R.S.KARTHIKEYAN
13/8/2025

Interface:

```
using System;
using System.Collections.Generic;
using System.Globalization;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Day_7
{
    public interface IBrush
    {
       public string Name { get; set; }

       public string Type { get; set; }

       public int Price { get; set; }


    }
}
```

Brush:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Day_7
{
    public class Brush : IBrush
    {
        public string Name { get; set; }

        public string Type { get; set; }

        public int Price { get; set; }

        public Brush(string Name,string Type , int Price)
        {
            this.Name = Name;
            this.Type = Type;
            this.Price = Price;
        }
    }
}
```

Paint:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Day_7
{
    public class Painter
    {
        public string Name { get; set; }

        public string Gender { get; set; }

        public IBrush Brush { get; set; }

        public Painter(string Name, string Gender, IBrush Brush)
        {
            this.Name = Name;
            this.Gender = Gender;
            this.Brush = Brush;
        }
    }
}
```

Program.cs

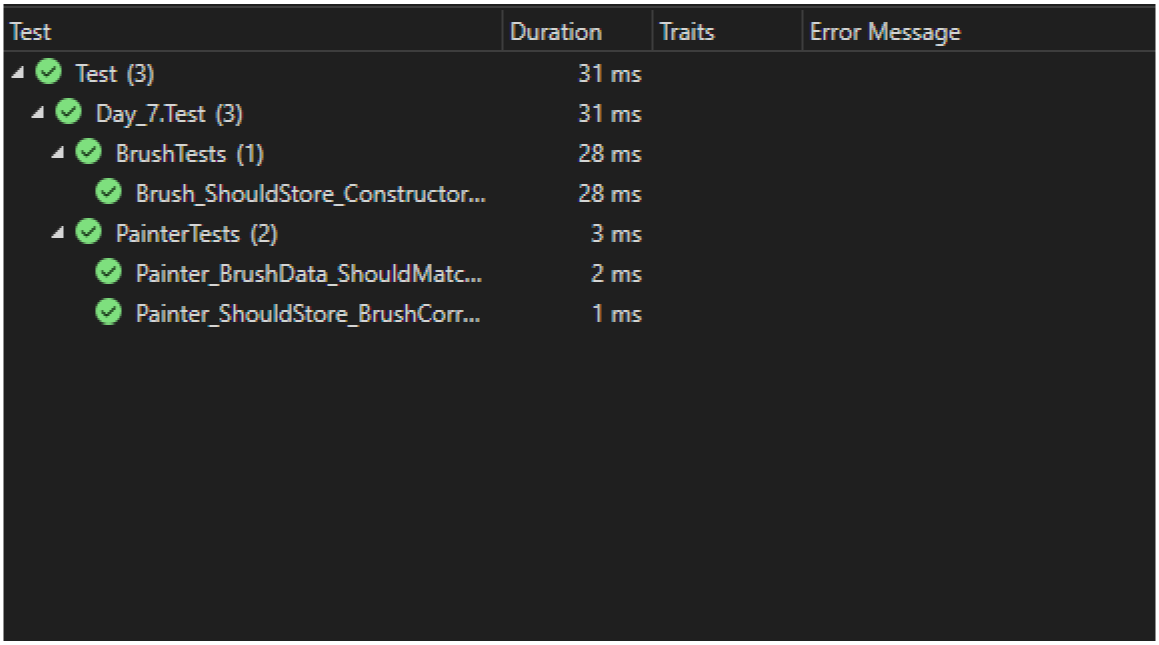
```
using Day_7;

IBrush RollerBrush = new Brush("Repto Brush","Roller",300);

IBrush Brush = new Brush("Repto Brush", "Normal", 150);

Painter paint1 = new Painter("Karthikeyan", "Male", RollerBrush);
Painter paint2 = new Painter("Sugi", "Female", Brush);


Console.WriteLine($"Painter  Name: {paint1.Name} \n Paint Quality : {paint1.Gender} \n Brush Name : {paint1.Brush.Name} \n" +
   $" Brush Price : {paint1.Brush.Price} \n Brush Type : {paint1.Brush.Type} ");

Console.WriteLine($"Painter  Name: {paint2.Name} \n Paint Quality : {paint2.Gender} \n Brush Name : {paint2.Brush.Name} \n" +
   $" Brush Price : {paint2.Brush.Price} \n Brush Type : {paint2.Brush.Type} ");
```

Test:

```
using NUnit.Framework;
using Day_7;

namespace Day_7.Test
{
   public class BrushTests
   {
      [Test]
      public void Brush_ShouldStore_ConstructorValues()
      {
         var brush = new Brush("Repto Brush", "Roller", 300);

         Assert.That(brush.Name, Is.EqualTo("Repto Brush"));
         Assert.That(brush.Type, Is.EqualTo("Roller"));
         Assert.That(brush.Price, Is.EqualTo(300));
      }
   }

   public class PainterTests
   {
      [Test]
      public void Painter_ShouldStore_BrushCorrectly()
      {
         var brush = new Brush("Repto Brush", "Normal", 150);
         var painter = new Painter("Beaste", "Female", brush);

         Assert.That(painter.Name, Is.EqualTo("Beaste"));
         Assert.That(painter.Gender, Is.EqualTo("Female"));
         Assert.That(painter.Brush, Is.EqualTo(brush));
         Assert.That(painter.Brush.Name, Is.EqualTo("Repto Brush"));
      }
```

```
        [Test]
        public void Painter_BrushData_ShouldMatchAssigned()
        {
            var rollerBrush = new Brush("Repto Brush", "Roller", 300);
            var painter = new Painter("Karthikeyan", "Male", rollerBrush);

            Assert.Multiple(() =>
            {
                Assert.That(painter.Brush.Name, Is.EqualTo("Repto Brush"));
                Assert.That(painter.Brush.Type, Is.EqualTo("Roller"));
                Assert.That(painter.Brush.Price, Is.EqualTo(300));
            });
        }
    }
}
```

Output:



Test	Duration	Traits	Error Message
◢ ✓ Test (3)	31 ms		
◢ ✓ Day_7.Test (3)	31 ms		
◢ ✓ BrushTests (1)	28 ms		
✓ Brush_ShouldStore_Constructor...	28 ms		
◢ ✓ PainterTests (2)	3 ms		
✓ Painter_BrushData_ShouldMatc...	2 ms		
✓ Painter_ShouldStore_BrushCorr...	1 ms		