1 - O paradigma da POO é um modelo de programação focado em classes e objetos e em como eles se organizam, considerando quais métodos (funções/comportamnetos) e atributos (dados/características) possuem . A poo possui diversas vantagens, como maior flexibilidade, manutenção, organização, modularidade (é dividida em partes, as classes), e o principal, a reutilização de código, isso tudo devido a suas características: o polimorfismo, a herança, encapsulamento, etc.

Ela se diferencia da programação estruturada, onde o foco maior está no desenvolvimento do melhor algoritmo pra resolver determinado problema, separada em dados e funções, e com mais dificuldade em manter grandes códigos, pois pequenas mudanças podem acarretar em erros mais adiante no código. Dessa forma, a orientada é focada no projeto das classes, e assim, dos objetos, que são instâncias das classes, que combinam dados e métodos, ela é muito mais flexível, fácil de manter, e mais reutilizável devido a suas características já citadas, polimorfismo, herança, etc. E permite modelar o mundo real de forma mais direta, com classes representando entidades reais e suas características.

2 – Uma classe é como um modelo que define como os objetos serão, seus atributos (dados) e quais métodos (comportamentos) possuirão ao ser criados. Ela tem a finalidade de agrupar os dados e métodos, e assim, ser a base para a POO fornecendo estrutura para suas características: herança, encapsulamento, polimorfismo, etc. Um objeto nada mais é do que uma instância de uma classe, então está ligado direta e profundamente com a classe (classe de origem dele).

-------------------------------------------------------------------------------------

3 – Um objeto em POO é uma instância de uma classe, ele é uma referência para usarl determinados dados e métodos de uma classe, e assim como no mundo real, o que realmente manipulamos são objetos, e não seus atributos ou comportamentos, oq seria algo bem abstrato e sem sentido de se pensar, então precisamos de uma entidade de referência para usarmos suas características, como na realidade, um objeto.

4 – Atributos são os dados que a instância irá possuir, como cor, tipo, modelo, ano de fabricação, tempo de uso, etc. Já os métodos são as ações e comportamentos que um objeto poderá fazer, são iguais a funções, como por exemplo os métodos: acelerar(), freiar(), abastecer(), etc.

5 – Esses modificadores controlam quem pode acessar os atributos e métodos e são essenciais para o encapsulamento em POO. O public permite que qualquer um possa acessar aquele método ou atributo, dentro e fora da classe, ou por outra classe. O private limita o acesso a somente a própria classe, é o modificador mais restrito. E o protected permite o acesso somente a própria classe e a classes herdadas a essa original, como “classes filhas” a essa.

6 – O tipo static serve para criar métodos que podem ser acessados sem a necessidade de um objeto, e atributos que possuem sempre o mesmo valor independente de quantos objetos existirem, ou seja, na memória só existirá uma variável static para todos os objetos.

7 – Um construtor é um método sem tipo de retorno e com o mesmo nome da classe que serve para criar seus respectivos objetos.

8 – Sobrecarga de métodos é quando existe dois ou mais métodos com o mesmo nome, independente do tipo que retornam. A única regra é que apesar do mesmo nome, os métodos precisam ter parâmetros diferentes, para poderem ser diferenciados.

9 – Quando o atributo ou método não será utilizado fora da classe, ou seja, o usuário (programa principal) não precisa ter acesso a ele. Pois se o usuário não precisa desse atributo, o código ficaria desorganizado, com informações a mais do que o necessário e sem funcionalidade para o programa principal, como por exemplo um endereço de IP ou o ID de um produto, algo necessário dentro da classe, mas sem uso para um usuário.

10 – Uma classe pode sim ser considerada um tipo de dado. Pois quando declaramos um objeto, uma instância dentro do programa, declaramos seu tipo como sendo o nome da classe, ou seja, esse objeto terá as características, os atributos, os métodos dessa classe, por isso, será do tipo dessa classe.

13 – O conceito de objeto permite o agrupamento de dados de forma bem simples em um programa, e com isso, facilita a manutenção, reutilização, e uma boa divisão do código, a partir das classes, ou seja, a modularidade (divisão do código em módulos independentes) é praticamente a definição de classes em POO, que são os moldes para instanciar um objeto.

20 - A POO permite a representação de entidades do mundo real em forma de código de maneira mais direta, por conta dos conceitos de classe e objeto. Esses conceitos facilitam a abstração da entidade no mundo real para o código, pois basta se perguntar quais atributos (características) são necessários, e quais métodos (comportamentos) ela precisaria fazer para aquele determinado programa, criar a classe, e com isso, criar um objeto, uma instância daquela classe, que possui todos os dados já escritos, e assim como na realidade, a entidade funcionaria como o programa gostaria.