

Taller 1

Tipos de datos abstractos (ADT) y estructuras básicas

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](#).

Ejercicio a desarrollar

Se desea implementar una simulación (simplificada) del movimiento de un BalonDeBasket que rebota sobre una superficie. Para tal efecto se requiere implementar el ADT BalonDeBasket que representa el balón de basket por medio de sus coordenadas en el plano (x,y) y su vector velocidad (vx,vy). Dentro del API del BalonDeBasket se necesitan las siguientes operaciones:

draw : Dibuja un balón de basket en sus coordenadas actuales.

move : Ajusta la posición según un parámetro delta_t que representa el tiempo desde el último movimiento. La altura del balón se establece de acuerdo con la ecuación de caída libre y la posición horizontal se toma como un movimiento con velocidad constante.

collision : Detecta si el balón golpea contra el piso y ajusta el vector velocidad así:

- La velocidad vertical cambia de signo.
- La velocidad horizontal continua igual.
- collision toma como referencia la altura del piso $h=0$.

fueraDelCampo: Determina si el balón salió del campo, devuelve true en caso afirmativo.

Para representar el campo de baloncesto (y como clase principal de la simulación) se define el ADT CampoBaloncesto, el cual realiza las siguientes funciones:

- Define el tamaño del campo, considerar solo el ancho (reglamentario es 15m).
- Define un [Bag](#) que contiene los BalonDeBasket.
- El constructor acepta un parámetro N y crea N objetos BalonDeBasket con posiciones aleatorias, velocidades en 'x' aleatorias y velocidades iniciales en 'y' de cero.
- La única operación del API es runSimulation la cual consta de un ciclo para simular el movimiento por T unidades de tiempo. Las T unidades de tiempo se dividen en intervalos delta_t y por cada intervalo se debe, para cada balón en la bolsa, determinar si tuvo colisión, hacer el movimiento correspondiente a un delta_t y dibujarlo en su nueva posición.
- Si cuando se hace el movimiento de un balón este sale del campo, se borra de la bolsa para eliminarlo de la simulación.

El método main de CampoBaloncesto inicializa la simulación con N BalonDeBasket e invoca el método runSimulation.

Entregables:

Clases implementado los ADT definidos: BalonDeBasket y CampoBaloncesto.

Métodos main: En BalonDeBasket hace pruebas unitarias de la clase. El main de CampoBaloncesto es el main principal del programa.

Enviar solo el código fuente como un archivo comprimido (se aceptan .zip, .rar, .7z, .tgz). No incluir dentro del comprimido la biblioteca algs4.jar.

Se puede realizar el trabajo en equipos de máximo 2 personas.

Seguir el estándar de nombres: <NombreApellido1>-<NombreApellido2>-Practica1.zip.

Anexo

Ecuaciones del movimiento en el plano, bajo la presencia de un campo gravitatorio uniforme:

$$x = x_0 + v_{ox} \cdot t$$
$$y = y_0 + v_{oy} \cdot t + \frac{g \cdot t^2}{2}$$

donde x_0 , y_0 son la posición inicial, v_{ox} , v_{oy} son la velocidad inicial, g es la aceleración de la gravedad y t es el tiempo.