NEXT – JAVA e POO

Lista de Exercícios da Semana 01

1- **JAVA "Estruturado"**: Escreva um programa que leia um número N inteiro positivo do teclado e que escreva no console os N primeiros números da sequência de Fibonacci. Se o número digitado for negativo, o programa deve mostrar no console a mensagem "Número inválido".
2- **JAVA "Estruturado"**: Escreva um programa que leia do teclado duas notas de um aluno e que implemente a lógica dada abaixo, a fim de imprimir no console a média final do aluno e o status de aprovação.
    a. Se a média das duas notas for menor do que 4.0, o aluno está REPROVADO DIRETO, e a média final é a média das duas notas.
    b. Se a média das duas notas for maior ou igual a 7.0, o aluno está APROVADO DIRETO, e a média final é a média das duas notas.
    c. Se a média das duas notas, aqui denominada de média parcial, for menor que 7.0 e maior ou igual a 4.0, o programa deve ler do teclado a nota da final. Neste caso, o programa deve calcular a média final, que será igual à (média parcial + nota da final)/2.
        i. Se a média final for maior ou igual a 5.0. o aluno está APROVADO NA FINAL.
        ii. Se a média final for menor que 5.0. o aluno está REPROVADO NA FINAL.
3- **Arrays em JAVA:** Escreva um programa que leia 10 números do teclado e que imprima os números lidos em ordem crescente. Use array para armazenar os números lidos e implemente o algoritmo de ordenação do array segundo a lógica dada como referência na classe postada no classroom – Exemplo Ordenação Array.
4- **JAVA "Estruturado"**: Escreva um programa que leia dois números reais do teclado representando base e altura de um retângulo e que imprima no console a área e o perímetro do retângulo em função dos valores lidos.
5- **Classes, objetos, construtores e encapsulamento:** Escreva uma classe pública Retangulo com dois atributos privados base e altura, métodos públicos de acesso a esses atributos, um construtor público que recebe valores de base e de altura e que inicialize os atributos da classe com os valores recebidos, e mais dois métodos de visibilidade default calcularBase e calcuarPerimetro que retornem respectivamente a base e o perímetro, calculados em função dos atributos base e perímetro. Depois, escreva um programa que leia dois números reais do teclado representando base e altura de um retângulo e que, USANDO A CLASSE Retangulo e um OBJETO Retangulo, calcule e imprima no console a área e o perímetro do retângulo em função dos valores lidos. Ver as classes postadas no classroom: Exercício de Referência Triângulo. Compare esta solução OO com a solução estruturada da questão anterior.
6- **Classes, objetos, construtores e encapsulamento:** Escreva uma classe pública Cliente, com dois atributos privados cpf e nome, métodos públicos de acesso a esses atributos, considerando que cpf, uma vez inicializado, não pode ser alterado, e um construtor público que recebe valores para inicializar os atributos da classe.

7- **Classes, objetos, construtores e encapsulamento:** Escreva uma classe pública ContaPoupanca, com três atributos privados saldo, cliente (do tipo Cliente, criado na questão anterior) e número, um inteiro longo. Tal classe deve ter métodos públicos de acesso aos atributos e dois construtores públicos, sendo com valores para inicializar os atributos saldo e número, e outro para inicializar os três atributos da classe. A classe deve ter três métodos, todos eles de visibilidade default, creditar, debitar e render. Os creditar e debitar recebem como parâmetros valores reais que devem ser, respectivamente, adicionados e subtraídos do saldo. O render deve receber como parâmetro uma taxa de juros em percentual, e o saldo deve ser atualizado segundo a fórmula: saldo = saldo * (1 + taxaJuros/100).

8- **Classes, objetos, construtores e encapsulamento:** Escreva um programa que, usando as classes criadas nas questões 6 e 7, crie clientes e contas poupanças com valores arbitrários de atributos, e que realize operações de crédito, de débito e de render, conferindo através de saídas apropriadas no console o correto funcionamento destes métodos.

9- **Encapsulamento:** Analise o código das classes postadas no classroom – Código Modificadores de Acesso, indicando onde há erro de compilação em função da restrição de visibilidade imposta pelos modificadores de acesso aos atributos, métodos, construtores e classes dados. Confira as respostas colocando o código em uma IDE e compilando o mesmo. A IDE deve acusar erros de complicação nas linhas que apresentem problemas de restrição de acesso.