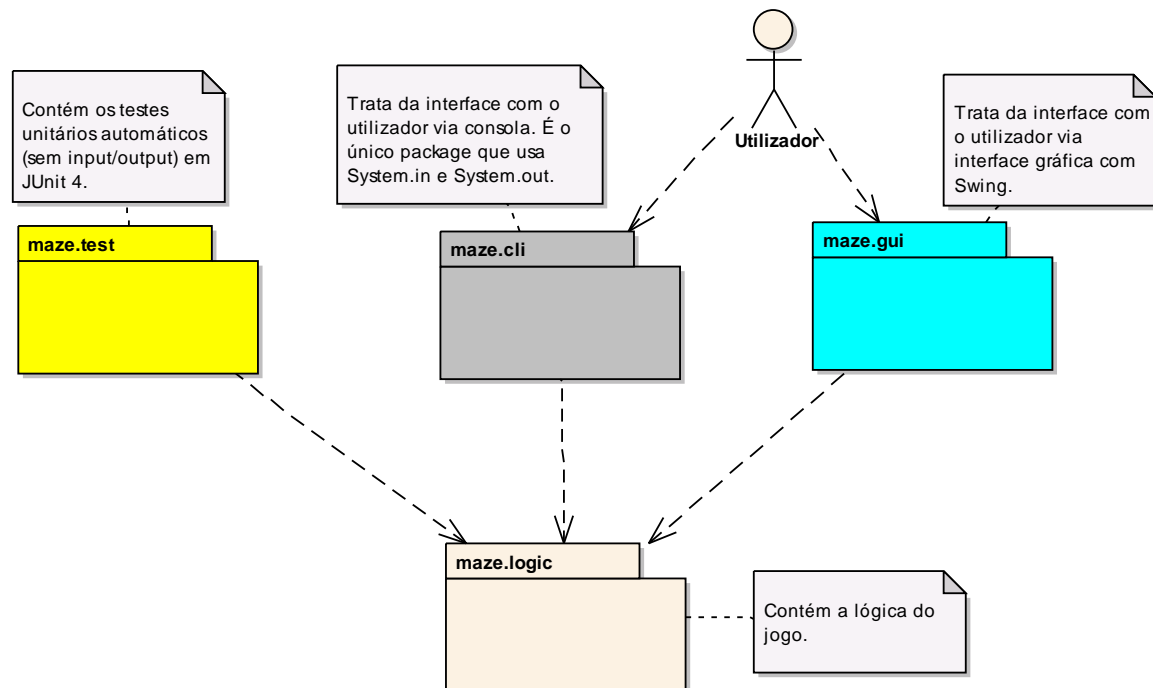


## Projeto Guiado – 3ª Iteração

### Desenvolvimento de interface gráfica para o utilizador – 1ª parte

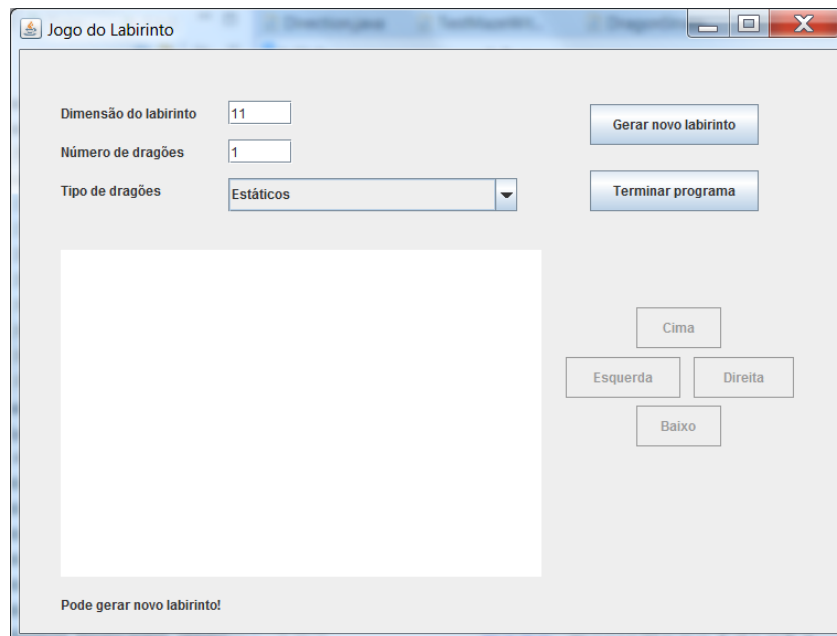
3. Adicionar ao programa do jogo do labirinto uma interface gráfica para o utilizador (GUI – *Graphical User Interface*) em Swing. Todo o código da GUI deve ficar num *package* separado (como por exemplo, *maze.gui*). Deve ser possível continuar a arrancar o programa com a interface de linha de comando. O programa deve ter uma estrutura semelhante à indicada no diagrama UML seguinte (diagrama de *packages*). Nesta iteração, o estado atual do jogo é mostrado numa área de texto da mesma forma que na consola, sendo a movimentação do herói controlada por botões.



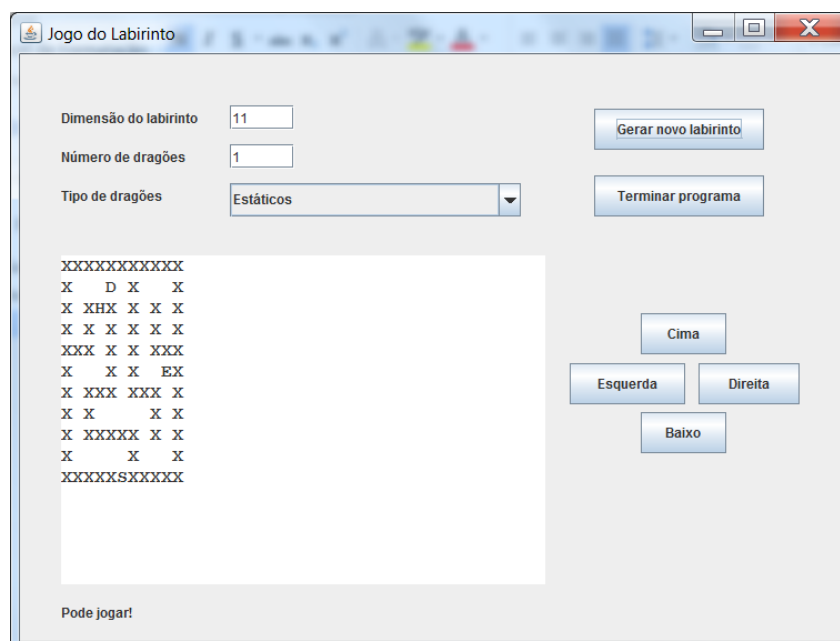
3.1. Com a ajuda do WindowBuilder criar uma janela de nível de topo (Application Window), contendo os seguintes componentes (ou outros que permitam atingir a mesma finalidade):

- Caixa de texto (JLabel + JTextField) para o utilizador introduzir a dimensão do labirinto, com o valor 11 por omissão;
- Caixa de texto (JLabel + JTextField) para o utilizador introduzir o número de dragões, com o valor 1 por omissão;
- ComboBox (JLabel + JComboBox) para o utilizador selecionar o tipo de dragões; os valores possíveis são adicionados programaticamente com addItem;
- Botão (JButton) para gerar novo labirinto;
- Botão (JButton) para terminar o programa;
- Área de texto (JTextArea) não editável para mostrar o estado atual do labirinto (como na consola), configurada como uma *font* proporcional, como Courier New;
- Botões (JButton) para mover o herói para a esquerda, direita, cima ou baixo (inicialmente desativados);
- Etiqueta (JLabel) para indicar o estado atual do programa e próximos passos do utilizador (inicialmente diz para gerar labirinto).

Um exemplo meramente ilustrativo é indicado na figura abaixo.



**3.2.** No editor visual do WindowBuilder, fazer duplo clique no botão de gerar labirinto, e preencher o código do corpo *listener*. O *listener* deve gerar um novo labirinto de acordo com os parâmetros definidos, deve mostrá-lo na área de texto (`setText()`), deve ativar (`setEnabled(true)`) os botões de movimentação do herói e deve atualizar a etiqueta com o estado atual. Experimentar, executando múltiplas vezes o gerador. Um exemplo meramente ilustrativo depois de gerar um novo labirinto é indicado na figura abaixo.



**3.3.** Criar um *listener* para o botão de terminar programa. Basta fazer `System.exit(0)`. Experimentar.

**3.4.** Criar *listeners* para os botões de movimentação do herói, recorrendo a uma rotina auxiliar para evitar código duplicado. Essa rotina deve movimentar o herói na direção pretendida, deve mostrar a nova situação do labirinto na área de texto, e deve atualizar a informação na etiqueta de estado. Se o jogo terminar, deve colocar mensagem explicativa na etiqueta de estado e deve desativar os botões de movimentação (`setEnabled(false)`).

**3.5.** Robustecer o programa para tratar adequadamente dados de entrada inválidos (dimensão do labirinto e número de dragões).