Factory Method

Padrão de design

Exemplo

por Felipe Arantes

Introdução

O exemplo criado consiste em um gerenciamento de logística que precisa realizar o transporte terrestre, aquático e aéreo.

- Se o transporte for terrestre, precisa-se dizer qual o tamanho da caixa a ser transportada. Se a caixa for pequena o transporte será via motocicleta, se for media será por carro e se for grande por caminhão.

- Se for aquático, o transporte será por navio.

- Se for aéreo, o transporte será por avião.

Interfaces e Enums

```
public interface ITransport
{
    public void Deliver();
}
```

```
public interface IBox
{
    public SizeEnum Size { get; }
}
```

```
public enum SizeEnum
{
    Small = 1,
    Medium = 2,
    Large = 3,
}
```

Caixas

```
public class SmallBox : IBox
{
    public SizeEnum Size => SizeEnum.Small;
}

public class MediumBox : IBox
{
    public SizeEnum Size => SizeEnum.Medium;
}

public class LargeBox : IBox
{
    public SizeEnum Size => SizeEnum.Large;
}
```

Veículos Terrestres

```
public class Motorbike : ITransport
{
    public void Deliver()
    {
        Console.WriteLine("Small Box");
        Console.WriteLine("Motorbike Delivery");
    }
}
```

```
public class Truck : ITransport
{
    public void Deliver()
    {
        Console.WriteLine("Large Box");
        Console.WriteLine("Truck Delivery");
    }
}
```

```
public class Car : ITransport
{
    public void Deliver()
    {
        Console.WriteLine("Medium Box");
        Console.WriteLine("Car Delivery");
    }
}
```

Veículos Aéreos e Aquáticos

```
public class Plane : ITransport
{
    public void Deliver()
    {
        Console.WriteLine("Plane Delivery");
    }
}


public class Ship : ITransport
{
    public void Deliver()
    {
        Console.WriteLine("Sea Delivery");
    }
}
```

Fábrica Padrão – Superclasse

```
public abstract class LogisticsFactory
{
    public abstract ITransport CreateTransport();
}
```

Fábrica Terrestre – Subclasse

```
public class RoadFactory : LogisticsFactory
{
    private readonly IBox _box;

    public RoadFactory(IBox box)
    {
        _box = box;
    }

    public override ITransport CreateTransport()
    {
        if ( _box is null )
            throw new NotSupportedException();

        return _box.Size switch
        {
            SizeEnum.Small => new Motorbike(),
            SizeEnum.Medium => new Car(),
            SizeEnum.Large => new Truck(),
            _ => throw new NotSupportedException(),
        };
    }
}
```

Fábrica Aquática – Subclasse

```
public class SeaFactory : LogisticsFactory
{
    public override ITransport CreateTransport()
    {
        return new Ship();
    }
}
```

Fábrica Aérea – Subclasse

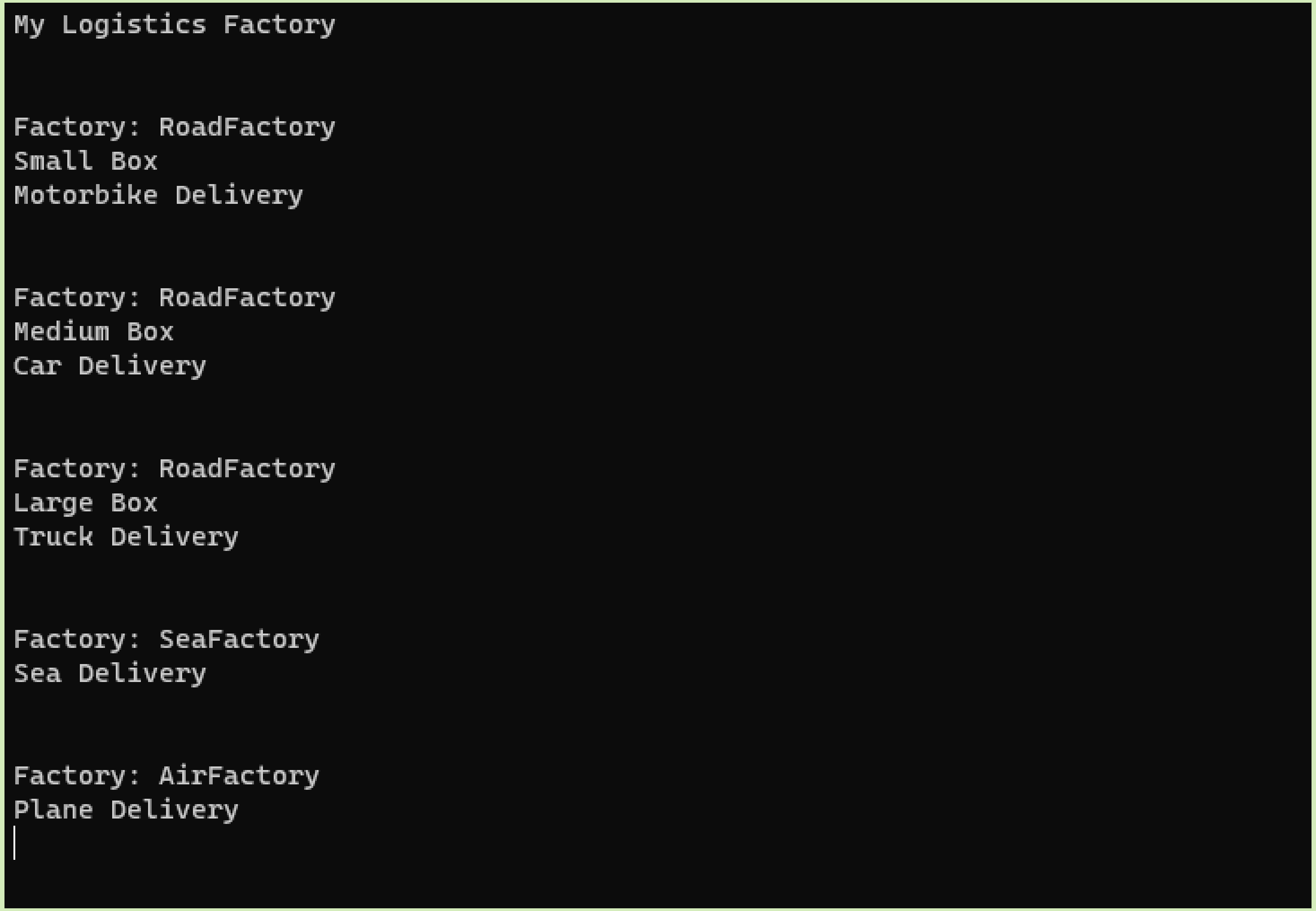
```
public class AirFactory : LogisticsFactory
{
    public override ITransport CreateTransport()
    {
        return new Plane();
    }
}
```

Program - Execução

```
Console.WriteLine("My Logistics Factory");

// An array of creators
var logisticsFactory = new LogisticsFactory[]
{
    new RoadFactory(new SmallBox()),
    new RoadFactory(new MediumBox()),
    new RoadFactory(new LargeBox()),
    new SeaFactory(),
    new AirFactory()
};

// Iterate over creators and create products
foreach ( var factory in logisticsFactory )
{
    Console.WriteLine();
    Console.WriteLine();
    Console.WriteLine("Factory: " + factory.GetType().Name);

    var transport = factory.CreateTransport();
    transport.Deliver();
}
```

```
// Wait for user
Console.ReadKey();
```

Resultado



```
My Logistics Factory


Factory: RoadFactory
Small Box
Motorbike Delivery


Factory: RoadFactory
Medium Box
Car Delivery


Factory: RoadFactory
Large Box
Truck Delivery


Factory: SeaFactory
Sea Delivery


Factory: AirFactory
Plane Delivery
```