

Aluno: _____

Construa um jogo no estilo campo minado. O programa deve ter um registro que contenha um número inteiro, que será usado para implementar a lógica do jogo e uma String, cujo conteúdo será apresentado na tela.

Ao iniciar o jogo um array de registros com 10 posições é preenchido automaticamente, de forma que o campo numérico deve ser preenchido com valores aleatórios inteiros entre 0 e 200, e o campo String deve ser preenchido com um caractere qualquer (o mesmo para todas as posições). Após o preenchimento o array deve ser apresentado na tela, mas somente o campo String, sem os números.

O jogador terá 10 chances para indicar posições do array nas quais ele acha que o campo numérico tem números pares. A cada palpite, em caso de acerto, o jogo deve “limpar” os números pares adjacentes ao número da posição indicada por ele, tanto para frente quanto para trás até encontrar um número ímpar – para cada lado. Por exemplo:

Supondo que o array tem esse conteúdo e o usuário chuta a posição 2

Número	23	22	26	66	68	99	15	9	12
String	?	?	?	?	?	?	?	?	?
Posição	0	1	2	3	4	5	6	7	8

Esse campo é para processamento, somente
Esse campo será exibido na tela

Os seguintes números serão “limpos” (os números em azul, sublinhados para frente, os números em vermelho, tachados, para trás, e o próprio número da posição indicada pelo jogador):

Número	<u>23</u>	22	<u>26</u>	<u>66</u>	<u>68</u>	99	15	9	12
String	?	?	?	?	?	?	?	?	?
Posição	0	1	2	3	4	5	6	7	8

A forma como o programa passará a considerar que uma posição já foi limpa e a forma de exibir o array devem ser definidos pelo próprio programador, por exemplo, alterando o conteúdo de cada posição manipulada.

A cada jogada a pontuação do jogador deve ser atualizada (quantidade total de posições que foram “limpas”) e o array deve ser novamente apresentado para que o usuário saiba quais posições foram limpas. Caso o jogador repita uma posição que já foi limpa ele perderá a jogada sem contar a pontuação. Além disso, caso o jogador indique uma posição que tem um número ímpar essa posição deve ser limpa sem aumentar a pontuação do jogador.

Após completar as 10 jogadas o jogo deve mostrar uma mensagem convidando o jogador a jogar novamente ou encerrar o programa.

PONTUAÇÃO:

- Criar e instanciar o array de registros: 1,0
- Preencher o array de registros com números aleatórios e caracteres: 1,5
- Exibir o array “escondendo” os números e apresentando somente os caracteres: 0,5
- Permitir que o jogador “chute” uma posição: 0,5
- A partir da posição indicada pelo jogador, limpar as posições adequadas, atualizar e exibir a pontuação e o array em caso de acerto: 2,5
- A partir da posição indicada pelo jogador, limpar a posição adequada, atualizar e exibir o array em caso de erro: 1,0
- Permitir que o usuário jogue o jogo novamente: 1,0
- Implementar o item “b” em uma sub-rotina: 0,5
- Implementar os itens “e” e “f” em uma sub-rotina: 1,0
- Implementar o item “c” em uma sub-rotina: 0,5.

ORIENTAÇÕES PARA PROGRAMAR EM JAVA SEM NECESSITAR DE INTERNET

Exemplo de programa em Java:

```
public class Teste // o nome dessa classe tem que ser o mesmo nome do arquivo
{
    public static void main(String[] args)
    {
        System.out.println("Alô mundo!");
    }
}
/* o arquivo deve ter a extensão .java */
```

- Para escrever o programa use uma IDE ou editor de textos comum, como o Bloco de Notas, por exemplo.
- Para compilar o programa abra um terminal (o cmd do Windows), navegue até o diretório onde o arquivo está salvo e execute o comando: `javac <nome do arquivo .java>`
- Caso tenha algum erro a mensagem do compilador será exibida no terminal, edite o programa para resolver os problemas e repita o passo 3
- Para executar o programa digite o comando: `java <nome do arquivo sem o .java>`

Abraço e boa prova