2ª Lista de Exercícios - Programação Orientada a Objetos com Java

Instruções:

- Leia atentamente os requisitos da questão. Todos devem ser cumpridos para que a questão seja considerada correta.
- A lista deve ser feita individualmente.
- Em caso de cópia, as duas (ou mais) listas serão desconsideradas e receberão nota 0.
- A entrega deve ser feita até o prazo estipulado, envios posteriores não serão aceitos.
- A entrega deve ser feita através de algum servidor de git e marcado como entregue no Google classroom.
- Esta lista faz parte do conjunto de avaliações.
- Crie um método main para cada uma das questões para testar as funcionalidades implementadas.
- Procure sempre seguir as melhores práticas de programação.
- Crie um projeto para a lista e envie o mesmo.

Questões:

1- Implemente uma classe chamada Matriz, que pertence a um pacote chamado school.cesar.q1.domain e que possui um array de arrays como atributo que guarda os seus valores. Os valores devem ser números inteiros. Essa classe deve ter 2 construtores, um que recebe um número inteiro e inicializa a matriz com o mesmo número de linhas e colunas e outro que recebe dois inteiros que inicializa a matriz com o primeiro argumento determinando o número de linhas e o segundo o número de colunas. Os valores da Matriz ao ser instanciada, devem ser preenchidos aleatoriamente, entre 0 e 100. Esta classe deve possuir um método que retorna os elementos da diagonal principal como um array de inteiros. Para este método, deve haver uma validação se a matriz é quadrada, caso não seja, uma mensagem de erro deve ser impressa na tela e um array de zeros, que deve ter o tamanho da linha, deve ser retornado.
Implemente uma classe chamada Matrizes, no pacote school.cesar.q1.utils, que possui um método estático chamado multiplicar que recebe duas matrizes como parâmetro, verifica se a multiplicação é possível e caso seja retorna uma nova Matriz como resultado da multiplicação, e caso não seja, retorna null.

2- Implemente uma classe chamada EquacaoSegundoGrau, que pertence ao pacote school.cesar.q2. Esta classe deve possuir um construtor que recebe três argumentos reais chamados a,b e c e associa a 3 atributos da classe de mesmo

nome. Implemente um método chamado getRaizes que retorna os possíveis valores de x (as raízes da equação). Esses argumentos são equivalentes as seguintes posições na equação de segundo grau:

$$a * x^2 + b * x + c = 0$$

Para calcular os valores de x, utilize a fórmula de Bhaskara.
Sobrescreva os método hashcode, equals (que deve comparar se duas equações tem os mesmo valores de a,b,c e raízes) e o método toString que deve imprimir os valores de a,b,c e das raízes da equação.

3- Implemente uma classe chamada SomaDupla que pertence ao pacote school.cesar.q3. Nesta classe implemente um método estático que recebe como parâmetros um ArrayList de números inteiros chamado valores e um outro inteiro chamado alvo. Esse método deve retornar uma String cujo conteúdo é formado pelos índices dos dois primeiros elementos do ArrayList cuja soma é igual ao valor de alvo. Caso esses índices não existam, a String deve retornar com a mensagem: “Índices Indeterminados”.
No seu método main, leia do teclado os valores que compõem o ArrayList de valores até que o usuário insira a palavra *parar*. A seguir leia do teclado o valor do alvo, execute o método implementado e imprima na tela o valor dos índices ou a mensagem de erro. Não é necessário validar caso o input do usuário não seja um número inteiro ou a palavra parar.

4- Implemente uma classe chamada Data que pertence ao pacote school.cesar.q4.domain e que possui os atributos inteiros dia, mês e ano além do atributo dia da semana. O dia da semana deve ser definido através de uma enumeração.
O construtor deve receber como argumentos o dia, o mês, o ano e o dia da semana. Não é necessário validar se a data passada como input de fato corresponde ao dia da semana definido, porém deve ser validado se as datas são possíveis, como por exemplo, se o mês é fevereiro o dia não pode ser maior que 29 (não é necessário validar se ano é bisexto), e para os meses de 30 dias não deve ser aceito 31 ou maior como dia. Os meses devem ser restritos entre 1 e 12. Caso uma data incorreta seja passada, o sistema deve automaticamente ajustar para o último dia válido do mês. Caso o valor do dia passado seja menor ou igual a zero, o dia será setado automaticamente para o dia 1. Caso o mês passado seja incorreto, o mês utilizado deve ser o 12 para argumentos maiores ou iguais a 13 e 1 em números menores ou iguais a zero. Para anos não há restrições.

Implemente um método que recebe outra data como parâmetro e verifica se a data atual é menor, igual ou maior que a data passada como parâmetro. Uma String deve ser retornada.
Implemente uma sobrecarga do método de comparação que recebe um dia da semana como parâmetro e retorna se data atual tem o dia da semana, igual, anterior ou posterior ao dia da semana passada como parâmetro. O primeiro dia da semana é o domingo.

5 - Implemente uma classe chamada Craps que pertence ao pacote school.cesar.q5. Esta classe simula um jogo de mesmo nome, cujas regras são descritas a seguir:

Dois dados de 6 lados são lançados:
Se a soma dos lados for 7 ou 11 no primeiro lançamento o usuário é vencedor.
Se a soma dos lados for 2,3 ou 12 (chamado de craps), no primeiro lançamento, a banca vence e o jogador perde.
Se a soma for 4,5,6,8,9 ou 10, essa soma se torna a pontuação do jogador. Para vencer, o usuário deve rolar novamente os dados até conseguir a mesma pontuação de uma das 3 últimas jogadas.
Caso algum lançamento resulte na soma 7, o jogador perde.

Crie um método main que simula 30 partidas de Craps, uma partida deve acontecer até que o usuário ganhe ou perca. Sempre que houver o lançamento dos dados imprima na tela o resultado da pontuação obtida e se o jogador venceu, perdeu, ou se a partida deve continuar.