The Complete ASP.Net Core MVC With Entity Framework Core

Program.cs:

using GameTOP;

class Program

    {

**static** void Main(string[] args)

        {

            var jogo = new **JogoFODA**(new **Jogador1**("Ronaldo"), new **Jogador2**());

            jogo.IniciarJogo();

        }

    }

JogoFODA.cs:

using System;

namespace GameTOP

{

**public** class JogoFODA

        {

**private** **readonly** **iJogador** \_jogador1;

**private** **readonly** **iJogador** \_jogador2;

**public** JogoFODA(**iJogador** jogador1, **iJogador** jogador2)

            {

                \_jogador1 = jogador1;

                \_jogador2 = jogador2;

            }

**public** void IniciarJogo()

            {

               Console.Write(\_jogador1.Corre());

               Console.Write(\_jogador1.Chuta());

               Console.Write(\_jogador1.Passe());

               Console.Write(\_jogador2.Corre());

               Console.Write(\_jogador2.Chuta());

               Console.Write(\_jogador2.Passe());

            }

        }

}

Jogador1.cs:

using System;

namespace GameTOP

{

**public** class Jogador1 : **iJogador**

    {

**public** **readonly** string nome;

**public** Jogador1(**String** \_nome)

        {

            nome = \_nome;

        }

**public** string Chuta()

        {

            return $"{nome} chutou \n";

        }

**public** string Corre()

        {

            return $"{nome} correu \n";

        }

**public** string Passe()

        {

            return $"{nome} tocou \n";

        }

    }

}

Jogador2.cs:

namespace GameTOP

{

**public** class Jogador2 : **iJogador**

    {

**public** string Chuta()

        {

            return "Maradona estas pateando";

        }

**public** string Corre()

        {

            return "Maradona estas corriendo";

        }

**public** string Passe()

        {

            return "Maradona estas passando";

        }

    }

}

iJogador.cs:

namespace GameTOP

{

**public** interface iJogador

    {

         string Corre();

         string Chuta();

         string Passe();

    }

}

Conceito de interface:

Colocando jogador como uma interface ou contrato (iJogador) a classe que implementa o contrato pode ser implementada de qualquer forma que na hora de passar os argumentos, passamos como tipo de interface, ou seja, ele verifica se os métodos obrigatórios presentes na interface estão implementados e, caso positivo, não é apresentado erro.

Um exemplo onde isso pode ser visto é em classes de jogador de RPG, por exemplo:

Todos terão vida, nome, historia. Alguns serão implementados com magia, outros com armas, etc.

Veja o exemplo abaixo, Jogador1 e Jogador2 foram implementados de forma diferentes mas ambos implementaram os métodos obrigatórios definidos pela interface:

class Program

    {

**static** void Main(string[] args)

        {

            var jogo = new **JogoFODA**(new **Jogador1**("Ronaldo"), new **Jogador2**());

            jogo.IniciarJogo();

        }

    }

Jogo foda nesse caso estava esperando uma interface, demos como parâmetro uma classe que implementava esta interface.

Exemplos de inserção de referencia:

dotnet sln GameTOP.sln add GameTOP.Interface/GameTOP.Interface.csproj

dotnet add GameTOP/GameTOP.csproj reference GameTOP.Interface/GameTOP.Interface.csproj

dotnet new classlib -n GameTOP.lib

dotnet new sln -n GameTOP