

CC30A - Control 1- viernes 26 de septiembre de 2003

Tiempo: 1 hr 45 minutos – CON Apuntes – SIN consultas

1. El lenguaje Java provee el sgte TDA Vector para manejar listas de valores v_0, v_1, \dots, v_{n-1} :

operación	significado
<code>new Vector()</code>	crea vector sin valores
<code>void addElement(Object x)</code>	agrega x al final del vector
<code>int size()</code>	devuelve N° de valores del vector
<code>Object clone()</code>	devuelve una copia del vector
<code>boolean contains(Object x)</code>	true si x está en el vector (false si no)
<code>Object elementAt(int i)</code>	valor de índice i (el 1° tiene índice 0)
<code>Object firstElement()</code>	primer valor
<code>Object lastElement()</code>	último valor
<code>int indexOf(Object x)</code>	índice de 1ª ocurrencia de x (-1 si no está)
<code>int indexOf(Object x, int i)</code>	índice de 1ª aparición de x. Búsqueda comienza en índice i
<code>void insertAt(Object x, int i)</code>	inserta x en índice i
<code>void removeAt(int i)</code>	eliminar valor de índice i
<code>void setAt(Object x, int i)</code>	reemplaza valor de índice i por x
<code>boolean remove(Object x)</code>	elimina 1ª aparición de x. Devuelve true si tiene éxito

A) Escriba un método de encabezamiento `Vector reemplazar(Vector x, Object y, Object z)` que devuelva un nuevo Vector en que se han reemplazado todas las apariciones del valor y por el valor z en el vector x (sin modificar el vector x).

Por ej, si $v = A, B, C, A, D, A$ entonces `reemplazar(v, "A", "a")` devuelve el vector a, B, C, a, D, a

B) Suponiendo que un vector se representa por un arreglo **a** que contiene **n** elementos, escriba el método **addElement**. En caso que el arreglo **a** esté lleno (situación que debe detectarse atrapando la excepción correspondiente), debe crearse otro con un elemento más.

C) Suponiendo que un vector se representa con una lista enlazada, escriba el método **clone**. de modo que construya el nuevo vector en una pasada (tiempo n).

2. La siguiente clase permite realizar algunas operaciones con un árbol binario:

```
class AB{
    protected Nodo raiz;
    public AB(){raiz=null;}
    public AB copiar(){...} //entrega una copia del árbol
    public boolean existe(Object x){...} //entrega true si el valor x está en el árbol
    ...
}
class Nodo{
    public Object valor; public Nodo izq, der;
    public Nodo(Object x, Nodo y, Nodo z){ valor=x; izq=y; der=z; }
}
```

A) escribir el método copiar

B) escribir el método existe, buscando concurrente en el subárbol izquierdo y derecho.

CC30A-2003- Pauta Pregunta 1

A) 2 ptos

```
static public Vector reemplazar(Vector x, Object y, Object z){  
    Vector aux=(Vector)x.clone();           //0.75  
    int i=0;  
    while( (i=aux.indexOf(y)) >= 0 )        //0.5  
        aux.setElementAt(z,i);              //0.5  
    return aux;                             //0.25  
}
```

B) 2 ptos

```
public void addElement(Object x){  
    try{  
        a[n]=x; ++n;                        //0.25  
    }catch(RuntimeException e){            //0.25  
        Object[]b=new Object[++n];         //0.5  
        for(int i=0;i<n-1;++i) b[i]=a[i];   //0.5  
        b[n-1]=x;                          //0.25  
        a=b;                               //0.25  
    }  
}
```

C) 2 ptos

```
public Object clone(){  
    Vector aux = new Vector();              //0.25  
    if(primero==null) return aux;           //0.25  
    Nodo ant = aux.primero = new Nodo(primero.valor,null); //0.5  
    for(Nodo r=primero.sgte; r!=null; r=r.sgte) //0.25  
        ant = ant.sgte = new Nodo(r.valor,null); //0.5  
    return aux;                             //0.25  
}
```

Pauta Pregunta 2

A) 3 ptos

```
public AB copia(){
    AB aux = new AB();           //0.25
    AB.raiz = copia(raiz);       //0.5
    return AB;                   //0.25
}
protected Nodo copia(Nodo r){ //0.25
    if(r == null ) return null; //0.25
    return new Nodo(             //0.25
        r.valor,                 //0.25
        copia(r.izq),           //0.5
        copia(r.der) );         //0.5
}
```

B) 3 ptos

```
//1 pto
public boolean existe(Object x){
    if( raiz==null ) return false;           //0.1
    if(raiz.valor.equals(x)) return true;     //0.2
    T a=new T(x,raiz.izq), b=new T(x,raiz.der); //0.2
    a.start(); b.start(); a.join(); b.join(); //0.2
    return a.resultado() || b.resultado();    //0.3
}
//1 pto
class T extends Thread{                     //0.1
    private Nodo raiz;                       //0.1
    private boolean b;                       //0.1
    private Object x;                        //0.1
    public T(Object x,Nodo r){ this.x=x; raiz=r;} //0.2
    public void run(){b=existe(x,raiz);}      //0.2
}
public boolean resultado(){ return b; }      //0.2
//1 pto
protected boolean existe(Object x,Nodo r){   //0.2
    if(r==null) return false;                //0.2
    if(r.valor.equals(x)) return true;        //0.2
    return existe(x,r.izq) || existe(x,r.der); //0.4
}
}
```