

Sesión de problemas 1

Especificación e implementación de de TAD lineales

Objetivos

- Trabajar con la especificación no formal.
- Implementar (en pseudocódigo) TAD lineales.

1. TAD genérico bicola

Una *bicola de caracteres* es una secuencia de elementos de tipo carácter **con acceso por ambos extremos, izquierdo y derecho, para realizar operaciones de consulta de cada uno de esos extremos, añadir un elemento y borrar un elemento** (bien por el extremo izquierdo, bien por el derecho). **Especifica el TAD genérico bicola, propón una implementación dinámica del mismo y utiliza una bicola de caracteres para la resolución del siguiente problema:**

Una frase es un palíndromo si se lee igual de izquierda a derecha que de derecha a izquierda, ignorando mayúsculas, espacios en blanco y signos de puntuación. Diseña un programa que lea el fichero de texto «datos.dat» y escriba en la primera línea del fichero de texto «resultados.res» los números de orden de aquellas líneas del fichero de entrada que son palíndromos.

Restricciones para este programa: Los únicos tipos de datos que pueden utilizarse son: booleanos, enteros, caracteres, ficheros de texto y bicolas de caracteres (este último tipo debe ser el exportado por la concreción para el tipo carácter de un módulo genérico que implementa el TAD bicola genérica de elementos).



2. TAD para gestión de competición contrarreloj

Se pretende gestionar una competición contrarreloj. Para cada participante en la competición se debe guardar su mejor tiempo. Cada participante se identifica con un número natural (su dorsal), y los tiempos son también números naturales. Se necesitan las siguientes operaciones:

1. **iniciar**: nos devuelve la estructura vacía (competición sin participantes);
2. **registrar**: dado un participante p y un tiempo t , si p estaba ya en la estructura de datos con un tiempo inferior a t , entonces la operación no hace nada; por el contrario, si p no estaba en la estructura de datos o estaba con un tiempo mayor que t , entonces t pasa a ser el tiempo de p ;
3. **descalificar**: dado un participante p , lo elimina de la estructura de datos;
4. **ganador**: devuelve el participante con un tiempo menor;
5. **tiempo?**: dado un participante p , nos dice cuál es su (mejor) tiempo.

Escribe la especificación de un TAD que sirva para gestionar la competición.

3. TAD genérico *listas con puntos de interés*

Especifica e implementa el TAD genérico *listas con punto de interés*. Los valores del TAD son secuencias (vacías o no) de elementos en las que existe un elemento distinguido de la secuencia, denominado punto de interés, y que cuenta con operaciones relativas a dicho punto de interés. **Deben existir operaciones para crear una lista vacía y para saber si una lista es vacía o no. También deben existir operaciones similares a las de un iterador**: situar el punto de interés en el primer elemento de la secuencia, averiguar el valor del elemento que actualmente es punto de interés, avanzar el punto de interés al siguiente elemento, averiguar si el punto de interés está ya al final de la secuencia (es decir, detrás del último elemento). Además, **debe haber una operación para insertar un nuevo elemento justo antes del punto de interés actual** (y el punto de interés no se modifica) y **otra operación para borrar el elemento distinguido como punto de interés** (y el punto de interés avanza a la siguiente posición).

4. TAD genérico *cola cíclica de elementos*

Escribe una especificación para el tipo genérico *cola cíclica de elementos*. El tipo define una cola como las habituales colas FIFO, pero con la diferencia de que en lugar de la operación de desencolar el primer elemento de la cola, tiene una operación **penalizar** que quita el elemento que estaba primero y lo coloca al final (tras el último). Las operaciones son: **crear**, **encolar**, **primero** y **penalizar**. Propón una implementación dinámica para el tipo *colaCíclica* de elementos.