

Programação Orientada a Objetos

Aula 6

- Revisão
- SOLID
- Exercícios
- Continuação projeto Corrida de cachorros

Programação Orientada a Objetos

- SOLID
 - Boas práticas de programação
 - Código limpo
 - Manutenção
 - Facil de testar
 - Escalabilidade

Programação Orientada a Objetos

- “S” OLID “SRP”
- Princípio da responsabilidade unica
- Uma classe deve ter somente uma razão para existir e alterar, com isso o que quero dizer é que ele vai ter somente uma responsabilidade

- Exemplo que não segue o solid

- ```
class Empregado{
 public void SalvarNoBanco(){};
 public void GerarRelatorio(){}/
}
```

- Exemplo que segue o solid

- ```
class Empregado{  
    public void SalvarNoBanco(){};  
}  
  
class GeradorDeRelatorio{  
    public void GerarRelatorio(){};  
}
```

Programação Orientada a Objetos

- SOLID “SRP” Exercício do Princípio da Responsabilidade Única:
- Temos um programa que é responsável para armazenar informações de um produto e calcular o preço do produto baseado em diferentes descontos. Quero que seja modificado o código abaixo para aplicar o SRP

```
public class Produto
{
    public string Nome{ get; set; }
    public decimal Preco{ get; set; }
    public decimal Desconto{ get; set; }
    public void SalvarNoBanco()
    {
        // Lógica para salvar o produto no banco de dados
    }
}
```

```
public decimal CalcularPrecoFinal()
{
    // Lógica para calcular o preço final com desconto
    decimal precoComDesconto= Preco - (Preco * Desconto/ 100);
    return precoComDesconto;
}
```

Programação Orientada a Objetos

- SOLID “SRP” Exercício do Princípio da Responsabilidade Única:
- Agora que aplicou o SRP no nosso problema, temos uma nova solicitação da loja que usa o sistema, ela quer aplicar uma promoção de natal para os produtos que contém no seu nome a palavra “natal”, para estes produtos será aplicado um desconto de 20% os outros produtos terão um desconto de 15%
- A loja só tinha opção de pagamento em dinheiro, mas agora ela implementou o meio de pagamento por PIX e Cartão de crédito, com isso foi solicitado para que quando for pago com PIX tenha um desconto a mais de 10% e quando o pagamento for em cartão será aumentado o valor do produto em 5%, com isso altere o software para calcular o preço final com estas novas regras

Programação Orientada a Objetos

- SOLID “SRP” Exercício do Princípio da Responsabilidade Única:
- Agora a loja quer também salvar os preços e produtos em um banco de dados, para isso vamos usar o banco SQLITE, vai ajudar a lembrar os conceitos de banco de dados vistos. Para poder fazer isso segue um exemplo de código que salva em um banco de dados SQLite.
 - Abra o Visual Studio e selecione o seu projeto.
 - Vá para Tools > NuGet Package Manager > Manage NuGet Packages for Solution.
 - Na guia Browse, procure por System.Data.SQLite.
 - Selecione o pacote System.Data.SQLite e clique em Install para adicionar o pacote ao seu projeto.

Programação Orientada a Objetos

```
private const string ConnectionString = "Data Source=Products.db;Version=3;";

public void SaveToDatabase(Product product)
{
    using (var connection = new SQLiteConnection(ConnectionString))
    {
        connection.Open();

        using (var command = connection.CreateCommand())
        {
            command.CommandText = "INSERT INTO Products (Name, Price, DiscountPercentage) VALUES (@Name, @Price, @DiscountPercentage)";
            command.Parameters.AddWithValue("@Name", product.Name);
            command.Parameters.AddWithValue("@Price", product.Price);
            command.Parameters.AddWithValue("@DiscountPercentage", product.DiscountPercentage);

            command.ExecuteNonQuery();
        }
    }
}
```

Programação Orientada a Objetos

- Agora queremos mostrar os produtos que tem na loja, para isso implemente

```
public void MostrarProdutos()
{
    using (var connection = new SQLiteConnection(ConnectionString))
    {
        connection.Open();
        using (var command = connection.CreateCommand())
        {
            command.CommandText = "SELECT * FROM Products";
            using (var reader = command.ExecuteReader())
            {
                while (reader.Read())
                {
                    Console.WriteLine($"Product: {reader["Name"]}, Price: {reader["Price"]}, Discount: {reader["DiscountPercentage"]}");
                }
            }
        }
    }
}
```


Programação Orientada a Objetos

- Baseado no código anterior, altere crie um novo método, para Trazer todos os Produtos
- `public Produtos[] TragaTodosOsProdutos()`, e use esta implementação no `MostrarProdutos()`



Serviço Nacional de Aprendizagem Industrial

PELO FUTURO DO TRABALHO

0800 048 1212     **sc.senai.br**

Rodovia Admar Gonzaga, 2765 - Itacorubi - 88034-001 - Florianópolis, SC