

Facultad de Informática
Ingeniería en Informática de Gestión
Estructuras de Datos y de la Información

Grupo A, Mayo 2002

Ejercicios adicionales

Ejercicio 1 Especificar un tipo abstracto de datos *AVERIA* para describir el sistema de reservas en los vuelos de la compañía aérea Avería. La especificación debe incluir al menos las siguientes operaciones:

- *crea* : $\rightarrow averia$, para crear un sistema de reservas vacío, sin información;
- *anade* : $averia\ vuelo \rightarrow averia$, para añadir un nuevo vuelo al sistema de reservas;
- *existe* : $averia\ vuelo \rightarrow bool$, para comprobar si un vuelo existe o no en el sistema;
- *reserva* : $averia\ vuelo\ pasajero \rightarrow averia$, para registrar una reserva para un pasajero en un vuelo dado, con la condición de que un mismo pasajero no puede hacer más de una reserva en un mismo vuelo;
- *lleno* : $averia\ vuelo \rightarrow bool$, para comprobar si un vuelo ya no admite más reservas;
- *asiento* : $averia\ vuelo\ pasajero \rightarrow nat$ para determinar el número de asiento asignado a un pasajero en un vuelo, suponiendo que tal asignación se realiza por orden de reserva empezando por el uno, y devolviendo cero para los pasajeros sin reserva;
- *cuenta* : $averia\ vuelo \rightarrow nat$, para calcular el número de reservas realizadas en un vuelo;
- *cancela* : $averia\ vuelo \rightarrow averia$, para anular un vuelo en el sistema de reservas, al mismo tiempo que se eliminan todas las posibles reservas para ese vuelo;

Los tipos *vuelo* y *pasajero* vienen dados respectivamente por las especificaciones *VUELOS* y *PASAJEROS*, que incluyen la operación *capacidad* : $vuelo \rightarrow nat$ además de operaciones de igualdad. Se suponen asimismo disponibles las especificaciones *BOOL* y *NAT*.

Ejercicio 2 Especificar un tipo abstracto de datos *VIDEOCLUB* que represente la gestión de los alquileres de un videoclub. La especificación debe incluir al menos las siguientes operaciones:

- *crea* para generar un videoclub vacío,
- *an_copia*, para añadir una copia de una película haciendo posible su alquiler,
- *socio* para incluir a una persona como nuevo socio del videoclub, suponiendo que no lo es,

- *alquilar*, para establecer que un socio alquila una copia de una película, suponiendo que hay copias disponibles de la película,
- *devolver*, para establecer que un socio ha devuelto su copia de la película que tenía alquilada,
- *borrar*, para eliminar todas las copias de una película, siempre que hayan sido devueltas,
- *es_socio*, para determinar si una persona es socio del videoclub,
- *num_copias*, para calcular el número de copias existentes de una película, alquiladas o no,
- *num_alqui*, para calcular el número de copias alquiladas de una película,
- *ha_alquilado* para determinar si una persona ha alquilado una copia de una película.

Suponer que se dispone de las especificaciones de *BOOL*, *NAT*, *PELIS* y *GENTE* con tipos y operaciones de igualdad para las películas y las personas, respectivamente.

Ejercicio 3 Tenemos como dato un vector $D[0..n]$ que contiene objetos, cada uno de los cuales consta de tres atributos con información sobre grandes desastres:

- Fecha del desastre,
- Número de víctimas,
- Fecha del último desastre que tuvo más víctimas que el actual.

Los objetos del vector están ordenados cronológicamente, del más antiguo al más reciente, pero el tercer atributo está vacío en todos los objetos. Escribir un programa que complete el vector rellenando el atributo de cada registro a partir de la información que contiene el vector:

- Escribir un programa que no utilice ninguna estructura auxiliar, y tenga un coste en tiempo en $O(n^2)$.
- Escribir un programa que utilice una pila como estructura auxiliar, y tenga un coste en tiempo en $O(n)$.

Ejercicio 4 Dado un número natural $n \geq 2$, se llaman *números supervivientes* a los que resultan de ejecutar el siguiente proceso: se comienza generando una cola que contiene los números desde 1 hasta n , en este orden. Se elimina de la cola un número de cada dos (es decir, los números 1,3,5 etc.); de la nueva cola se elimina ahora un número de cada tres; etc. El proceso termina cuando cuando se va a eliminar un número de cada m y el tamaño de la cola es menor que m . Los números que se queden en la cola en ese momento son los supervivientes. Diseñar un procedimiento que reciba n como parámetro y devuelva una cola formada por los números supervivientes.