Aula 18

Prof: Henrique Augusto Maltauro

Desenvolvendo Algoritmos

Programação Orientada a Objetos (POO)

- Quatro pilares da POO

Principais fundamentos da POO.

- Abstração ✓
- Herança ✓
- Encapsulamento
- Polimorfismo

Programação Orientada a Objetos (POO)

- Quatro pilares da POO (encapsulamento)

O encapsulamento é um dos principais fundamentos da POO, que determina um limite de acesso aos atributos e/ou métodos de uma classe.

Ou seja, é determinado quais lugares podem ou não podem acessar aqueles atributos e/ou métodos.

Programação Orientada a Objetos (POO)

- Quatro pilares da POO (encapsulamento)
  - (vamos imaginar um veículo)

Todo veículo tem uma velocidade de movimento, mas essa velocidade não pode ser alterada de forma direta, ela precisa ser alterada de acordo com o quanto o veículo é acelerado ou freado.

Programação Orientada a Objetos (POO)

- Quatro pilares da POO (encapsulamento)
  - (vamos imaginar um veículo)

Dentro de um contexto de POO, o atributo velocidade não pode ser alterado de forma direta, ele só pode ser alterado pelos métodos acelerar e frear.

Ou seja, o atributo velocidade não pode ser alterado fora da classe veículo.

Programação Orientada a Objetos (POO)

- Quatro pilares da POO (encapsulamento)
  - (vamos imaginar uma conta bancária)

Toda conta bancária possui um saldo, porém esse saldo não pode ser alterado de forma direta, ele só pode ser alterado de acordo com a entrada e saída de dinheiro da conta.

Programação Orientada a Objetos (POO)

- Quatro pilares da POO (encapsulamento)
  - (vamos imaginar uma conta bancária)

Dentro de um contexto de POO, o atributo saldo não pode ser alterado de forma direta, ele só pode ser alterado pelos métodos depositar e sacar.

Ou seja, o atributo saldo não pode ser alterado fora da classe conta bancária.

Programação Orientada a Objetos (POO)

- Quatro pilares da POO (encapsulamento)
  - (vamos imaginar um liquidificador)

Eu como pessoa, eu preciso ter acesso aos botões que fazem o liquidificador funcionar, mas eu não preciso ter um acesso direto ao motor dele, ou saber como funciona o motor dele para que eu possa utilizá-lo.

Programação Orientada a Objetos (POO)

- Quatro pilares da POO (encapsulamento)
  - (vamos imaginar um liquidificador)

Dentro de um contexto de POO, eu só preciso ter acesso aos métodos de ligar e desligar o liquidificador, e eu não preciso ter acesso aos métodos que executam todo o resto das funcionalidades dele.

Programação Orientada a Objetos (POO)

- Quatro pilares da POO (encapsulamento)

Se nós pensarmos no conceito da palavra encapsular, ela significa colocar em uma cápsula, separar do resto.

E para conseguirmos utilizar o encapsulamento, precisamos compreender e utilizar os modificadores de acesso.

Programação Orientada a Objetos (POO)

- Modificadores de acesso

Os modificadores de acesso são palavras-chave usadas para especificar a acessibilidade de um atributo ou método.

Ou seja, a ideia é definir quais classes podem ou não podem ter acesso a aquele atributo ou método.

Programação Orientada a Objetos (POO)

- C#: Modificadores de acesso (public)

O modificador public determina que aqueles atributos ou métodos daquela classe, podem ser acessados por todas as classes, eles podem ser utilizados sem restrições.

Programação Orientada a Objetos (POO)

- C#: Modificadores de acesso (public)

```
public string Nome;
```

Programação Orientada a Objetos (POO)

- C#: Modificadores de acesso (private)

O modificador private determina que aqueles atributos ou métodos daquela classe, só podem ser acessados por ela mesma.

Programação Orientada a Objetos (POO)

- C#: Modificadores de acesso (private)

```
private string Nome;
```

Programação Orientada a Objetos (POO)

- C#: Modificadores de acesso (protected)

O modificador protected determina que aqueles atributos ou métodos daquela classe, só podem ser acessados por ela mesma e por suas classes filhas.

Ou seja, somente podem ser acessados por ela mesma e pelas classes que herdam aquela classe.

Programação Orientada a Objetos (POO)

- C#: Modificadores de acesso (protected)

```
protected string Nome;
```

Programação Orientada a Objetos (POO)

- C#: Modificadores de acesso (static)

O modificador static determina que aqueles atributos ou métodos daquela classe, não precisam de uma instância da classe para serem utilizados.

Ou seja, a classe não precisa ser construída em um objeto para utilizar aqueles atributos ou métodos.

Programação Orientada a Objetos (POO)

- C#: Modificadores de acesso (static)

```
public static string Nome;
private static string Nome;
protected static string Nome;
```

Exercício

Exercício

1 - Criar a classe Veiculo. Ela deve possuir o atributo Velocidade, que deve ser do tipo inteiro, inicialmente com o valor de zero e os métodos Acelerar e Frear.

O atributo Velocidade deve ser protegido e os métodos Acelerar e Frear públicos. Sempre que o método Acelerar for executado, deve aumentar a Velocidade em um. Sempre que o método Frear for executado, deve diminuir a Velocidade em um.

Criar um método que seja capaz de imprimir no console a Velocidade.

Exercício

2 - Criar a classe ContaBancaria. Ela deve possuir os atributos NumeroDaConta do tipo inteiro, NumeroDaAgencia do tipo inteiro, Saldo do tipo decimal, todos eles protegidos, e deve possuir os métodos Depositar e Sacar, ambos públicos, os atributos NumeroDaConta e NumeroDaAgencia devem ser definidos com parâmetros no método construtor.

Os métodos Depositar e Sacar devem receber um valor do tipo decimal, que irá determinar o quanto o atributo Saldo deverá aumentar ou diminuir.

Criar um método para cada um dos atributos serem impressos no console.