Linguagem de Programação II

Análise e Desenvolvimento de Sistemas

2º semestre de 2019

Prof. Me. Renato Carioca Duarte

Polimorfismo

- Polimorfismo significa muitas formas, na orientação a objetos você pode enviar uma mesma mensagem para **diferentes objetos** e fazê-los responder da maneira correta.
- Por exemplo, você pode enviar a mensagem mover() para cada objeto semelhante a um veículo (carro, trem ou navio) e cada um vai se comportar de maneira diferente para atender a sua solicitação. Nesse caso, a mensagem mover() é polimórfica.

Polimorfismo

"Polimorfismo é o princípio pelo qual duas ou mais classes derivadas de uma mesma superclasse podem invocar métodos que têm a mesma identificação (assinatura) mas comportamentos distintos, especializados para cada classe derivada, usando para tanto uma referência a um objeto do tipo da classe base"

Polimorfismo

Existem dois tipos básicos de polimorfismo:

1. Polimorfismo em tempo de compilação (Overloading/Sobrecarga);

   Onde podemos permitir que classes forneçam diferentes implementações de métodos que são chamados com o mesmo nome;

2. Polimorfismo em tempo de execução (Overriding/Sobrescrita);

   Onde podemos invocar métodos da classe derivada através da classe base em tempo de execução;

Polimorfismo usando Sobrecarga

- O polimorfismo em tempo de compilação utiliza a sobrecarga de métodos e operadores sendo também chamado de ligação precoce (*early binding*). A utilização da sobrecarga de métodos realiza a tarefa com distintos parâmetros de entrada.
- Assim como outras linguagens de programação modernas, C# permite que um método seja **sobrecarregado**. Basicamente, quando você define dois ou mais métodos com o mesmo nome e diferindo apenas pelo número (ou tipo) de parâmetros, o método em questão é dito ser *sobrecarregado*.
- Métodos sobrecarregados são bastante úteis quando você deseja criar métodos que realizam operações semelhantes, porém sobre parâmetros de tipos diferentes ou em um número diferente de parâmetros.
- Por exemplo, suponha que você precise criar métodos para somar os valores de seus parâmetros. Sem o uso de sobrecarga de métodos você teria que criar um método diferente para cada uma das operações, mesmo apesar delas fazerem praticamente a mesma coisa:

Polimorfismo usando Sobrecarga

```
class Calcular
{
    public int Soma(int num1, int num2)
    {
        return (num1 + num2);
    }

    public int Soma(int num1, int num2, int num3)
    {
        return (num1 + num2 + num3);
    }
}
```

Polimorfismo usando Sobrecarga

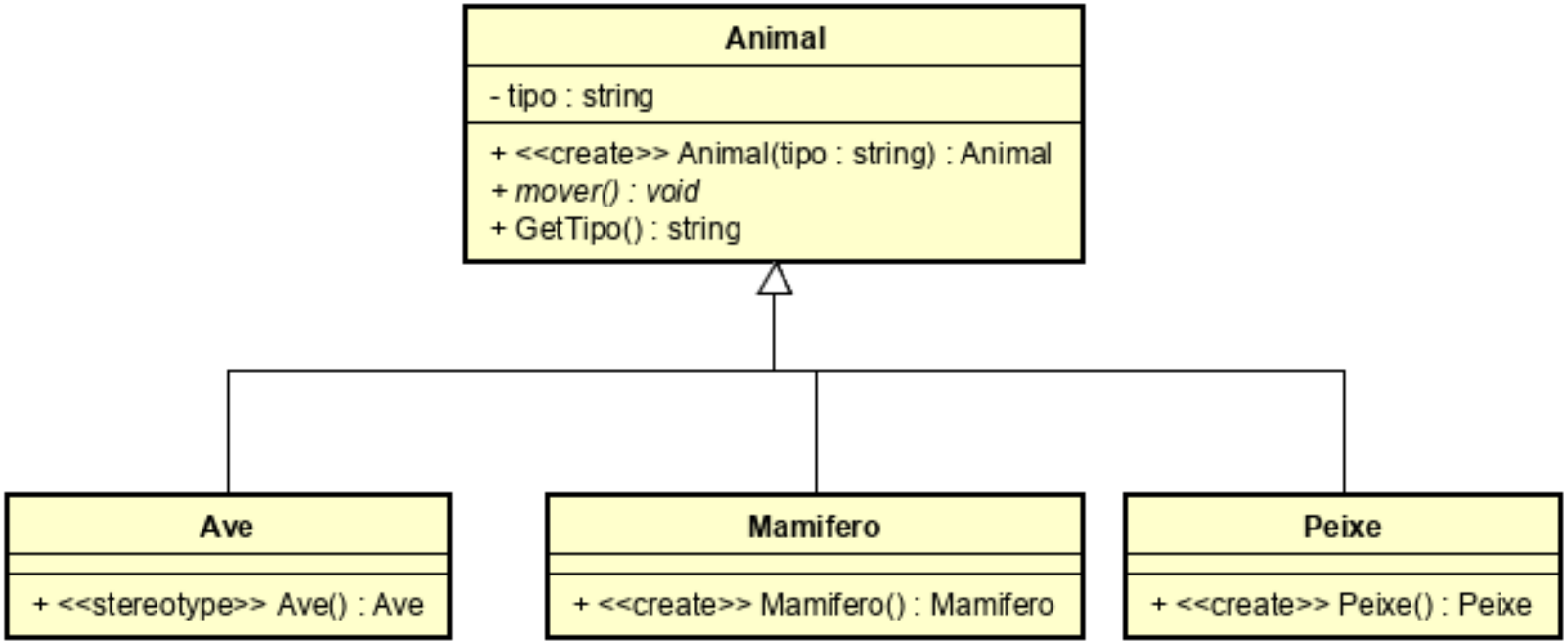
```
class Program
{
    static void Main(string[] args)
    {
        Calcular calc = new Calcular();
        Console.WriteLine("\nPolimorfismo com sobrecarga\n");
        Console.WriteLine("Somando 2 números...");
        Console.WriteLine(calc.Soma(45,43));
        Console.WriteLine("Somando 3 números...");
        Console.WriteLine(calc.Soma(45, 43, 100 ));
        Console.ReadKey();
    }
}
```

Polimorfismo usando Sobrescrita

- O polimorfismo em tempo de execução pode ser feito usando herança e métodos virtuais. Quando sobrescrevemos(override) os métodos virtuais estamos alterando o comportamento dos métodos para a classe derivada. Isto também é conhecido como ligação tardia (*late binding*).
- A implementação do polimorfismo pode ser realizada fazendo uso de interfaces, ou classes abstratas, onde ocorrem apenas a implementação das assinaturas dos métodos, ou seja, do contrato.
- Desta forma o comportamento deve ser implementado nas classes concretas que implementam as interfaces ou estendem as classes abstratas.

Polimorfismo usando Sobrescrita

- Exemplo:



Polimorfismo usando Sobrescrita

```
class Animal
{
    private string tipo;

    public Animal(string tipo) { this.tipo = tipo; }

    public string getTipo ()
    {
        return tipo;
    }

    public virtual string mover() { return "mover";  }
}
```

Polimorfismo usando Sobrescrita

```
class Ave : Animal
{
    public Ave(string tipo) : base(tipo)
    {
    }

    public override string mover()
    {
        return "voa";
    }
}
```

Polimorfismo usando Sobrescrita

```
class Mamifero : Animal
{
    public Mamifero(string tipo) : base(tipo)
    {
    }

    public override string mover()
    {
        return "andar";
    }

}
```

Polimorfismo usando Sobrescrita

```
class Peixe : Animal
{
    public Peixe(string tipo) : base(tipo)
    {
    }
    public override string mover()
    {
        return "nadar";
    }
}
```

Polimorfismo usando Sobrescrita

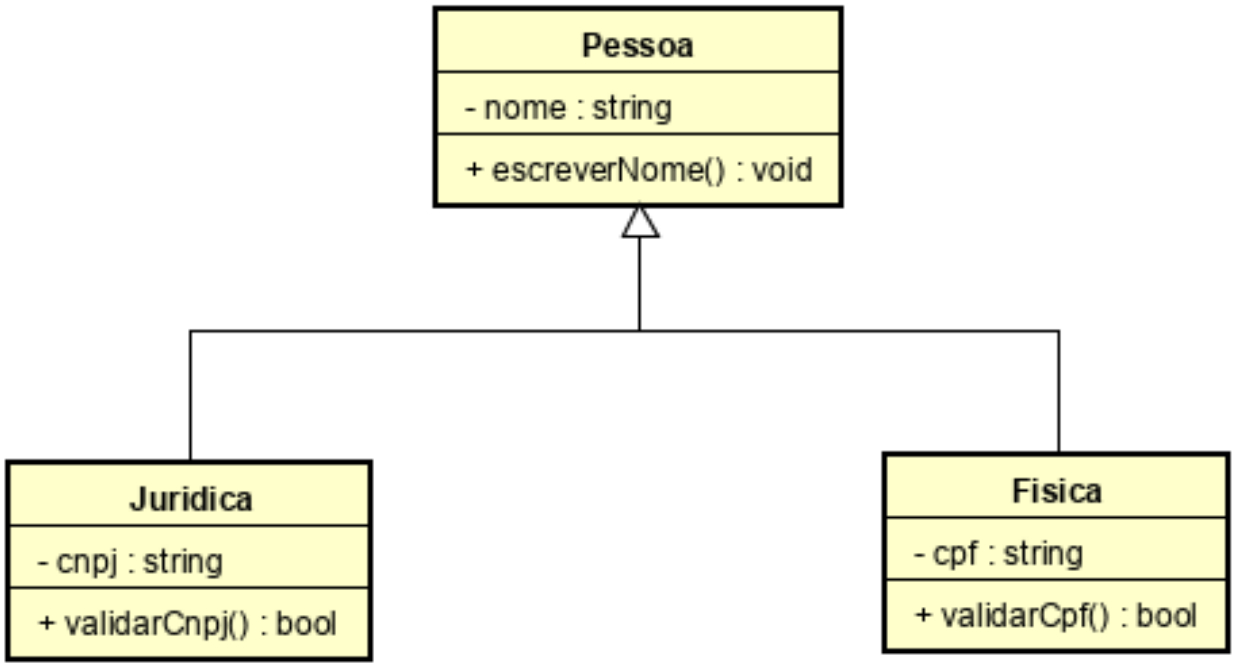
```
using System.Collections;

namespace polimorfismo_sobrescrita
{
    class Program
    {
        static void Main(string[] args)
        {
            ArrayList lista_animais = new ArrayList();
            lista_animais.Add(new Ave("rouxinol"));
            lista_animais.Add(new Peixe("carpa"));
            lista_animais.Add(new Mamifero("coelho"));
            foreach ( Animal a in lista_animais)
            {
                Console.WriteLine(a.getTipo());
                Console.WriteLine(a.mover());
            }
            Console.ReadKey();
        }
    }
}
```

Exercício

- O Polimorfismo permite, em uma de suas metodologias de aplicação, que diferentes classes tenham métodos com a mesma assinatura (mesmo contrato), porém estes métodos (em suas respectivas classes) podem possuir comportamentos diferentes, de acordo à necessidade de cada classe que o implementa.
- A implementação do polimorfismo pode ser realizada fazendo uso de interfaces, ou classes abstratas, onde ocorrem apenas a implementação das assinaturas dos métodos, ou seja, do contrato.
- Desta forma o comportamento deve ser implementado nas classes concretas que implementam as interfaces ou estendem as classes abstratas.

Exercício

- Escreva o código C# para a herança abaixo:



Exercício

- Escreva o código a interface e das 3 classes que implementam o polimorfismo.