

# LABIRINTO

## CLASS CaminhoMaisCurto

**Método *public List<Coordinate>solucao (Labirinto labirinto)***

Para encontrarmos o caminho mais curto utiliza-se a seguinte lógica:

→ Adicionamos inicialmente a posição de **partida** na Lista que irá armazenar as coordenadas de cada posição dos possíveis caminhos até chegar à **chegada**;

→ Enquanto a Lista não está vazia, pegue uma posição ainda não visitada e faça o seguinte:

→ Se chegarmos à parede (posição bloqueada, “X”) ou esta posição já for visitada anteriormente, segue-se para a próxima iteração;

→ Se encontrarmos a posição de **chegada**, volte da posição atual (**chegada**) até a posição inicial (**partida**) da Lista para encontrar o caminho mais curto;

→ Caso contrário, adicionamos todos os vizinhos imediatos da posição atual nas quatro direções (posição à direita, esquerda, acima e abaixo).

**Método *private List<Coordinate>voltaCaminho (Coordinate cur)***

Recebe a coordenada de uma posição e a partir dela percorre o “nó pai” iterativamente. Ou seja, volta-se da posição atual (**cur**) até a posição inicial (**partida**) a partir dos “nós pais” e assim, encontra-se o caminho mais curto.

## CLASS CaminhoMaisLongo

**Método *public List<Coordinate>solucao (Labirinto labirinto)***

Faz-se a chamada do método *Explora* para verificar se existe um caminho possível para sair da posição de **partida** e chegar no posição de **chegada**. Se for retornado false, então não existe um caminho possível.

**Método *private boolean explora (Labirinto labirinto, int x, int y, List<Coordinate>caminho)***

Para encontrarmos o caminho mais curto utiliza-se a seguinte lógica:

- Se a posição não é válida ou se já foi percorrido essa posição, retorna-se *false*;
- Se a posição for a chegada, retorna-se *true*;
- Caso contrário, adicione a posição na Lista e recursivamente explora os quatro vizinhos (à direita, esquerda, acima e abaixo) da posição.
  - Se a partir dos vizinhos NÃO for encontrado a posição de chegada, remove-se a posição atual da Lista de caminhos e retorna-se *false*.