

1) Os princípios fundamentais da Programação Orientada a objeto são:

Encapsulamento: Ocultar e restringir o acesso direto aos dados.

Herança: Permite que uma classe(subclasse) utilize métodos de outra classe maior(superclasse), promovendo reutilização de código.

Polimorfismo: Permite que um objeto possa ser referenciado de múltiplas formas, permitindo que métodos se comportam diferente dependendo do objeto passado, facilitando utilização e reduzindo acoplamento

Abstração: Foco em aspectos essenciais de um objeto, sem focar em como foi implementado um método.

Esses princípios são importantes para modularidade do código, reutilização de código, manutenção, escalabilidade, segurança e corresponde a uma modelagem do mundo real.

2) A linguagem C é estruturada, ela se baseia em funções e estruturas de controle, tendo um foco em ações diretamente e tendo uma sequência de execução. Os dados e as funções são separados(as funções manipulam dados externos).

Já a linguagem Orientada a Objetos baseia-se em objetos que são combinações de dados com métodos. O foco está em entidades semelhantes a do mundo real.

As principais vantagens são, reuso de código Modularidade, Facilidade de manutenção e herança e polimorfismo.

3) Em POO a classe é o modelo que define os atributos e métodos que os objetos terão, já o objeto é uma instância dessa classe, ou seja, uma cópia concreta com valores próprios.

Exemplo: A classe Carro define características como modelo e ano, enquanto meuCarro = new Carro() é um objeto específico, por exemplo, um carro "Civic 2020".

10)a) O acoplamento é a extrema dependência de um módulo em outro módulo, sendo a medida de quanto uma parte do código conhece ou depende de detalhes internos da outra. É importante reduzir o acoplamento para que o sistema fique otimizado para futuras manutenções e escalabilidade, uma vez que com acoplamento alto, os módulos não são independentes, sendo necessário mudar tudo ao modificar algo.

c) A minha implementação caracteriza os atributos como privados para não ter acessos indevidos, adiciona getters and setters para melhor manipulação. Adicionar livro agora não depende diretamente de Livro, pois apenas recebe um objeto livro ao invés de criar um dentro do método, o método listar livros não acessa diretamente os atributos de livro, sendo acessado agora por um toString. A classe livro agora é autônoma, biblioteca gerencia apenas uma coleção de livros.