

Ejercicio 1.- Se quiere diseñar el TAD fecha, que contiene día, mes y año (en ese orden) como número naturales y que debe estar formado por las siguientes operaciones:

- $/_/__/$: natural natural natural \rightarrow fecha
- Esanterior?: fecha fecha \rightarrow bool, determina si una fecha es anterior a otra.
- Diasiguiente: fecha \rightarrow fecha, devuelve la fecha resultante de aumentar la fecha en un día.
- Distancia: fecha fecha \rightarrow natural, calcula cuantos días de diferencia hay entre dos fechas.
- Pasardias: fecha natural \rightarrow fecha, devuelve la fecha resultante tras aumentar fecha en un número de días.
- Coincidentes?: fecha fecha \rightarrow bool, determina si dos fechas caen en el mismo día de la semana (lunes, martes, ..).

Especificar el TAD fecha, incluyendo posibles operaciones auxiliares. No es necesario considerar los años bisiestos, y puede utilizarse libremente cualquier constante natural si necesidad de definirla.

Ejercicio 2.- Extender la especificación BOOLEANOS del tipo *bool* añadiendo las operaciones lógicas de implicación, equivalencia y disyunción exclusiva.

Ejercicio 3.- Diseñar un Tipo Abstracto de Datos para trabajar con conjuntos de elementos (el tipo exacto de los elementos no se conoce, así que será un *parámetro formal* y puede suponerse una operación $_==_$: elemento elemento \rightarrow bool), considerando que en un conjunto cada elemento aparece un sola vez. El TAD deberá tener las siguientes operaciones:

- es_vacio?: conjunto \rightarrow bool, decide si un conjunto está vacío.
- coger: conjunto \rightarrow elemento, devuelve uno de los elementos del conjunto.
- insertar: elemento conjunto \rightarrow conjunto, para añadir un elemento a un conjunto (si se inserta un elemento a un conjunto que ya lo contiene no vuelve a ponerse);
- borrar: elemento conjunto \rightarrow conjunto, para quitar un elemento de un conjunto (si se quiere quitar un elemento que no está, el conjunto no debe cambiar);
- $_ \cap _$: conjunto conjunto \rightarrow conjunto, hace la intersección de dos conjuntos;
- $_ \cup _$: conjunto conjunto \rightarrow conjunto, para hacer la unión de dos conjuntos;
- cardinal: conjunto \rightarrow natural, que cuenta cuántos elementos tiene un conjunto;
- está?: elemento conjunto \rightarrow bool, para comprobar si un elemento ya pertenece a un conjunto o no.

Para escribir la especificación en pseudocódigo de las operaciones indicadas se supondrá que ya disponemos de las operaciones que dependen directamente de la implementación: es_vacio, coger un elemento del conjunto, insertar y borrar un elemento.

Ejercicio 4.- Incluir en el TAD conjunto las operaciones:

- Restar dos conjuntos.
- Es_subconjunto.

Ejercicio 5.-Especificar el TAD multiconjunto finito sobre un tipo de elemento dado como parámetro. Debe incluir las siguientes operaciones:

- Crear multiconjunto vacío
- Añadir un elemento
- Calcular la multiplicidad de un elemento
- Predicado que decide si un mult es vacío
- Quitar una aparición de un elemento
- Borrar todas las apariciones de un elemento
- Unión de mult
- Intersección
- Diferencia
- Cardinal, contando todos
- Cardinal, contando solo elementos diferentes
- Partimos de la idea de conjunto, hace falta un elemento con operación de igualdad.

Ejercicio 6.-Especificar las cadenas finitas sobre un alfabeto dado como parámetro. Operaciones:

- Cadena vacía
- Añadir un elto por la izq o por la derecha
- Generar una cadena unitaria formada por un elto dado
- Concatenar
- Longitud
- Consultar el elto más a la izq.
- Consultar más a la derecha
- Eliminar el elto más a la derecha y más a la izquierda
- Decidir si es vacía
- Decidir si elto pertenece a cadena
- Si dos cad son iguales
- Inversa de una cad

Para escribir la especificación en pseudocódigo de las operaciones indicadas se supondrá que ya disponemos de las operaciones que dependen directamente de la implementación: Cadena Vacía, insertar por la izquierda, eliminar primero, ver primero y decidir si es vacía.

Ejercicio 7.- Incluir en el TAD del ejercicio 6 las operaciones:

- rotar_izq $(a_1, a_2, \dots, a_n) \rightarrow (a_2, a_3, \dots, a_n, a_1)$
- rotar_der $(a_1, a_2, \dots, a_n) \rightarrow (a_n, a_1, a_2, \dots, a_{n-1})$
- L_esimo $((a_1, a_2, \dots, a_n), i) \rightarrow a_i$
- Medio $((a_1, a_2, \dots, a_n)) \rightarrow a_{(n+1)/2}$

Ejercicio 8.-Una caja de seguridad tiene un cerrojo formado por tres ruedas, cada una de ellas con los dígitos 0 a 9, de forma que las ruedas pueden avanzar su valor de manera independiente sin afectar a las otras. La caja solo se puede abrir si los valores de las tres ruedas coinciden con su código secreto.

Suponiendo disponible la especificación de los naturales ampliada con la multiplicación de naturales (operación $_*$: natural natural \rightarrow natural), se piden los sisguinets TAD's y operaciones:

- Especificar el TAD rueda con las operaciones necesarias y diseñar una representación para el TAD cerrojo.
- avanzar1: cerrojo \rightarrow cerrojo, avanzar2 avanzar3, que pasan al siguiente valor de alguna de las ruedas deol cerrojo (primera, segunda, tercera respectivamente)
- valor_cerrojo:cerrojo \rightarrow natural, obtien el número formado por los dígitos de las tres ruedas del cerrojo.
- cambiar:cerrojo natural \rightarrow cerrojo que gira las ruedas hasta que el valor del cerrojo coincida con natural