PROGRAMAÇÃO ORIENTADA A OBJETOS. Pt 2

POO

*Todos os conceitos fazem parte do livro: Conceitos de Computação Com o Essencial de JAVA. Cay Horstmann. 2005

Encapsulamento

É um conceito da Orientação a Objetos para restringir acesso aos dados do nosso objeto.

Isso é uma forma de garantir mais segurança para o objeto, sendo que só o código dentro da classe poderá modificar suas propriedades.

Modificadores de acesso

Public: acesso livre para modificar a classe tanto dentro, quanto fora dela.

Private: só pode modificar a classe dentro dela mesmo.

Protected: além de ser acessível dentro da própria classe, também é acessível para classes que herdam. (Veremos mais na parte de herança)

Modificadores de acesso: static e final

Static: É usado para classes que não precisam ser instanciadas. Nosso maior exemplo é o System, porém temos outras classes, como Math.

Final: É uma constante. Depois de declarado, o valor dessa variável nunca poderá ser alterado

Enumerações

É um tipo de dado especial que permite uma variável seja um conjunto de constantes pré definidas.

Por serem constantes, os nomes dos campos de um tipo enum estão em letras maiúsculas.

Associação

É um vínculo que ocorre entre classes (normalmente de muitos para muitos), porém, essas classes não dependem uma da outra para existir.

Exemplo: aluno e professor. O aluno pode ter vários professores ou nenhum. E vice-versa.

Agregação

Muito parecido com associação, mas dependem de uma exclusividade com determinado objeto.

Exemplo: Uma pessoa pode ter vários carros, mas o carro só pode ter um proprietário.

Composição

É uma relação que possui dependência de outro objeto. Se o objeto principal for destruído, os objetos que compõem não podem existir mais.

Exemplo: Um pedido e os itens do pedido. Se o pedido for excluído, os ítens do pedido também deverão.

Composição

É uma relação que possui dependência de outro objeto. Se o objeto principal for destruído, os objetos que compõem não podem existir mais.

Exemplo: Um pedido e os itens do pedido. Se o pedido for excluído, os ítens do pedido também deverão.

Exercício 1

Crie uma classe Aluno com os campos nome, matrícula e notas[3].

Essa classe deve ser encapsulada e no momento de adicionar uma nova nota, ela deve verificar se o somatório será maior que 100, caso for. Deixe uma mensagem que não foi possível e peça para adicionar novamente uma nota.

Exercício 2

Crie uma classe Pedido com os campos Itens[10], status e valorTotal[3].

Essa classe deverá ter os métodos para adicionar itens, que receberá como parâmetro o item e o valor.

Também deverá ter um método para alterar o status do pedido.

Exercício 3

Crie uma classe Calculadora com os métodos das operações matemáticas estáticos.

Soma, Subtração, Multiplicação e Divisão.

Use essa classe sem instanciar ela.