

PRÁCTICA 14

TIPOS ABSTRACTOS DE DATOS

Nota: Para todos los ejercicios además de las operaciones solicitadas, se deben incluir las operaciones de creación, asignación, consulta y modificación de atributos.

1.- a) Defina e implemente un TAD triángulo con los siguientes atributos: base, altura y ángulos internos. Dicho TAD debe contener las operaciones: *CalcularSuperficie* y *CantidadDeLadosIguales*.

b) Utilizando el TAD definido en a) realice un programa que genere 10 triángulos (leyendo los datos necesarios). Luego de la lectura de todos los datos, informe la cantidad de triángulos equiláteros, isósceles y escalenos encontrados.

2.- Defina e implemente un TAD Persona con los atributos: Nombre, DNI, ocupación, altura y peso. Implemente un programa que utilizando el TAD Persona:

- a) Almacene en un vector 45 personas.
- b) Luego de almacenar la información:
 - i. Informe todos los datos de la persona con mayor peso.
 - ii. Informe los nombres de las personas cuya altura no supera el 1.70 metros.
 - iii. Modifique la ocupación de la persona con DNI 32000512 a *“empleado de comercio”*.

3.- a) Implemente el TAD ListaDeEnteros. El TAD debe contener al menos las siguientes operaciones: crear una lista vacía, agregar un elemento adelante, agregar un elemento al final, borrar un elemento, devolver la cantidad de elementos, y operaciones que permitan recorrer la lista (iniciar: prepara la lista para recorrerla, avanzar: avanza a la posición siguiente, fin: devuelve si se llegó al final de la lista, elemento: devuelve el elemento actual donde se está posicionado).

b) Utilizando el TAD ListaDeEnteros, realice un programa que genere una lista con números ingresados de teclado (la lectura finaliza al ingresar el 0). Luego de cargar la información informe el contenido de la lista.

4.- Construir el TAD árbol binario de búsqueda de enteros, que además de las operaciones básicas, debe incluir las siguientes operaciones:

- a) calcular la altura de un árbol binario.
- b) calcular el número de nodos de un árbol binario.
- c) hallar el valor máximo del árbol.
- d) hallar el valor mínimo del árbol.

5.- a) Defina e implemente el TAD micro de 20 asientos (numerados de 1 a 20). Implemente las siguientes operaciones:

Crear un micro	<i>[crea un micro vacío para un destino dado].</i>
Ver destino	<i>[retorna el destino del micro]</i>
Ocupar un asiento específico	<i>[dado un número de asiento el mismo se ocupa].</i>
Cantidad de asientos libres	<i>[retorna la cantidad de asientos no ocupados].</i>
Liberar un asiento específico	<i>[dado un nro. de asiento el mismo se libera].</i>
Asiento ocupado	<i>[dado un nro. de asiento, retorna Verdadero en caso que el mismo esté ocupado, Falso en caso contrario].</i>
Ver asiento libre	<i>[retorna un número de asiento libre].</i>
Asiento válido	<i>[dado un número de asiento retorna verdadero si el mismo pertenece al micro, falso en caso contrario]</i>
Hay asiento libre	<i>[retorna verdadero en caso que exista al menos un asiento libre, falso en caso contrario].</i>

b) Una empresa de transporte de pasajeros de larga distancia, dispone de una estructura con la información de su flota de micros (tipo exportado del TAD), la misma está ordenada por destino y debe ser eficiente para la búsqueda por dicho criterio. *Suponga que hay un único micro por destino.*

Realice un programa que lea de teclado los datos: nombre de persona y destino deseado (la lectura finaliza cuando se ingresa el nombre 'ZZZ'). A cada persona se le debe informar, en caso que la empresa tenga un micro que va al destino deseado y disponga de asiento libre en el micro, el número de asiento asignado.

6.- a) Defina el TAD teatro, donde un teatro está formado por 30 filas de 40 butacas cada una. Implemente las operaciones de crear el teatro, dada una fila determinar si existe una butaca libre, dada una fila retornar el número de la primera butaca libre, dado un número de fila y butaca venderla, dado un número de fila y butaca liberarla.

b) Utilizando el TAD definido en a) crear un teatro vacío. Además se tiene una lista de personas quienes indican si quieren comprar o devolver una localidad. Cuando una persona quiere comprar una localidad debe indicar la fila que le interesa, en caso que dicha fila no tenga una butaca disponible, no se efectúa la operación. En cambio cuando se realiza una devolución la persona indica número de fila y butaca.

Nota: cada persona requiere el pedido de una sola localidad.

7.- Dado el siguiente TAD viaje el cual NO DEBE IMPLEMENTAR.

TAD Tviaje

interface

Type exportado viaje;

procedure crearUnViaje(**var** v: viaje; códigoAuto: integer; km: integer; dir1, dir2: string);

{Crea un viaje "v" realizado por el auto "códigoAuto", con distancia recorrida "km", dirección de origen "dir1" y dirección destino "dir2"}

function verCodigoAuto(v: viaje): integer;

{Devuelve el código del auto que realizó el viaje "v"}

function verKilometros(v: viaje): integer;

{Devuelve la cantidad de kilómetros que realizó el auto en el viaje "v"}

procedure asignarViaje(**var** v1: viaje; v2: viaje);

{Asigna el viaje "v2" al viaje "v1"}

Utilizando el TAD *Tviaje* realice el siguiente programa:

Una remisería dispone una lista con la información de los viajes (tipo exportado del TAD) realizados durante el mes de septiembre. Esta información se encuentra ordenada por código de auto y para un mismo código de auto pueden existir 1 o más viajes.

- a) Realizar un módulo que reciba la lista con la información de los viajes y genere una nueva estructura con la siguiente información: código de auto, total de kilometraje recorrido y cantidad de viajes realizados por el auto. Esta estructura debe estar ordenada por kilometraje recorrido y debe ser eficiente para la búsqueda por este criterio de ordenación. (Tener en cuenta, que para cada auto el total de kilómetros recorrido se calcula totalizando el kilometraje de cada viaje que realiza el auto).
- b) Luego de generada la estructura en el punto anterior y a partir de las estructuras anteriores utilice la más adecuada para calcular e informar:
 - I. Cantidad de autos con cantidad total de kilómetros recorridos entre 50.000 y 100.000.
 - II. El auto con más viajes realizados.

NOTA: Modularizar. Declarar todas las estructuras de datos utilizadas. Realizar un programa que simule el llamado a los módulos realizados.