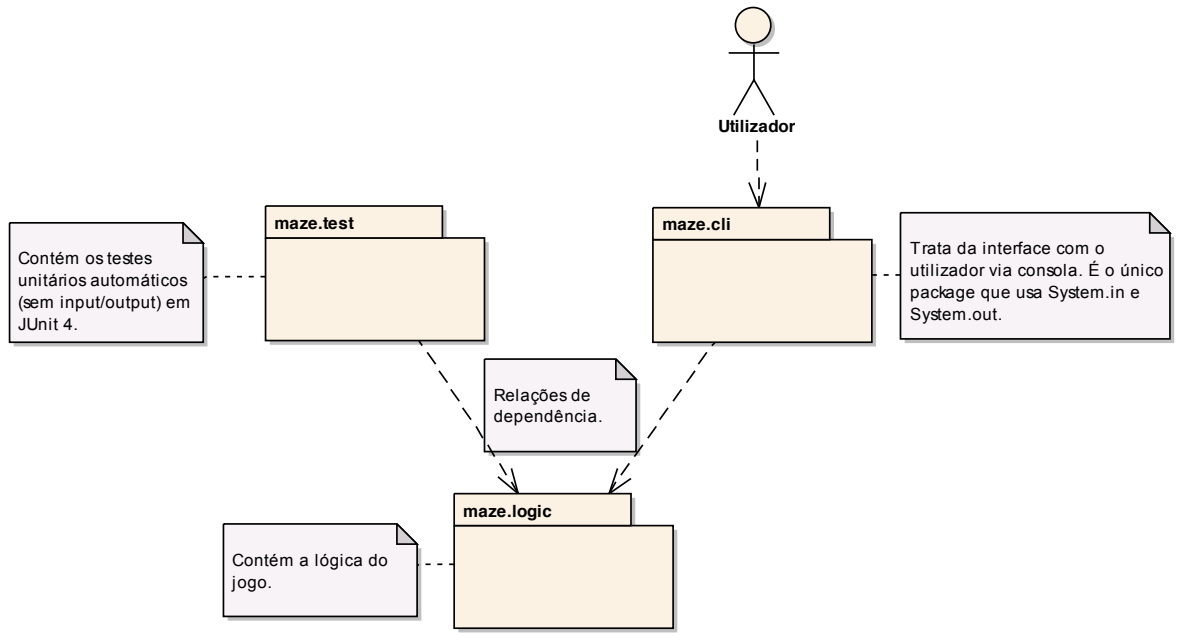
Projecto Guiado – 3ª Iteração
Testes unitários

3.1 Desenvolver uma classe de testes unitários automáticos (sem input/output) em JUnit 4, usando o labirinto da 1ª aula (ou outro julgado adequado) para efeito de teste, para testar o jogo no modo mais simples, em que só há um dragão imóvel. Criar vários métodos de teste para testar as seguintes situações (um método para cada situação):

i. herói mover-se uma posição (quando se manda deslocar em direção a célula livre);
ii. herói imóvel (quando se manda deloscar em direção a uma parede);
iii. apanhar a espada;
iv. ser morto pelo dragão (derrota);
v. matar o dragão;
vi. alcançar a saída após apanhar a espada e matar dragão (vitória);
vii. alcançar a saída sem ter apanhado a espada ou morto o dragão (não termina).

Sugestão: em cada método, preparar um array ou string com a sequência de comandos de movimentação do herói, invocar uma rotina auxiliar para executar essa sequência de comandos, e verificar depois se o estado final é o esperado.

O resto do programa deve continuar a funcionar da mesma forma. No final deste exercício, a estrutura do programa deve ser semelhante à indicada na figura seguinte (os nomes dos *packages* podem ser diferentes).



3.2 Criar métodos de teste adicionais (na mesma ou noutra classe de teste) para testar o jogo com os comportamentos mais complexos do dragão (movimentação, dormir, múltiplos dragões, cuspir fogo) e do herói (apanhar escudo, apanhar dardos, lançar dardos). Para efeito de teste, convém que os movimentos do dragão sejam comandados pelo código de teste, em vez de ocorrerem de forma aleatória. O resto do programa (interação com o utilizador) deve continuar a funcionar da mesma forma.

3.3 Com a ferramenta EclEmma de análise de cobertura para analisar a cobertura dos testes (cobertura de instruções) e acrescentar mais casos de teste se necessário para garantir que a cobertura do pacote com a lógica do jogo é de pelo menos 75%.