

Práctica 7

Objetivos: Practicar recorridos y búsquedas en tablas

Descripción: Escribir un programa que permita dibujar en una ventana las trayectorias de dos animales que se han medido con un GPS

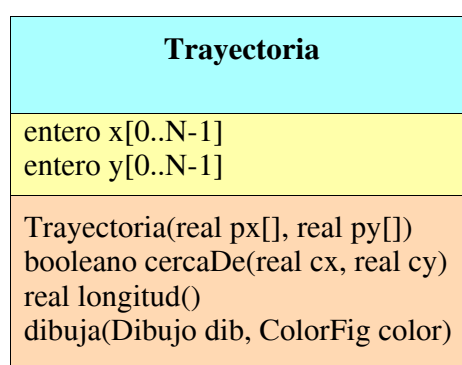
El programa se compone de dos clases:

- **Trayectoria:** Tiene una lista de puntos, cada uno de ellos con sus coordenadas x e y
 - las coordenadas se expresan en unidades que son puntos de la pantalla (píxeles)
- **Animales:** Tiene el programa principal

Práctica 7: Clase Trayectoria

La clase tiene el diagrama de clase que se muestra

- **atributos:**
 - dos arrays de números enteros llamados x e y , para guardar una lista de puntos en el plano (el punto i es el de coordenadas $x[i]$, $y[i]$)
- **métodos**
 - constructor al que se le pasan dos arrays del mismo tamaño (px y py), uno con las coordenadas x , y otro con las coordenadas y de una trayectoria; debe crear para los atributos x e y dos nuevos arrays y copiar en ellos los datos de px y py



Práctica 7: Clase Trayectoria

- **cercaDe()**: retorna un booleano que indica si alguno de los puntos de la trayectoria está cerca (en un radio de 5 unidades) del punto de coordenadas **cx,cy**; usar para este método el esquema de búsqueda en tablas de teoría; la distancia entre dos puntos se calcula como

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

- **longitud()**: retorna la longitud del recorrido consistente en ir desde el primer punto de la trayectoria hasta el último, pasando por los intermedios en el orden en que aparecen en las tablas; usar el esquema de recorrido (parcial) en tablas

Práctica 7: Clase Trayectoria

- métodos (cont.)

- **dibuja()**: dibuja la lista de puntos en el objeto de la clase **fundamentos.Dibujo** que se pasa como parámetro; el dibujo se hace dibujando un polígono abierto (con **dibujaLineas**); el color del lápiz se pasa como parámetro; no se invoca al método **espera()** (esto lo hará el programa principal)

Práctica 7: Clase **Animales**

Esta clase tiene el programa principal, que hace lo siguiente:

- Crea dos objetos de la clase **Trayectoria**, para las trayectorias de cada animal; los datos son fijos (no leer de teclado)
- Crea un objeto de la clase **Dibujo** (paquete **fundamentos**)
- Pinta las trayectorias de los animales con el método **dibuja()**, usando colores distintos para cada una
- Lee de teclado las coordenadas de un punto que es un comedero, y lo pinta en el dibujo
- Comprueba si alguno de los animales ha estado cerca del comedero con **cercaDe()** y muestra los resultados en pantalla
- Llama al método **espera()** del dibujo para mostrar el dibujo

Práctica 7: Clase **Animales**

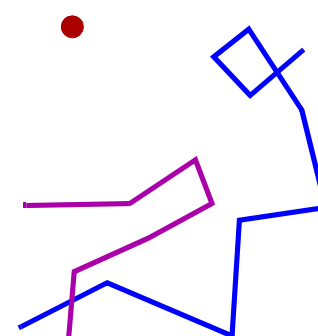
Opcional. Poner en el dibujo una leyenda:

- nombre de cada animal (texto) junto a una pequeña línea de su color

La figura muestra un ejemplo del dibujo esperado

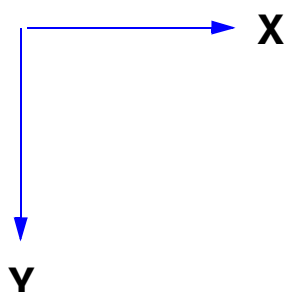
Elefante ———
Antílope ———

comedero



Práctica 7 (cont.)

Recordar que el origen de coordenadas está en la esquina superior izquierda de la pantalla



Informe. Entregar:

- la especificación y el diseño de los métodos **cercaDe** y **longitud()** de la clase **Trayectoria**
- el código de las clases **Trayectoria** y **Animales**