UNIFEI	Universidade Federal de Itajubá Instituto de Engenharia de Sistemas e Tecnologias da Informação - IESTI
1º Laboratório	**ECOT12 – Projeto de Software – Prof. Enzo Seraphim**

Siga os passos a seguir para implementar orientado à objetos uma partida de campo minado para um jogador.

1) Crie um pacote chamado br.edu.unifei.ecot12.lab01 e uma classe chamada CampoMinado.

2) Adicione dois atributos privado do tipo matriz de carácteres chamados visual e jogo, ambos com dimensões 10x10. A matriz visual será a matriz exibida na tela a cada rodada. Assim, todos os seus elementos devem ser inicializado com o caractere '?' (interrogação). A matriz jogo contem o gabarito sobre os locais onde estão as bombas e deve ser inicializada com espaço (' ') em todos elementos.

3) Declare mais 2 atributos: um booleano chamado fimJogo que vai determinar se o jogo já acabou; um inteiro chamado desarmes para guardar a quantidade de desarmes.

4) Para fazer leitura e escrita nas variáveis devem haver métodos públicos (gets e sets, respectivamente) para os atributos. No entanto, reflita para quais atributos devem haver permissão de leitura e escrita.

5) No construtor da classe deve-se sortear aleatoriamente 10 elementos na matriz onde serão guardadas as bombas na matriz do jogo. Use o símbolo '✳' para representar a bomba. Para sortear números aleatoriamente use a classe Random de java.util. Garanta que sempre existirão 10 bombas, pois 2 ou mais sorteios podem cair no mesmo lugar. Em seguida, deve-se preencher as células ao redor das bombas com uma indicação da quantidade de bombas que estão por perto. Se o elemento da matriz está vazio (' ') deve-se guardar a contagem de bombas que existem ao redor desse elemento. Não deve aparecer o zero nas posições que não contêm bombas ao redor, mas sim ' ' (espaço em branco).

Importante:
Na tabela de codificação de caracteres (ASCII): '0'=48, '1'=49, '2'=50, '3'=51, '4'=52, '5'=53, '6'=54, '7'=55, '8'=56 e '9'=57.
Para mostrar (ou guardar) o caractere '0' basta somar 48 (ou '0') na contagem de vizinhos.
Cuidado para não acessar uma região de memória que não pertence a matriz. Verifique se o elemento está na borda, e não tente, por exemplo, acessar o elemento (i-1) se i = 0 (não existe elemento -1).

Os vizinhos de um elemento i, j da matriz são:

i-1, j-1	i-1, j	i-1, j+1
i, j-1	i, j	i, j+1
i+1, j-1	i+1, j	i+1, j+1

A seguir um exemplo após a execução da contagem:

	0	1	2	3	4	5	6	7	8	9
0	1	1	1				1	✳	2	1
1	1	✳	2	1			1	2	✳	1
2		3	✳	2				1	1	1
3		2	✳	2						
4		1	1	1	1	1	1			
5					1	✳	1			
6					1	1	2	2	2	1
7							1	✳	✳	1
8	2	2	1				1	2	2	1
9	✳	✳	1							

6) Faça um método chamado desarmar que recebe parâmetro dois inteiros linha e coluna. Essa função devolve verdadeiro a posição ainda não teve desarme e caso contrário falso. Caso o desarme ainda não foi feito deve-se incrementar a variável desarmes. Para desarmar função deve receber na matriz visual o valor da matriz jogo com a posição linha e coluna passada para função. Essa função ainda deve verificar se mudou o estado do fimJogo para verdadeiro quando o valor de desarmes foi igual a 90 ou se o desarme aconteceu em um local que tem bomba.

7) Faça uma classe App com uma função main que realiza várias partidas enquanto o estado do fim jogo é falso. A cada partida deve-se imprimir na tela os valores da matriz "visual", sendo que, cada linha deve conter 10 elementos. Em seguida, o usuário deve informar uma linha e coluna para realizar o desarme.