

Lista 4: Programação Orientação a Objetos

- 1) A fim de representar empregados em uma firma, crie uma classe chamada Empregado. esta classe deverá possuir as três informações a seguir como atributos:
 - a. Um primeiro nome,
 - b. Um sobrenome,
 - c. Um salário mensal.

Sua classe deve ter um construtor que inicializa os três atributos. Forneça um método set e get para cada atributo. Se o salário mensal não for positivo, configure-o como 0.0. Crie duas instâncias da classe e exiba o salário anual de cada instância. Então dê a cada empregado um aumento de 10% e exiba novamente o salário anual de cada empregado.

- 2) Crie uma classe chamada Fatura que possa ser utilizada por uma loja de suprimentos de informática para representar uma fatura de um item vendido na loja. Uma fatura deve incluir as seguintes informações como atributos:
 - a. O número do item faturado
 - b. A descrição do item
 - c. A quantidade comprada do item
 - d. O preço unitário do item.

Sua classe deve ter um construtor que inicialize os quatro atributos. Se a quantidade não for positiva, ela deve ser configurada como 0. Se o preço por item não for positivo ele deve ser configurado como 0.0. Forneça um método set e um método get para cada variável de instância. Além disso, forneça um método chamado getValorFatura que calcula o valor da fatura (isso é, multiplicar a quantidade pelo preço por item) e depois retorna o valor como um double.

- 3) Crie um programa que tenha uma classe produto constituída de nome, preço e quantidade. Implemente um método chamado "comprarProduto". Este método quando chamado deverá modificar uma variável do produto indicando que o produto foi comprado. Para cada produto informado escreva o nome do produto comprado e o valor total a ser pago, considerando que são oferecidos descontos segundo o valor do produto conforme a tabela abaixo:
 - a. Até R\$ 100: sem desconto
 - b. De R\$ 100 a R\$ 200: 10% de desconto
 - c. De R\$ 200 a R\$ 500: 20% de desconto
 - d. Acima de R\$500: 25% de desconto

- 4) Um contato é constituído de código, nome da pessoa, endereço, e-mail, telefone e observação.
- a. Crie 4 construtores e atribua o valores por meio dos construtores. Os construtores devem receber os seguintes parâmetros:
 - I. Código.
 - II. Código e nome.
 - III. Código, nome e e-mail.
 - IV. Telefone.
 - b. Crie pelo menos 5 contatos.
 - c. O atributo e-mail não poderá ser modificado fora da classe contato, mas poderá ser acessado por outras classes.
 - d. Faça um método para imprimir o código, nome, endereço, e-mail, telefone e observação dos contatos cadastrados.
 - e. Faça um validador do código do contato. Um código válido é maior igual a 1000 e menor igual a 9999. Não permita que seja cadastrado um código inválido para nenhum contato.
 - f. Faça um validador para o telefone. O telefone sempre deverá ter 8 dígitos. Não permita que seja cadastrado um telefone inválido para nenhum contato.
 - g. Crie um método que possa duplicar um contato. Desta forma, um novo contato terá os mesmos dados do contato que for requerido a duplicação.
 - h. Faça um método para verificar se o contato está totalmente preenchido. Ou seja, tenha valores diferentes do padrão para: código, nome, endereço, e-mail, telefone e observação.
- 5) Escreva um programa completo para jogar o jogo da velha. Para tanto crie uma classe Jogo da Velha:
- a. A classe deve conter como dados privados um array bidimensional 3x3 para representar a grade do jogo
 - b. Utilize uma variável do tipo String para representar as possibilidades de ocupação de uma casa na grade (vazia, jogador 1 ou jogador 2)
 - c. O construtor deve inicializar a grade como vazia
 - d. Forneça um método para exibir a grade
 - e. Permitir dois jogadores humanos
 - f. Forneça um método para jogar o jogo; todo movimento deve ocorrer em uma casa vazia; depois de cada movimento, determine se houve uma derrota ou um empate.