Especificação do Trabalho 01

Elabore uma aplicação em Java para implementar um jogo de Sudoku, a qual deve permitir que o jogador possa jogar quantas vezes desejar, cancelar o jogo em andamento ou encerrar a aplicação. Ao final, o programa deve apresentar um relatório de cada jogo, incluindo: quantidade de tentativas válidas e inválidas, tempo decorrido, matriz inicial e matriz resolvida.

Para este trabalho, adotaremos um Sudoku com uma matriz 9 por 9:

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

Fonte: https://en.wikipedia.org/wiki/Sudoku

A regra geral para finalizar o jogo é completar todas as 81 células usando números de 1 a 9, sem repetir os números numa mesma linha, coluna ou grade (3x3). Na figura anterior, a primeira grade está destacada. Para uma matriz 9 por 9, temos 9 grades. Outra regra é um jogo de Sudoku deve possuir uma única solução (entretanto, esta última regra não precisa ser garantida nesta aplicação).

A cada rodada, o jogo deve solicitar que o jogador entre com a identificação da célula e do valor a ser colocado. A aplicação deve verificar se o valor é válido, conforme o preenchimento atual da matriz. Se o valor informado for zero, a aplicação deve interpretar que o jogador deseja limpar a respectiva posição da matriz.

Informações sobre o jogo podem ser encontradas em https://pt.wikipedia.org/wiki/Sudoku.

A cada rodada, um novo Sudoku deve ser criado. O preenchimento inicial deve ser feito de modo aleatório, não permitindo repetições de matrizes iniciais já utilizadas. Existem várias abordagens para identificar o nível de dificuldade de um jogo de Sudoku. Nesta aplicação, adotaremos três níveis de dificuldade:

- Fácil – 36 dicas, onde cada grade possui, no mínimo, 3 dicas.
- Médio – 32 dicas, onde cada grade possui, no mínimo, 3 dicas.
- Difícil – 27 dicas, onde cada grade possui, no mínimo, 2 dicas.

Existem muitas propostas para a geração de um jogo de Sudoku. Para este trabalho, você pode utilizar como referência o link https://blog.forret.com/2006/08/14/a-sudoku-challenge-generator/. Entretanto, você tem liberdade para consultar outras fontes. Seja qual for a sua solução, ela deverá estar documentada.

Especificação do Trabalho 01

Orientações para o desenvolvimento

- Apresentar o diagrama de classe (UML) completo na ferramenta StarUML. Inicie pela modelagem e aplique o princípio da abstração. Evite começar pela programação. Na representação dos relacionamentos, não esqueça de utilizar nomes de papel, cardinalidade (mesmo quando for 1) e navegabilidade.
- Observe as boas práticas de orientação a objetos discutidas em sala e trabalhadas nos roteiros. Adote nomes adequados para as classes, observe a coesão tanto nas classes quanto nos métodos.
- As classes de entidade (que armazenam informações) e de processamento não devem interagir diretamente com o usuário. Crie classes específicas para entrada e saída de dados.
- Todas as classes e métodos devem ser documentados com notação javadoc. Para classes, você deve colocar uma breve descrição e utilizar o annotation "@author". Para cada método, você deve ter uma ou mais linhas explicando o comportamento e utilizar os annotations "@param" (explicando cada parâmetro) e "@return" (quando a operação for uma função). Veja a documentação disponibilizada no ambiente Material Didático (slides de referência sobre a linguagem Java).
- Não é necessário trabalhar com interface gráfica, mas a interação com o jogador e a apresentação de informações na tela deve ser amigável. Por exemplo, a cada rodada, a matriz deve ser apresentada de modo a facilitar a escolha da célula a ser preenchida.
- O jogo termina quando a matriz estiver completamente preenchida. Lembre-se que a cada rodada, a aplicação deve avaliar se as regras estão sendo atendidas. Quando alguma regra for quebrada, uma mensagem de aviso deve ser emitida, o valor informado deve ser desconsiderado e um novo valor deve ser solicitado.
- Para o algoritmo de geração da matriz, avalie a criação de uma classe Matrix (Matriz).
- Sua aplicação deve implementar a classe Console (especificação UML abaixo) para a leitura dos dados. Leia os slides de referência para conhecer a classe Scanner. O parâmetro String nas operações representam o texto a ser apresentado ao usuário (por exemplo, "Entre com o nome do jogador: "). Elas devem retornar o valor informado pelo usuário. A operação "waitEnter()" forçar o usuário a clicar na tecla "Enter" para continuar (você pode implementar esta operação a partir das anteriores). Vocês tem liberdade para fazer adaptações.