

Herança: As classes Carro e Moto herdam atributos e métodos da classe Aluguel. Isso promove a reutilização de código

Classes Abstratas: A classe abstrata Aluguel serve como uma base para as classes concretas Carro e Moto. Ela contém atributos e métodos comuns a ambas as subclasses, permitindo a reutilização de código e fornecendo uma estrutura comum para suas implementações específicas.

Classes Concretas: São as classes Carro e Moto que utilizamos para herdar uma classe abstrata implementando as funcionalidades definidas pela classe mãe

Métodos abstratos: Define um conjunto de operações que qualquer implementação dessa interface deve fornecer.

Método Concreto: um exemplo de método concreto no nosso código é os métodos do CRUD, pq pegamos a assinatura da interface e implementamos ele na classe que herda, e definimos o método lá.

Polimorfismo: O polimorfismo é aplicado ao utilizar métodos da interface Aluguel em outras classes, como Carro e Moto. Isso significa que essas classes podem implementar esses métodos de maneiras diferentes, adequando-se às suas próprias necessidades, mas mantendo a mesma assinatura definida na interface.

Interface: A interface Aluguel define um conjunto de operações que podem ser realizadas em relação aos aluguéis de carros e motos no código

Encapsulamento: Os atributos da classe Aluguel são declarados como privados, o que significa que eles não podem ser acessados diretamente de fora da classe. Para permitir o acesso a esses atributos, métodos getters e setters são fornecidos, garantindo que o acesso e a modificação dos dados ocorram de maneira controlada.