

## Taller 1

### Uso de bibliotecas (APIs) y ADTs

#### **Introducción**

Una biblioteca de funciones encapsula tipos de datos abstractos y algoritmos de uso frecuente en el desarrollo de software.

Los autores de el texto guía ofrecen un API para facilitar operaciones frecuentemente utilizadas. La biblioteca de funciones se encuentra disponible en la siguiente dirección:

<http://algs4.cs.princeton.edu/code/>

Allí están disponibles el código fuente, la documentación y archivos ejemplos para los distintos algoritmos. La biblioteca entera está disponible como un archivo JAR ([algs4.jar](#)), el cual se debe incluir en el CLASSPATH del proyecto (ver instrucciones en la misma página).

Para efectos de esta práctica nos interesan 5 clases incluidas en esta biblioteca:

[StdIn](#) : Funciones para lectura de datos por consola

[StdOut](#) : Funciones para escritura de datos en consola

[In](#) : Funciones para leer datos desde archivos o URLs

[Out](#) : Funciones para escribir datos en archivos

[StdDraw](#) : Funciones para hacer gráficas

[StdRandom](#) : Funciones para generar números aleatorios

[StdStats](#) : Cálculos estadísticos básicos

Otra biblioteca de uso frecuente está incluida en el API de Java es la clase Arrays, que nos permite hacer varias cosas utilizando arreglos de datos: Ordenarlos, convertirlos a String, hacer búsquedas, etc. La clase está documentada aquí: [java.util.Arrays](#).

## **Ejercicios a desarrollar**

Se quiere implementar una versión muy básica de Pacman, donde solo se tienen el Pacman y el Laberinto. El Pacman se implementa por un ADT que representa sus coordenadas en el laberinto y su dirección de movimiento. El Laberinto se representa por medio de una matriz binaria, donde los 0 son los espacios por los que puede caminar Pacman y los 1 son los muros.

El objeto Pacman avanza una posición en su dirección de movimiento por cada unidad de tiempo. El objeto Pacman debe validar si un paso es valido utilizando el método `validarPosicion` del objeto Laberinto. Para cambiar de dirección se puede hacer uso de los eventos de teclado definidos en la clase `StdDraw` ([`hasNextKeyTyped`](#), [`nextKeyTyped`](#), [`isKeyPressed`](#)).

El objeto Laberinto acepta en su constructor la ruta de un archivo de texto donde se tiene el laberinto descrito por espacios (los caminos) y por 'X' los muros. De esta forma al instanciar el laberinto se carga el mapa desde un archivo existente. Hacer uso de la biblioteca [`In`](#) para la lectura del archivo.

Para la graficación del Pacman y el Laberinto hacer uso de la biblioteca [`StdDraw`](#).

## **Ejercicios a desarrollar**

1. Implementar los ADT Pacman y Laberinto de acuerdo a las especificaciones del problema. Implementar en cada ADT un método `pintar` que se encargue de la graficación de cada ADT. Garantizar la encapsulación de la representación de cada ADT.
2. Implementar una clase `Test` que solo contiene el método `main`. Esta clase debe crear una instancia de cada ADT y hacer pruebas unitarias de los métodos que componen su interfaz pública.
3. Implementar una clase `Principal` que contiene el `main` del juego. Cuando se ejecuta esta clase se instancian los ADT Pacman y Laberinto. Dentro de un ciclo se repite la operación mover del Pacman de acuerdo con la entrada del teclado. Para regular la velocidad de movimiento, se define un retardo entre paso y paso (típicamente entre 100-500ms). Cuando se presiona la tecla ESC debe salir del programa.

## **Entregables:**

Enviar por correo electrónico código fuente del proyecto comprimido. Seguir el estándar de nombres: `Practica1-<NombreApellido1>-<NombreApellido2>.zip`. No incluir dentro del comprimido la biblioteca `algs4.jar`.

El trabajo se puede realizar en grupos de máximo 2 personas.

Fecha de entrega: 14/Ago/2019.