



Programación Declarativa
Ingeniería Informática
Cuarto curso. Primer cuatrimestre.



Escuela Politécnica Superior de Córdoba
Universidad de Córdoba
Curso académico: 2024 - 2025

Práctica número 5: tipos abstractos de datos y lectura y escritura

1. Tipo abstracto de datos “club” deportivo.
 - Utiliza “**listas de asociación**” para codificar las funciones de **gestión** (creación, acceso, consulta y modificación) del tipo abstracto de datos **club**, que posee los campos *nombre*, *ciudad* y *año de fundación*.
2. Tipo abstracto de datos “campeonato”
 - Utiliza las funciones del tipo abstracto **club** para codificar las siguientes funciones del tipo abstracto “**campeonato**”, que está compuesto por clubes.
 - **crear-campeonato**
 - Crea un campeonato sin clubes.
 - **campeonato-vacío?**
 - Comprueba si un campeonato está vacío.
 - **cargar-campeonato**
 - Lee los datos de los clubes desde un fichero y los almacena en un campeonato.
 - **grabar-campeonato**
 - Graba en un fichero los datos de los clubes.
 - **existe-club?**
 - Comprueba si un campeonato contiene a un club.
 - **insertar-club!**
 - Inserta un club en un campeonato.
 - **borrar-club!**
 - Borra un club de un campeonato.
 - **modificar-club!**
 - Modifica un club del campeonato.
 - **consultar-club**
 - Muestra por pantalla los datos de un club.
 - **consultar-clubes**
 - Muestra por pantalla los datos de todos los clubes.
 - **consultar-clubes-ciudad**
 - Muestra por pantalla los clubes de una ciudad.
 - *Etc.*
3. Codifica un programa que permita la gestión de un “campeonato”
 - Crear un campeonato.

- Comprobar si un campeonato está vacío.
 - Cargar los clubes de un campeonato desde un fichero.
 - Grabar los clubes de un campeonato en un fichero.
 - Insertar un club en un campeonato.
 - Modificar un club.
 - Borrar un club.
 - Consultar un club.
 - Consultar los clubes de una ciudad.
 - Etc.
-

4. Tipo abstracto de datos “polígono”

- Se desea representar el tipo abstracto de datos “polígono” en el plano euclidiano $P = ((x_1, y_1), \dots, (x_n, y_n))$
- Se podrá representar el polígono con un vector o una lista vértices.
 - Los vértices están unidos por segmentos o lados del polígono.
 - El último vértice está unido al primero.
- Codifica las siguientes funciones
 - **crear-polígono:**
 - Crea un polígono vacío: sin puntos.
 - **polígono-nulo?**
 - Comprueba si el polígono es nulo, es decir, si tiene menos de tres vértices.
 - **número-vértices**
 - Devuelve el número de vértices del polígono.
 - **existe-vértice?**
 - Comprueba si un vértice pertenece al polígono.
 - **mostrar-vértices**
 - Muestra todos los vértices del polígono.
 - **insertar-vértice**
 - Inserta un vértice en el polígono.
 - Se deben indicar las coordenadas del vértice y la posición que ocupará en el polígono.
 - **borrar-vértice**
 - Borra un vértice del polígono.
 - Se debe indicar la posición del punto que se desea borrar.
 - **grabar-polígono**
 - Graba el polígono en un fichero.
 - **cargar-polígono**
 - Lee los vértices del polígono desde un fichero y los almacena en un polígono.
 - **perímetro**
 - Calcula el perímetro del polígono: suma de los lados del polígono.
 - **área**
 - Calcula el área del polígono usando el método de Gauss
 - El polígono no debe ser nulo.

- Se debe utilizar la “fórmula determinante de Gauss”

$$A = \left| \frac{1}{2} \sum_{i=1}^N (x_i(y_{i+1} - y_{i-1})) \right|$$

donde

- ✓ N es el número de vértices del polígono
- ✓ Los vértices están numerados de forma cíclica, es decir:
 - $(x_0, y_0) = (x_N, y_N)$
 - $(x_{N+1}, y_{N+1}) = (x_1, y_1)$

5. Codifica un programa que permita la gestión de un “polígono”

- Crear un polígono nulo.
- Comprobar si un polígono es nulo.
- Mostrar un polígono.
- Comprobar si un vértice pertenece al polígono.
- Número de vértices del polígono.
- Insertar un vértice en el polígono.
- Borrar un vértice del polígono.
- Cargar un polígono desde un fichero.
- Grabar un polígono en un fichero.
- Calcular el perímetro de un polígono.
- Calcular el área de un polígono.
- Etc.