

Estructuras de Datos

Práctica 2. Abstracción

Juan Francisco Díaz Moreno
Grado en Ingeniería Informática
Especialidad de Ingeniería del Software
Universidad de Granada

Práctica 2. Abstracción.

Introducción.

El objetivo de esta práctica es la elaboración de dos tipos de datos abstractos (TDA) para gestionar fechas históricas, agrupándolas en cronologías y pudiendo trabajar con ellas.

En primer lugar, se creó el TDA `Mi_Fecha_Historica`, que gestiona un conjunto de eventos que han ocurrido en un año común.

Además, también se creó el TDA `Mi_Cronologia`, que permite agrupar un conjunto de `Mi_Fecha_Historica` y mantenerlas ordenadas por sus años.

TDA `Mi_Fecha_Historica`.

`Mi_Fecha_Historica` es un TDA que contiene 4 datos: el año de los eventos, el número de eventos que contiene, el número de espacios reservados en el vector de eventos, y el propio conjunto de eventos. De esta forma, una `Mi_Fecha_Historica` tendría esta apariencia:

- Año.
- Número de eventos.
- Eventos:
 - Evento1.
 - Evento2.
 - ...
 - EventoN.

Posee una diferencia con respecto al TDA propuesto en la práctica: el año es almacenado con un entero entre -9999 y 9999, en lugar de un entero positivo y un bool que indica si es aC o dC.

Para trabajar con este TDA, se proporcionan las funciones básicas como el constructor por defecto, el constructor de la clase (que funciona introduciéndole un año, un número de eventos y un vector con los mismos), el constructor de copia, el destructor...

En cuanto a los operadores, se han creado los de entrada y salida, el de asignación y el de suma. Este último sólo funciona si el año de las dos `Mi_Fecha_Historica` a sumar es el mismo. Los operadores de entrada y salida trabajan con el siguiente formato:

`Año#Número_de_eventos#Evento1#Evento2#...#EventoN#`

Tiene también dos funciones muy útiles para trabajar con este TDA: una que devuelve el año de la fecha histórica y otra que permite añadir un evento.

Posee una función de búsqueda, que devuelve el número de eventos de ese año que incluyen la clave que se le pase como parámetro, y formatea una `Mi_Fecha_Historica` que también se le pasa como parámetro con los eventos seleccionados.

Finalmente, tiene dos funciones privadas, una que redimensiona el “vector dinámico” cuando es necesario y otra que elimina eventos repetidos.

TDA `Mi_Cronologia`.

Este TDA permite trabajar de forma más cómoda con el TDA `Mi_Fecha_Historica`, pues permite agruparlas y trabajar con ellas. Se trata de un tipo de dato con 3 campos, el número de fechas almacenadas, el número de espacios reservados en el vector de fechas y las propias fechas, de tal forma que un `Mi_Cronologia` correcta tendría la siguiente apariencia:

- Número de fechas.
- Fechas:
 - Fecha1:
 - Año.
 - Número de eventos
 - Eventos:
 - Evento1.
 - Evento2.
 - ...
 - EventoN.
 - Fecha2:
 - Año.
 - Número de eventos.
 - Eventos:
 - Evento1.
 - Evento2.
 - ...
 - EventoN.
 - ...
 - FechaM:
 - Año.
 - Número de eventos.
 - Eventos:
 - Evento1.
 - Evento2.
 - ...
 - Evento N.

De igual forma que `Mi_Fecha_Historica`, cuenta con las funciones básicas, como son los distintos constructores, el destructor, los operadores de entrada y salida, y los de asignación y suma.

También permite añadir una `Mi_Fecha_Historica` y tiene dos funciones de búsqueda, una que devuelve una `Mi_Fecha_Historica` de un año determinado si existe en la cronología y otra devuelve todas las fechas históricas si contienen una clave en sus eventos.

Cuenta con tres funciones privadas, una que redimensiona el vector de fechas cuando es necesario, otra que ordena las fechas por año cada vez que se introduce una nueva y otra que permite encontrar la posición en la que se encuentra una fecha con un determinado año, devolviendo su posición en el vector.

Cabe destacar que tanto la función de introducción de fechas como el operador de suma tienen en cuenta si se introduce una nueva fecha con un año que ya está almacenado en la cronología, en cuyo caso suma las dos `Mi_Fecha_Historica` en lugar de introducir la nueva por separado.