Aula 16

Prof: Henrique Augusto Maltauro

Desenvolvendo Algoritmos

Programação Orientada a Objetos (POO)

Programação Orientada a Objetos (POO)

Conforme nós já havíamos visto anteriormente, dentro da área de programação, nós temos os paradigmas de programação, que são uma maneira de classificar as linguagens de programação de acordo com o seu comportamento.

Programação Orientada a Objetos (POO)

Entre esses paradigmas de programação, nós temos a programação estruturada (que o paradigma que estávamos utilizando até o momento) e a programação orientada a objetos (POO).

Para entendermos a POO, vamos primeiro revisar a forma como a programação estruturada funciona, e tratar de diferenciar os dois paradigmas.

Programação Orientada a Objetos (POO)

Na programação estruturada, o programa é basicamente escrito em uma única função podendo, é claro, ser quebrado em subfunções.

Mas o fluxo do programa continua o mesmo, como se pudéssemos copiar e colar o código das subfunções em uma única função, de tal forma que, no final, só haja uma grande função que execute todo o programa.

Programação Orientada a Objetos (POO)

Além disso, o acesso às variáveis não possuem muitas restrições na programação estruturada.

Em linguagens de programação estruturada, restringir o acesso a uma variável se limita a dizer se ela é visível ou não dentro de uma função, mas não se consegue definir que uma variável pode ser acessada por apenas algumas funções do programa.

Programação Orientada a Objetos (POO)

A POO traz uma abordagem muito diferente para a forma como o código é executado.

A execução das funções são totalmente separadas umas das outras, e existe uma restrição de acesso tanto nas funções, como nas variáveis.

Programação Orientada a Objetos (POO)

Para entendermos como funciona o fluxo da POO, primeiramente nós precisamos compreender que, o intuito da POO é fazer uma abstração digital dos objetos do mundo real.

Uma abstração é o processo de fazer uma imagem mental, de um conceito, um objeto, uma ideia, etc, e imaginar cada um dos aspectos daquilo, tanto os aspectos de características, como os aspectos de processos que precisam ser executados.

Programação Orientada a Objetos (POO)

Dentro disso, o objetivo da POO, é de imaginar os objetos do mundo real, como objetos digitais.

Nesse contexto, definimos que a POO possui dois conceitos chaves, no qual todo o resto é construído em cima:

- Classes
- Objetos

Programação Orientada a Objetos (POO)

- Classes (vamos imaginar um carro)

Todos os carros, independente do modelo, tem atributos similares, ou seja, todos eles tem um motor, todos eles tem 4 rodas, todos eles tem uma quantidade de portas, todos eles tem uma cor, todos eles tem um volante, todos eles tem um cambio, etc.

Programação Orientada a Objetos (POO)

- Classes (vamos imaginar um carro)

Todos os carros, independente do modelo, tem funcionalidades similares, ou seja, todos eles aceleram, todos eles freiam, todos eles viram pra esquerda, todos eles viram pra direita, etc.

Programação Orientada a Objetos (POO)

- Classes (vamos imaginar um carro)

Dentro de todo esse contexto, podemos dizer que a ideia de um carro, é a perfeita representação de uma classe.

Essa ideia representa uma receita, representa um molde, no qual a partir disso, vai ser construído um objeto.

Programação Orientada a Objetos (POO)

- Objetos (vamos imaginar um carro do modelo Saveiro)

A Saveiro tem diversos atributos, tem duas portas, tem direção hidráulica, tem dois airbags, tem freio ABS, tem ar-condicionado, tem um motor 1.6 a gasolina e de 4 cilindros, tem pneus de 15 polegadas, tem vidros elétricos, têm os bancos revestidos com tecido, etc.

Programação Orientada a Objetos (POO)

- Objetos (vamos imaginar um carro do modelo Saveiro)

A Saveiro tem diversas funcionalidades, ela acelera de acordo com o motor que ela tem, ela freia de acordo com o freio que ela tem, ela vira pra esquerda ou pra direita com um certa facilidade, de acordo com a direção que ela tem, etc.

Programação Orientada a Objetos (POO)

- Objetos (vamos imaginar um carro do modelo Saveiro)

Dentro de todo esse contexto, podemos dizer que a Saveiro, é a perfeita representação de um objeto.

A Saveiro não é mais apenas uma ideia, como quando estávamos falando sobre classes, agora ela tem todos os atributos e funcionalidades bem definidas.

Programação Orientada a Objetos (POO)

- Classes

Uma classe, representa de forma genérica a abstração de um objeto, possuindo um conjunto de atributos e funcionalidades que definem aquele objeto.

É uma receita, é um molde, para construir um objeto.

Programação Orientada a Objetos (POO)

- C#: Classes

Dentro do C#, representamos uma classe pela palavra-chave class.

Programação Orientada a Objetos (POO)

- C#: Classes

```
public class Carro
{
    // Bloco de código da classe
}
```

Programação Orientada a Objetos (POO)

- Classes (método construtor)

Sempre que uma classe for utilizada para a criação de um objeto, esse processo de criação é feito por um método construtor, que é responsável por construir um objeto a partir daquela classe.

Esse método, ele está presente dentro da classe, e recebe o mesmo nome da classe.

Programação Orientada a Objetos (POO)

- C#: Classes (método construtor)

```
public class Carro
{
    public Carro()
    {
        // Bloco de código do construtor
    }
}
```

Programação Orientada a Objetos (POO)

- Objetos

Um objeto é a representação da construção de uma classe, ou seja, é a classe de forma concreta.

Tende a representar um objeto do mundo real.

Programação Orientada a Objetos (POO)

- Objetos (instanciação)

A criação do objeto é feita executando o método construtor de uma classe, e esse processo é chamado de instanciação.

Também podemos definir que um objeto é a instância de uma classe.

Programação Orientada a Objetos (POO)

- C#: Objetos (instanciação)

```
public class Carro
{
    public Carro() { }
}
```

```
public static void Main()
{
    Carro saveiro = new Carro();
}
```

Programação Orientada a Objetos (POO)

- Classes/Objetos (atributos)

Os atributos, são as variáveis que uma classe vai definir, e que serão utilizadas pelos objetos para representar suas propriedades.

Programação Orientada a Objetos (POO)

- Classes/Objetos (atributos)

Através do ponto . podemos utilizar o atributo e definir um valor para ele, da mesma maneira que se define valores para variáveis normais.

Esse processo de utilizar um atributo, é chamado de invocar uma variável.

Programação Orientada a Objetos (POO)

- C#: Classes/Objetos (atributos)

```
public class Carro
{
    public string Modelo;
    public string Motor;
    public string Cor;
    public int QuantidadePortas;
}
```

```
public static void Main()
{
    Carro saveiro = new Carro();

    saveiro.Modelo = "Saveiro 2023";
    saveiro.Motor = "1.6 de 4 cilindros";
    saveiro.Cor = "Branco";
    saveiro.QuantidadePortas = 2;
}
```

Programação Orientada a Objetos (POO)

- Classes/Objetos (atributos)

Esses atributos, podem também podem ter o seu valor definido pelo método construtor.

Quando definidos pelo método construtor, normalmente, o método construtor recebe esses valores como parâmetros.

Programação Orientada a Objetos (POO)

- C#: Classes/Objetos (atributos)

```
public class Carro
{
    public string Modelo;
    public string Motor;
    public string Cor;
    public int QuantidadePortas;

    public Carro(string modelo, string motor, string cor, int quantidadePortas)
    {
        Modelo = modelo;
        Motor = motor;
        Cor = cor;
        QuantidadePortas = quantidadePortas;
    }
}
```

Programação Orientada a Objetos (POO)

- C#: Classes/Objetos (atributos)

```
public static void Main()
{
    Carro saveiro = new Carro("Saveiro 2023", "1.6 de 4 cilindros", "Branco", 2);
}
```

Programação Orientada a Objetos (POO)

- Classes/Objetos (métodos)

Os métodos vão ser definidos em uma classe, e serão utilizados pelos objetos para executar as suas funcionalidades.

Similar ao atributo, o processo de executar um método de um objeto, é chamado de invocar um método.

Programação Orientada a Objetos (POO)

- C#: Classes/Objetos (métodos)

```
public class Carro
{
    public void Acelerar()
    {
        // Bloco de código
    }

    public void Freiar()
    {
        // Bloco de código
    }
}
```

```
public static void Main()
{
    Carro saveiro = new Carro();

    saveiro.Acelerar();
    saveiro.Freiar();
}
```

Programação Orientada a Objetos (POO)

- Quatro pilares da POO

Toda a estrutura da POO se fortalece com as suas quatro principais características.

- Abstração
- Herança
- Encapsulamento
- Polimorfismo

Programação Orientada a Objetos (POO)

- Quatro pilares da POO (abstração)

Como já dito anteriormente a abstração é o processo de fazer uma imagem mental, de um conceito, um objeto, uma ideia, etc, e imaginar cada um dos aspectos daquilo, tanto os aspectos de características, como os aspectos de processos que precisam ser executados.

Programação Orientada a Objetos (POO)

- Quatro pilares da POO (abstração)

E dentro da POO, a ideia é trazer essa imagem mental, para dentro da estrutura de uma classe.

Imaginando e definindo cada atributo e cada método necessário para fazer com que o objeto que será construído a partir daquela classe, represente o mais fielmente possível um objeto do mundo real.

Exercício

Exercício

1 - Fazer o processo de abstração, e criar uma classe pessoa. Tem que ter pelo menos 10 atributos e 5 métodos.

Os métodos podem simplesmente imprimir na tela o que a pessoa está fazendo com um Console.WriteLine().