missinyou", "Alone in the dark", "10/20/1971")]
        public void BooksTest(string name, string description, float weight,
            double price, Genres genre, string author, string publisher,
            string date)
        {
            DateTime publicationDate = DateTime.Parse(date);
            Books book = new Books(name, description, weight, price, genre,
                author, publisher, publicationDate);

            string expected = name
```

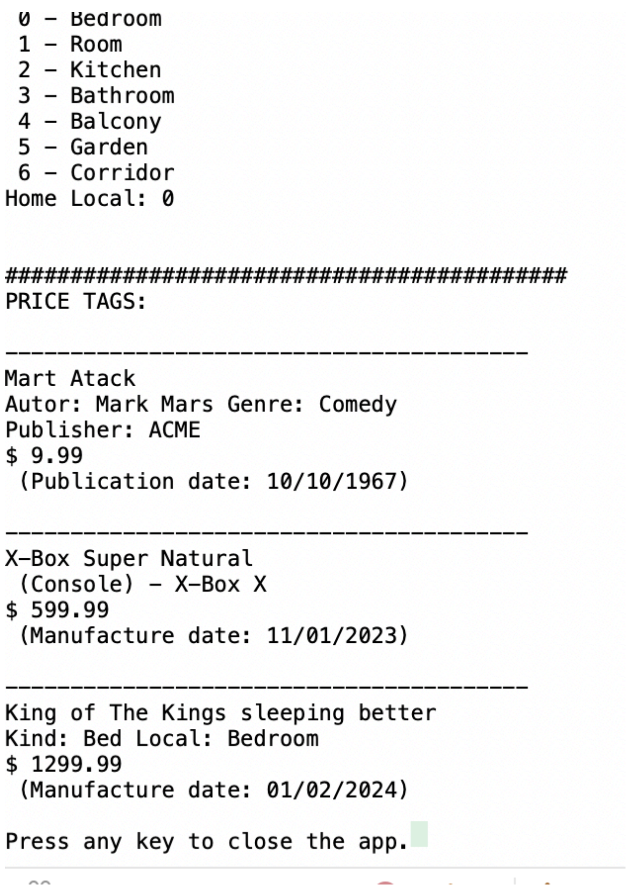
```
BasicInventorySystem.BasicInventorySystem.Unit.Tests.Basi...00001, kind: Console, model: "X-BOX S", date: "01/01/2023")
BasicInventorySystem.BasicInventorySystem.Unit.Tests.Basi...Missinyou", publisher: "Alone in the dark", date: "10/20/1971")
BasicInventorySystem.BasicInventorySystem.Unit.Tests.Basi...1199.99, kind: Bed, homeLocal: Bedroom, date: "01/10/2024")
```

```
[xUnit.net 00:00:00.00] xUnit.net VSTest Adapter v2.4.5+1caef2f33e (64-bit .NET 8.0.2)
[xUnit.net 00:00:00.19]   Discovering: BasicInventorySystem.Unit.Tests
[xUnit.net 00:00:00.21]   Discovered:  BasicInventorySystem.Unit.Tests
[xUnit.net 00:00:00.21]   Starting:    BasicInventorySystem.Unit.Tests
[xUnit.net 00:00:00.24]   Finished:    BasicInventorySystem.Unit.Tests
Success
'BasicInventorySystem.BasicInventorySystem.Unit.Tests.BasicInventorySystem.Unit.Tests.BasicInventorySystemUnitTest+i
emTest.ElectronicsTest(name: "X-Box", description: "Super Game", weight: 0.75, price: 359.99000000000001, kind:
Console, model: "X-BOX S", date: "01/01/2023")'

Success
'BasicInventorySystem.BasicInventorySystem.Unit.Tests.BasicInventorySystem.Unit.Tests.BasicInventorySystemUnitTest+i
emTest.BooksTest(name: "test", description: "tests book of the books", weight: 0.90000000000000002, price: 12.99, genre
Mystery, author: "Missinyou", publisher: "Alone in the dark", date: "10/20/1971")'

Success
'BasicInventorySystem.BasicInventorySystem.Unit.Tests.BasicInventorySystem.Unit.Tests.BasicInventorySystemUnitTest+i
emTest.FurnituresTest(name: "Super Queen", description: "Bed Queen soze extra confort nasa", weight: 10, price: 1199.99,
kind: Bed, homeLocal: Bedroom, date: "01/10/2024")'
```

Test results for **BasicInventorySystem** configuration **Debug**: **Passed:** 3 **Errors:** 0 **Inconclusive:** 0 **NotRun:** 0 **Time:** 00:00:00.0041492



```
Enter the number of itens: 3
Is this a (B)ook, (E)letronic or (F)urtniture? b
----------------------------------------
Item #1 data:
Name: Mart Atack
Description: Dangers alliens from Mars atack the earth
Weight: 0.28
Price: 9.99
Book details:
Author: Mark Mars
Publisher: ACME
Publication date (MM/DD/YYYY): 10/10/1967
 0 - Action
 1 - Comedy
 2 - Drama
 3 - Fantasy
 4 - Horror
 5 - Mystery
 6 - Romance
 7 - Thriller
Genre: 1
Is this a (B)ook, (E)letronic or (F)urtniture? e
----------------------------------------
Item #2 data:
Name: X-Box Super Natural
Description: X-Box X Pack with Super Natural Game and tow controls
Weight: 1.0
Price: 599.99
Model: X-Box X
Manufacture date (MM/DD/YYYY): 11/01/2023
 0 - Radio
 1 - Computer
 2 - Television
 3 - Home_Theater
 4 - Cell_Phone
 5 - Console
Eletronic: 5
Is this a (B)ook, (E)letronic or (F)urtniture? f
----------------------------------------
Item #3 data:
Name: King of The Kings sleeping better
Description: A King size bed with nasa tecnology
Weight: 12.75
Price: 1299.99
Manufacture date (MM/DD/YYYY): 01/02/2024
 0 - Sofa
 1 - Armchair
 2 - Chair
 3 - Chaise
 4 - Stool
 5 - Table
 6 - Bookcase
 7 - Cabinet
 8 - Desk
 9 - Sideboard
 10 - Bed
 11 - Wardrob
 12 - Shelving
 13 - Dresser
 14 - Bench
 15 - Ottoman
 16 - Cupboard
Furniture: 10
 0 - Bedroom
 1 - Room
 2 - Kitchen
 3 - Bathroom
 4 - Balcony
 5 - Garden
 6 - Corridor
```



```
 0 - Bedroom
 1 - Room
 2 - Kitchen
 3 - Bathroom
 4 - Balcony
 5 - Garden
 6 - Corridor
Home Local: 0


###########################################
PRICE TAGS:

----------------------------------------
Mart Atack
Autor: Mark Mars Genre: Comedy
Publisher: ACME
$ 9.99
 (Publication date: 10/10/1967)

----------------------------------------
X-Box Super Natural
 (Console) - X-Box X
$ 599.99
 (Manufacture date: 11/01/2023)

----------------------------------------
King of The Kings sleeping better
Kind: Bed Local: Bedroom
$ 1299.99
 (Manufacture date: 01/02/2024)

Press any key to close the app.
```

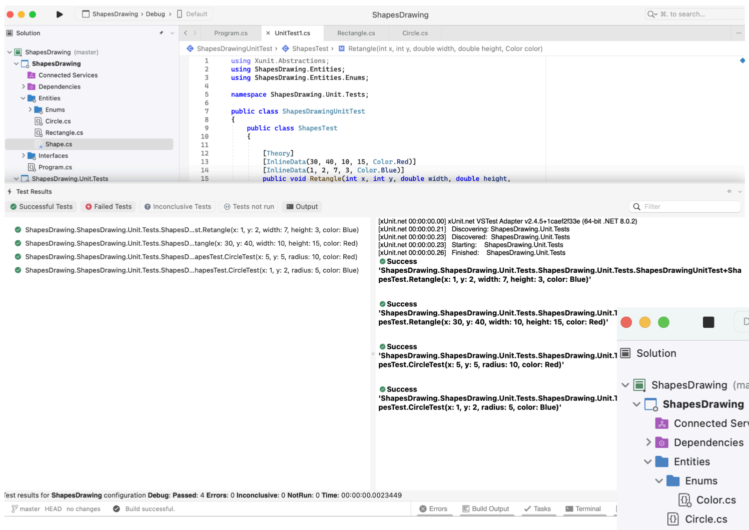
Exercise 2: Implementing Interfaces and Abstract Classes in a Shapes Drawing Application

Statement:

Create a shapes drawing application that demonstrates the use of interfaces and abstract classes. Your application should define an IShape interface and an abstract class Shape that implements the interface. Implement concrete shape classes like Circle and Rectangle that inherit from Shape. Each shape should be able to Draw itself and calculate its Area and Perimeter.

Topics to Practice:

- Interfaces and abstract classes
- Implementing interface methods
- Polymorphism and method overriding
- Encapsulation

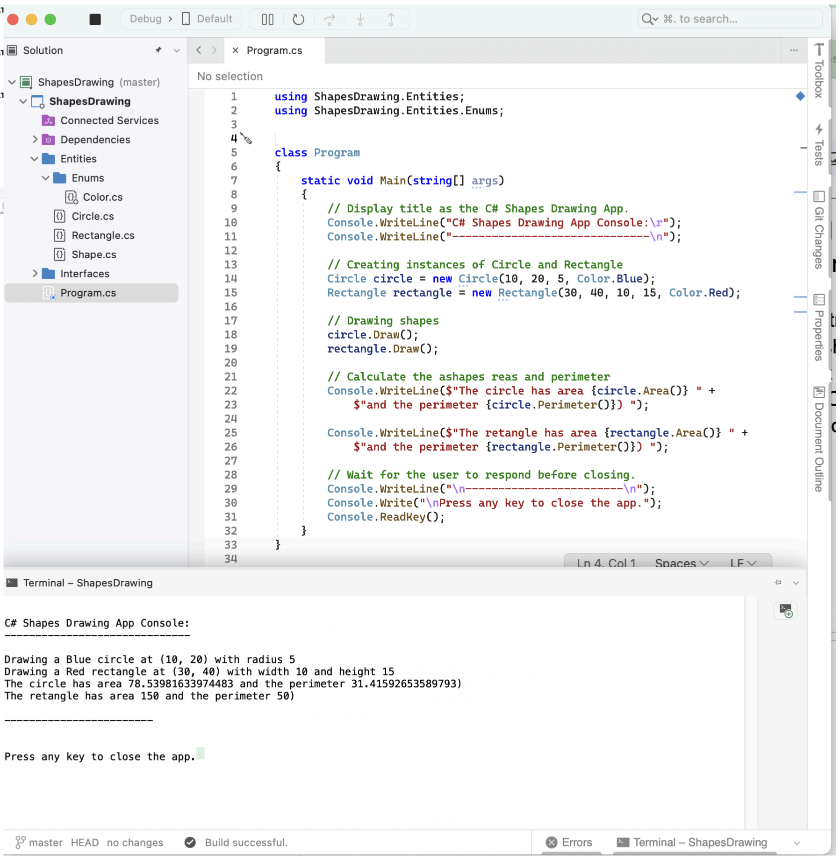


```
using Xunit.Abstractions;
using ShapesDrawing.Entities;
using ShapesDrawing.Entities.Enums;

namespace ShapesDrawing.Unit.Tests;

public class ShapesDrawingUnitTest
{
    public class ShapesTest
    {

        [Theory]
        [InlineData(30, 40, 10, 15, Color.Red)]
        [InlineData(1, 2, 7, 3, Color.Blue)]
        public void Retangle(int x, int y, double width, double height,
```

```
ShapesDrawing.ShapesDrawing.Unit.Tests.ShapesD...st.Retangle(x: 1, y: 2, width: 7, height: 3, color: Blue)
ShapesDrawing.ShapesDrawing.Unit.Tests.ShapesD...tangle(x: 30, y: 40, width: 10, height: 15, color: Red)
ShapesDrawing.ShapesDrawing.Unit.Tests.ShapesD...apesTest.CircleTest(x: 5, y: 5, radius: 10, color: Red)
ShapesDrawing.ShapesDrawing.Unit.Tests.ShapesD...hapesTest.CircleTest(x: 1, y: 2, radius: 5, color: Blue)
```

```
[xUnit.net 00:00:00.00] xUnit.net VSTest Adapter v2.4.5+1caef2f33e (64-bit .NET 8.0.2)
[xUnit.net 00:00:00.21]   Discovering: ShapesDrawing.Unit.Tests
[xUnit.net 00:00:00.23]   Discovered:  ShapesDrawing.Unit.Tests
[xUnit.net 00:00:00.23]   Starting:    ShapesDrawing.Unit.Tests
[xUnit.net 00:00:00.26]   Finished:    ShapesDrawing.Unit.Tests
Success
'ShapesDrawing.ShapesDrawing.Unit.Tests.ShapesDrawing.Unit.Tests.ShapesDrawingUnitTest+ShapesTest.Retangle(x: 1, y: 2, width: 7, height: 3, color: Blue)'

Success
'ShapesDrawing.ShapesDrawing.Unit.Tests.ShapesDrawing.Unit.T
pesTest.Retangle(x: 30, y: 40, width: 10, height: 15, color: Red)'

Success
'ShapesDrawing.ShapesDrawing.Unit.Tests.ShapesDrawing.Unit.T
pesTest.CircleTest(x: 5, y: 5, radius: 10, color: Red)'

Success
'ShapesDrawing.ShapesDrawing.Unit.Tests.ShapesDrawing.Unit.T
pesTest.CircleTest(x: 1, y: 2, radius: 5, color: Blue)'
```

est results for **ShapesDrawing** configuration **Debug: Passed:** 4 **Errors:** 0 **Inconclusive:** 0 **NotRun:** 0 **Time:** 00:00:00.0023449



```
using ShapesDrawing.Entities;
using ShapesDrawing.Entities.Enums;

class Program
{
    static void Main(string[] args)
    {
        // Display title as the C# Shapes Drawing App.
        Console.WriteLine("C# Shapes Drawing App Console:\r");
        Console.WriteLine("------------------------------\n");

        // Creating instances of Circle and Rectangle
        Circle circle = new Circle(10, 20, 5, Color.Blue);
        Rectangle rectangle = new Rectangle(30, 40, 10, 15, Color.Red);

        // Drawing shapes
        circle.Draw();
        rectangle.Draw();

        // Calculate the ashapes reas and perimeter
        Console.WriteLine($"The circle has area {circle.Area()} " +
            $"and the perimeter {circle.Perimeter()}) ");

        Console.WriteLine($"The retangle has area {rectangle.Area()} " +
            $"and the perimeter {rectangle.Perimeter()}) ");

        // Wait for the user to respond before closing.
        Console.WriteLine("\n------------------------\n");
        Console.Write("\nPress any key to close the app.");
        Console.ReadKey();
    }
}
```

```
C# Shapes Drawing App Console:
------------------------------

Drawing a Blue circle at (10, 20) with radius 5
Drawing a Red rectangle at (30, 40) with width 10 and height 15
The circle has area 78.53981633974483 and the perimeter 31.41592653589793)
The retangle has area 150 and the perimeter 50)

------------------------


Press any key to close the app.
```