



Ejercicio 1.- Extender la especificación PILA[ELEMENTO] del tipo pila visto en clase añadiendo las siguientes operaciones (pueden ser parciales):

- contar: pila \rightarrow natural, para ver cuántos elementos tiene la pila.
- fondo: pila \rightarrow elemento, que consulta el elemento más profundo de la pila.
- montar: pila pila \rightarrow pila, para poner la segunda pila encima de la primera (respetando el orden de los elementos).
- quitar: pila natural \rightarrow pila, que quita tantos elementos de la pila como indica el parámetro natural; por ejemplo, quitar(p, 3) eliminaría tres elementos de la pila.

Ejercicio 2.- (*Examen del Grado en Ingeniería en Informática, Noviembre 2012*) Suponiendo conocida la especificación PILA[ELEMENTO], y suponiendo que el TAD de elemento tiene una operación $_ \leq _$: *elemento elemento* \rightarrow *bool*, que comprueba si un elemento es menor o igual que otro, crear operaciones para:

- contar cuántos elementos hay en una pila.
- obtener la inversa de una pila.
- comprobar si los elementos de una pila fueron introducidos en orden de mayor a menor (el mayor debería estar en el fondo de la pila, y el menor en la cima).
- comprobar si los elementos de una pila fueron introducidos en orden de menor a mayor (el menor debería estar en el fondo de la pila, y el mayor en la cima).
- eliminar el elemento que se encuentre en el fondo de la pila.

Ejercicio 3.- (*Examen del Grado en Ingeniería de Computadores, Noviembre 2012*). Dar la especificación del TAD básico PILA[ENTERO]. Extender dicha especificación con operaciones adicionales para (pueden ser parciales):

- sacar_en_pos: pila natural \rightarrow pila, que elimina el número entero que se encuentra en la posición indicada por el parámetro natural, siendo la posición número uno la cima; por ejemplo, sacar_en_pos(p,2) debería quitar el dato que está justo debajo de la cima de p.



- sacar_entre: pila natural natural \rightarrow pila, que elimina de la pila todos los enteros que se encuentren entre las posiciones indicadas por los parámetros naturales; así, sacar_entre(p, 2, 4) quitaría los elementos que están en las posiciones 2, 3 y 4.

Ejercicio 4.- (*Examen del Grado en Ingeniería Informática, Noviembre 2011*) Se conoce el TAD CONJUNTO[ELEMENTO] para representar los datos *conjunto* de *elemento* con las siguientes operaciones:

- \emptyset : \rightarrow conjunto
- insertar: elemento conjunto \rightarrow conjunto
- borrar: elemento conjunto \rightarrow conjunto
- está?: elemento conjunto \rightarrow bool
- vacío?: conjunto \rightarrow bool

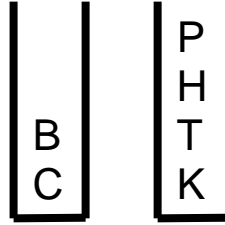
así como la especificación para PILAS[CONJUNTO[ELEMENTO]] (las pilas que están formadas por conjuntos). Añadir operaciones para:

- comprobar si un elemento está en todos los conjuntos de la pila.
- comprobar si todos los conjuntos de la pila tienen, al menos, los mismos elementos que el conjunto de la cima.
- quitar un elemento de todos los conjuntos de la pila.
- crear un único conjunto con todos los elementos de los conjuntos de la pila.
- quitar todos los conjuntos vacíos de la pila.

Ejercicio 5.- Una *secuencia* es una estructura lineal con un punto de interés donde se realizan las inserciones, las modificaciones y las consultas. Se quiere representar las secuencias mediante un par de pilas: la primera pila contiene los datos que están antes del punto del interés, y la segunda pila contiene los datos que están después del punto de interés.



Por ejemplo, la secuencia de letras “C B P H T K”, donde el elemento subrayado representa el punto de interés, se representaría como



Especificar el TAD para el tipo *secuencia*, incluyendo las siguientes operaciones:

- crear una secuencia vacía.
- insertar un elemento en el punto de interés.
- eliminar el elemento que se encuentra en el punto de interés.
- consultar el elemento en el punto de interés.
- avanzar el punto de interés una posición a la derecha.
- trasladar el punto de interés al comienzo de la secuencia.
- determinar si el punto de interés se encuentra al final de la secuencia.
- determinar si la secuencia es vacía.

Ejercicio 6.- Suponiendo disponible `_==_`: elemento elemento \rightarrow bool, que determina si dos datos de tipo elemento son iguales, extender la especificación del tipo cola vista en las clases de teoría con las siguientes operaciones (pueden ser parciales)

- contar: cola \rightarrow natural, para ver cuántos elementos hay en la cola.
- último: cola \rightarrow elemento, que devuelve el dato que está en última posición.
- invertir: cola \rightarrow cola, que da la vuelta a los elementos de una cola.
- iguales?: cola cola \rightarrow bool, que comprueba si dos colas son iguales (mismos elementos y en las mismas posiciones).
- simétrica?: cola \rightarrow bool, para ver si la cola tiene los mismos datos en los dos sentidos (de primero a último y viceversa).



Ejercicio 7.- (*Examen del Grado en Ingeniería Informática, Noviembre 2010*) Especificar el TAD colas de caracteres (se tienen las generadoras constantes para todas las letras del alfabeto y también está disponible una operación de orden para ver si una letra es anterior a otra \leq : `caracter caracter \rightarrow bool`, pero el resto de las posibles operaciones auxiliares hay que especificarlas), añadiendo operaciones:

- concatenar dos colas de caracteres.
- mezclar alternativamente los elementos de dos colas de caracteres (no tienen porqué tener necesariamente la misma longitud).
- quitar la primera mitad (redondeando la cantidad a la baja si es necesario) de la cola.
- comprobar si la cola está ordenada alfabéticamente.
- ver si la cola representa una palabra, entendiendo por palabra una sucesión de caracteres que no tiene dos vocales o dos consonantes seguidas.

Ejercicio 8.- (*Examen del Grado en Sistemas de Información, Noviembre 2012*) Usando las especificaciones `BOOLEANOS` y `COLA[BOOLEANOS]`, crear operaciones:

- contar cuántos elementos están a `TRUE` en la cola,
- eliminar los elementos `FALSE` que se encuentren al comienzo de la cola,
- eliminar todos los elementos `FALSE` que se encuentren en la cola,
- cambiar de valor todos los elementos de la cola,
- obtener la disyunción exclusiva (operación `XOR`), y
- conseguir el valor lógico resultante de evaluar una cola mediante el operador `XOR`.

Nota: Se utilizará la operación <i>eq</i> para ver si dos elementos son iguales.



Ejercicio 9.- (*Examen del Grado en Ingeniería de Computadores, Noviembre 2012*) Teniendo disponible la especificación de ENTERO y sus operaciones de orden, dar la especificación del TAD básico COLA[ENTERO]. Extender dicha especificación con las siguientes operaciones adicionales (pueden ser parciales):

- haymáspositivos?: cola \rightarrow bool, que determina si en la cola hay más números positivos que negativos;
- quitardos: cola \rightarrow cola, que quita un entero positivo y un entero negativo de la cola (no necesariamente seguidos).

Ejercicio 10.- (*Examen del Grado en Ingeniería Informática, Noviembre 2011*) Dar la especificación del TAD COLA[ELEMENTO]. Ampliar dicha especificación creando un tipo nuevo *doblecola* para la gestión de dos colas a la vez, que permita las operaciones:

- descomponer: cola doblecola \rightarrow , que pone los elementos que hay en una cola repartiéndolos de manera alterna en una doblecola; y
- mezclar: doblecola \rightarrow cola, que mezcla alternativamente los elementos de una doblecola en una única cola.

Ejercicio 11.- Un sistema operativo recibe mensajes de los dispositivos conectados al sistema. Dichos mensajes se dividen en dos categorías: críticos y no críticos. El sistema operativo atiende primero a los mensajes críticos por orden de llegada (si los hubiera) y, después, a los no críticos también por orden de llegada.

Se puede suponer diseñado el TAD para los datos de tipo *mensaje*, incluyendo una operación que indique si un mensaje es crítico o no. Especificar un TAD para el tipo *sistema* que gestione la cola de mensajes con el comportamiento descrito, y que disponga al menos de las siguientes operaciones:

- crear un sistema vacío.
- añadir un nuevo mensaje al sistema.
- consultar el primer mensaje que haya que atender.
- consultar el número de mensajes críticos en el sistema.
- consultar el número de mensajes no críticos en el sistema.
- consultar si el sistema está vacío.



Ejercicio 12.- (*Examen del Grado en Ingeniería de Computadores, Noviembre 2011*) Suponiendo que están disponibles las especificaciones de COLA[ELEMENTO] Y PILA[ELEMENTO], se quiere crear un tipo nuevo *colapila* para la gestión de una cola y una pila a la vez, de manera que todas las inserciones y eliminaciones de elementos de una *colapila* se hagan en su parte de cola. Crear un TAD para *colapila* con operaciones para

- crear una *colapila*.
- comprobar si una *colapila* tiene algún elemento.
- añadir un elemento a una *colapila*.
- quitar un elemento de una *colapila*.
- pasar toda la parte de cola de la *colapila* a su parte de pila.
- pasar toda la parte de pila de la *colapila* a su parte de cola.
- vaciar completamente una *colapila*.