

# Exercício 4

Tipos genéricos de dados  
Iteradores



**Ciências**  
**ULisboa**

Unidade Curricular de  
Laboratórios de Programação

2021/2022

# Trajetórias num espaço 2D

## Objetivos

- Reforço de conhecimentos sobre classes genéricas;
- Uso de iteradores em classes Java.

## Antes de Começar

- Descarregar o arquivo `studentsExercise4.zip` disponível na página de LabP e descompactá-lo. Esse arquivo contém:
  1. Os ficheiros `Building.java`, `Obstacle.java`, `PathType.java`;
  2. O ficheiro `RunWalkPath.java`;
  3. O ficheiro `TestsWalkPath.java`, contendo vários testes *JUnit*;
  4. O ficheiro de texto `expectedOutput.txt` contendo o resultado esperado da execução de `RunWalkPath`.
- No Eclipse, criar um novo projeto Java e, em seguida:
  - a) Copiar para dentro da pasta `src` desse projeto os ficheiros dados com código Java;
  - b) Configurar o *Build Path* desse projeto Java para incluir, na *classpath*, a biblioteca *JUnit5* (veja como o fazer na página 14 do tutorial sobre o Eclipse IDE).

## Algumas informações úteis

Para realizar este exercício deverá consultar os guiões sobre tipos genéricos de dados e sobre iteradores (publicados na página de LabP juntamente com este exercício) e a matéria dada sobre os mesmos temas em AED.

Poderá ainda consultar a documentação da API do java sobre as interfaces `Iterator<E>`

(<https://docs.oracle.com/en/java/javase/12/docs/api/java.base/java/util/Iterator.html>) e `Iterable<E>`

(<https://docs.oracle.com/en/java/javase/12/docs/api/java.base/java/lang/Iterable.html>).

## Contexto

Um espaço retangular onde podemos encontrar vários elementos de diversas naturezas (edifícios, monstros, desafios, etc) é interessante no âmbito de qualquer jogo simples em que os intervenientes se movimentam e reagem aos elementos que vão encontrando.

Neste exercício não vamos criar nenhum jogo, mas vamos construir a base necessária à existência desse tal espaço, através do qual os intervenientes do jogo se podem movimentar, seguindo uma dada trajetória.

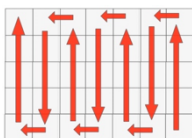
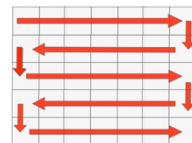
Pretende-se a criação de um conjunto de tipos java (classes e enumerados) que permitam:

- a construção de um espaço desses, com determinadas dimensões,
- o seu povoamento com elementos variados pertencentes a um determinado universo,
- a iteração através desses elementos seguindo um determinado tipo de trajetória através do espaço.

Neste espaço, a posição de cada elemento é definida por duas coordenadas inteiras, podendo o espaço ser visto como uma matriz (com linhas e colunas).

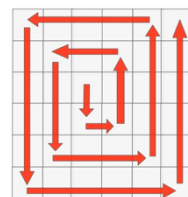
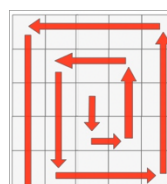
Para já, queremos poder iterar sobre os elementos do espaço usando qualquer uma das três trajetórias seguintes:

- **Lagarta horizontal:** o primeiro elemento nesta trajetória é o que está na posição definida pela primeira linha e primeira coluna; a ordem dos elementos nesta trajetória é a que está representada na figura à direita.



- **Lagarta vertical:** o primeiro elemento nesta trajetória é o que está na posição definida pela última linha e última coluna; a ordem dos elementos nesta trajetória é a que está representada na figura à esquerda.

- **Espiral:** este tipo de trajetória só é possível em espaços quadrados, ou seja, com igual número de linhas e colunas; o primeiro elemento nesta trajetória (ver figuras à direita) é o que está na posição exatamente a meio do espaço (no caso do número de linhas e colunas ser ímpar) ou na posição mais acima e à esquerda do quadrado 2x2 central do espaço (no caso do número de linhas e colunas ser par); a ordem dos elementos nesta trajetória é a que está representada nas figuras.



## O que é para fazer, então?

O objetivo deste Exercício 4 é a construção da classe genérica `Landscape<E>`, tal como descrita mais à frente, cujas instâncias são espaços 2D quadrangulares, populados por elementos de um dado tipo, que podem ser iterados seguindo determinadas trajetórias.

No `zip studentsExercise4` pode encontrar os seguintes enumerados:

- `PathType`, que define os três tipos de trajetória de iteração descritos atrás;
- `Building`, que define vários tipos de edifícios;
- `Obstacle`, que define vários tipos de obstáculos.

E as seguintes classes:

- `RunWalkPath`, que cria dois *landscapes* e itera sobre os seus elementos, usando os vários tipos de iteradores;
- `TestsWalkPath`, uma classe que permite testar a classe `Landscape`.

Deverá construir então a classe genérica `Landscape<E>`, que implementa o *interface* `Iterable<E>`. Os objetos desta classe têm um espaço 2D associado (matriz) e um tipo de trajetória atual (que permite saber que tipo de iterador deve fornecer). Deverá oferecer os seguintes métodos públicos:

- `public Landscape(E[][] space):` construtor que, assumindo que `space` é uma matriz (todas as linhas têm o mesmo número de elementos) inicializa o espaço do novo *landscape* com esta matriz; por defeito, o tipo de iterador do novo *landscape* definirá uma trajetória de `HORIZONTAL_SNAIL`;
- `public void setPathType(PathType pathT):` método que altera o tipo de iterador deste *landscape*, para um que defina uma trajetória do tipo `pathT`; caso `pathT` seja `SPIRAL` e o espaço deste *landscape* não seja quadrado, o novo tipo de iterador será `HORIZONTAL_SNAIL`;
- `public Iterator<E> iterator():` método que devolve um novo iterador do tipo atualmente definido para este *landscape*;
- `public String toString():` método que devolve a representação textual deste *landscape*.

### NOTA 1:

Esta classe deve definir 3 classes internas privadas (rever guião sobre iteradores), que implementam o *interface* `Iterator<E>`, definindo cada um dos três tipos de iteradores para as trajetórias apresentadas anteriormente.

### NOTA 2:

Tal como explicado no guião, o ciclo `for-each` utiliza de forma implícita o iterador associado ao objeto sobre o qual é feita a iteração. O iterador que é usado

implicitamente é precisamente o objeto do tipo `Iterator` devolvido pelo método `iterator()` definido no *interface* `Iterable`.

## Entrega

Deve entregar um *zip* de nome `E4fcxxxxx.zip`, onde `xxxxx` é o seu número de aluno, contendo somente o ficheiro `Landscape.java`.

Data e hora limite: 29 de Abril às 23h55

**ATENÇÃO:** Antes de submeter o trabalho, verifique que documentou a sua classe (incluindo `@author` com o seu número de aluno).