

1. Classe

Podemos entender a Classe como um modelo. Modelo esse que pode ser usado para criar objetos do mundo real para dentro do mundo digital. Por ser um modelo, cada classe tem seus atributos que podemos entender como características inerentes àquela classe e seus métodos que seriam as funções/ações que aquele objeto realiza ou deve realizar.

Por exemplo, se temos a classe Carro, algumas características (ou atributos) do carro mais superficiais seriam: motor, pneus, portas, bancos, volante, dentre outros. Agora, analisando as funções de um carro (ou os seus métodos) teríamos algo como acelerar, frear, ligar faróis, etc.

A Classe é um modelo, que podemos armazenar e acessar valores dentro de seus atributos e realizar operações usando os seus métodos.

2. Objeto.

Um objeto é uma instância de memória de um tipo específico. O tipo específico diz respeito à classe a qual aquele objeto foi criado, podendo então armazenar e acessar dados e realizar operações daquela determinada classe. Por exemplo, uma Classe Garrafa teria como atributo volume e como método armazenarConteúdo. Assim, seria possível criar o objeto garrafaPlastica e então, acessar os atributos (volume) e realizar operações (encher a garrafa).

3. Método e Atributo.

Atributos são características de uma objeto, ou ainda, as variáveis de uma Classe. Métodos são as operações que podem ser realizadas de uma determinada Classe. São declaradas dentro das classes e podem receber atributos como parâmetros e retornar a saída da operação realizada.

4. Encapsulamento.

Encapsulamento é o ato de encapsular os atributos de uma Classe, ou seja, tornar esses atributos visíveis apenas para o escopo da Classe a que foram criados.

5. Visibilidade de atributos e métodos (public , private e protected).

A visibilidade de atributos e métodos determina o escopo de acesso de tais atributos e métodos. A visibilidade de público indica que aquele atributo/método pode ser acessado por qualquer outra entidade dentro do código da aplicação. A visibilidade protegido, indica que aquele atributo/método só pode ser acessado por entidades que estão no mesmo pacote. A visibilidade privado, indica que aquele atributo/método só pode ser acessado dentro da classe em que foi criado.

6. Herança.

Herança é a capacidade de uma determinada classe (subclasse) herdar métodos e atributos de outra classe (superclasse).

7. Polimorfismo

Polimorfismo é a capacidade de uma subclasse declarar e executar métodos de mesmo nome de sua superclasse porém, execute instruções diferentes do que é atribuído na superclasse.

8. Classes Abstratas.

Classes abstratas são classes em que não é possível criar objetos a partir das mesmas, sendo possível apenas herdar seus atributos e métodos.

9. Construtores.

Construtor é um método que é sempre executado inicialmente sempre que um objeto novo de uma classe é criado. Como ele é a primeira coisa que é executado quando é criado o objeto, pode-se atribuir valores para os atributos e determinar certas coisas sempre que o objeto é criado

10. Get e Set

Get e Set são métodos que usamos para acessar e atribuir o conteúdo de atributos privados da mesma classe. A visibilidade privado torna o escopo dos atributos restritos somente àquela classe em que foram definidos, por isso, usa-se os métodos get e set para acessar e atribuir determinados valores.

11. Sobrecarga de Métodos.

Sobrecarga de métodos é um tipo de polimorfismo, ou seja, um método de uma subclasse que tem nome idêntico a um método que herda de sua superclasse, porém, recebe parâmetros diferentes dos que o método da superclasse recebe e pode retornar tipos diferentes do que o método da superclasse retorna.

12. Sobrescrita de Métodos.

Sobrescrita de métodos é um tipo de polimorfismo, ou seja, um método de uma subclasse que tem nome idêntico a um método que herda de sua superclasse, porém, possui instruções diferentes daquelas que há no método da superclasse.

13. Palavras reservadas (super, this e final)

A palavra reservada super é usada quando temos uma subclasse querendo acessar o construtor da superclasse ou chamar determinado método da superclasse.

A palavra reservada final é usada quando não queremos que haja modificação em determinada entidade. Para atributos, garante que aquele atributo só terá uma única atribuição (construtor ou por definição) e não será mais modificado. Para métodos, garante que aquele método não poderá ser sobrescrito ou sobrecarregado, ou seja, não poderá ser modificado em subclasses. Para classe, garante que aquela classe não pode ser superclasse, ou seja, nenhuma classe pode herdar nada dela e assim não poderá haver modificações em seus métodos ou atributos.

A palavra reservada this é usada quando queremos referenciar um atributo ou método próprio daquela classe.