A Programação Orientada a Objetos (POO) é um paradigma de programação que organiza o software a partir de entidades chamadas objetos, que representam elementos do mundo real ou conceitual. Esses objetos são definidos por meio de classes, que funcionam como moldes.

Segundo Forbellone e Eberspächer (2014), a POO visa tornar o código mais reutilizável, modular e de fácil manutenção, sendo amplamente adotada em linguagens como C# e Java.

1. Classe

A classe é a estrutura que define os atributos (características) e métodos (comportamentos) de um objeto. É a base para a criação de objetos, funcionando como um modelo.

Fonte: Forbellone, A. L. V., & Eberspächer, H. F. – Programação Orientada a Objetos com C#.

2. Objeto

O objeto é a instância concreta de uma classe. Ele possui estados e comportamentos definidos pela classe de origem. Em outras palavras, é quando aquele modelo (classe) é colocado em uso no programa.

Fonte: Microsoft Learn – Introdução à POO em C#.

3. Encapsulamento

Encapsulamento significa proteger os dados de um objeto, restringindo o acesso direto a seus atributos e permitindo a interação por meio de métodos. Isso melhora a segurança e integridade dos dados.

Fonte: Curso Alura – Programação Orientada a Objetos com C#.

4. Herança

A herança é um mecanismo que permite que uma classe herde características e comportamentos de outra. Com isso, é possível reutilizar código e criar hierarquias entre objetos.

Fonte: Microsoft Learn – POO em C#.

5. Polimorfismo

Polimorfismo refere-se à capacidade de objetos diferentes responderem de formas distintas à mesma ação. Ele permite modificar ou estender comportamentos sem alterar o código da superclasse.

Fonte: Curso DIO – Formação Fundamentos do .NET com C#.

6. Abstração

A abstração consiste em esconder detalhes internos e mostrar apenas as informações relevantes. Na prática, ela permite que o desenvolvedor trabalhe com interfaces e conceitos genéricos, focando no “o que” e não no “como”.

Fonte: Forbellone, A. L. V., & Eberspächer, H. F. – Programação Orientada a Objetos com C#.

7. Associação, Agregação e Composição

Esses três conceitos representam formas de relacionamento entre classes:

• Associação: vínculo genérico entre dois objetos.

• Agregação: um objeto faz parte de outro, mas pode existir independentemente.

• Composição: um objeto é parte essencial do outro e não existe separadamente.

Esses conceitos são úteis para modelar a estrutura e o comportamento do sistema de forma mais próxima da realidade.

Fonte: Curso Alura – POO com C#.

Referências (todas em português)

1. Forbellone, A. L. V., & Eberspächer, H. F.

Programação Orientada a Objetos com C#. Pearson Education do Brasil, 2014.

2. Microsoft Learn – Introdução à POO em C#

https://learn.microsoft.com/pt-br/dotnet/csharp/fundamentals/tutorials/oop

3. Alura – Curso de Programação Orientada a Objetos com C#

https://www.alura.com.br/curso-online-programacao-orientada-a-objetos-csharp

4. DIO – Formação Fundamentos do .NET com C#

https://www.dio.me/bootcamp/fundamentos-dotnet