**EXERCÍCIOS TEORICOS DO 1 AO 8 LISTA 2 P00**

1a) A programação orientada a objetos (POO) simplifica programas complexos ao dividir-los em objetos. Esses objetos permitem organizar o código de forma mais claro, promovendo reutilização e flexibilidade na construção de sistemas modulares.

uma exemplificação de uma POO seria: a criação de uma classe (carro). Nessa classe definiríamos os atributos como marca, modelo, cor e ano. Após feito, nos definiríamos os métodos para o carro, como, ligar, desligar, acelerar. Depois de criada a classe podemos criar instancias especificas de carros com diferentes marcas

modelos e abis, cada um capaz de executar os métodos definidos na classe.

1b) Na POO classe é uma estrutura que representa um conjunto de objetos que possuem características comuns. Uma classe define os métodos e atributos que os objetos desse tipo possuem. Exemplificando: nós poderíamos criar uma classe carro

que englobaria todos os carros com base nas definições e comportamentos especificados pela classe.

1c) Uma instância refere-se a um objeto especifico criado a partir de uma classe. Quando criamos um objeto utilizando os atributos e métodos definidos em uma classe, estamos instânciado essa classe. Exemplificando: quando estamos montando o código

a instancia seria quando determinamos se a informação é pública, privada ou restringida.

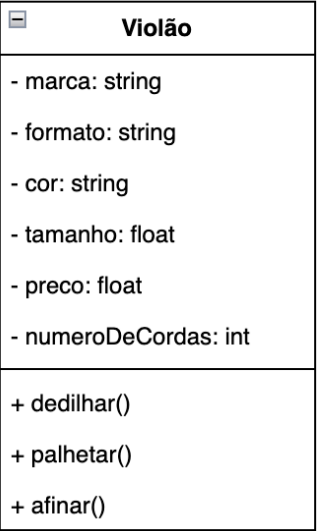
1d) Na POO, um objeto é um elemento computacional que representa, no domínio da solução, alguma entidade desse domínio de interesse deste problema sob análise, podendo ser abstrata ou concreta. Objetos similares são agrupados em classes.

2a) Classes são frequentemente mais usadas para modelar conceitos mais complexos e abstratos, como objetos reais e entidade de negócios. Nas classes, os membros são privados por padrão, o que significa que você precisa especificar modificadores de acesso

como( public, private ou protect) para controlar o acesso deles, especificadores que tornam os dados mais seguros se comparados com o struct . Structs são usados para representar tipos de dados simples, como coordenadas por exemplo, não contendo nenhum

tipo de modificadores de acesso como nas classes.

3) Atributos são as características que iremos colocar para os objetos ligados a classe feita. Já os métodos estão relacionados ao o que é possível fazer com esses objetos, quase sempre ligados a verbos.



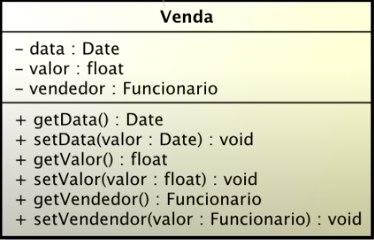
Nesse exemplo os atributos são os itens da segunda caixa, que estão com um - antes deles, marca, formato, cor, tamanho e etc. Já os métodos são as ações que são possíveis de serem feitas com esse objeto são precedidos por um símbolo de soma +, dedilhar, palhetar e afinar.

4) A melhor opção que melhor define as três características elementares de um objeto na POO é a letra: C – Estado, Comportamento e Identidade.

5) A herança é um conceito que permite que uma classe (derivada ou subclasse) herde os atributos e métodos de outra classe (classe base ou classe mãe). O polimorfismo, permite que um mesmo método seja invocado de diferentes formas, o comportamento desse método pode variar de acordo com o tipo de objeto que você invoca.

6) O encapsulamento permite que você abstraia a complexidade interna de uma classe ou objeto. Em vez de expor todos os detalhes de implementação, você pode fornecer uma interface simples e de alto nível que esconda os detalhes complicados e permita que os usuários da classe se concentrem apenas no que for relevante. Para isso usamos o private, para escondermos alguns dados de implementação, usamos o public para deixar os dados visíveis para o usuário e o restrict para restringir os dados para somente alguns objetos.

7) Os getters e setters ajudam a manter os campos de uma classe privada, ocultando sua aplicação dos usuários externos da classe. Eles permitem que você mantenha o controle sobre como os atributos de uma classe são acessados e modificados. Sendo útil para garantir que os dados da classe sejam usados de maneira segura e consistente.



8) Herança é um mecanismo para reutilização de classes. É um principio de orientação a objetos, que permite que classes compartilhem atributos e métodos. Ela é usada na intenção de reaproveitar o código ou comportamento generalizado ou especializar operações ou atributos.