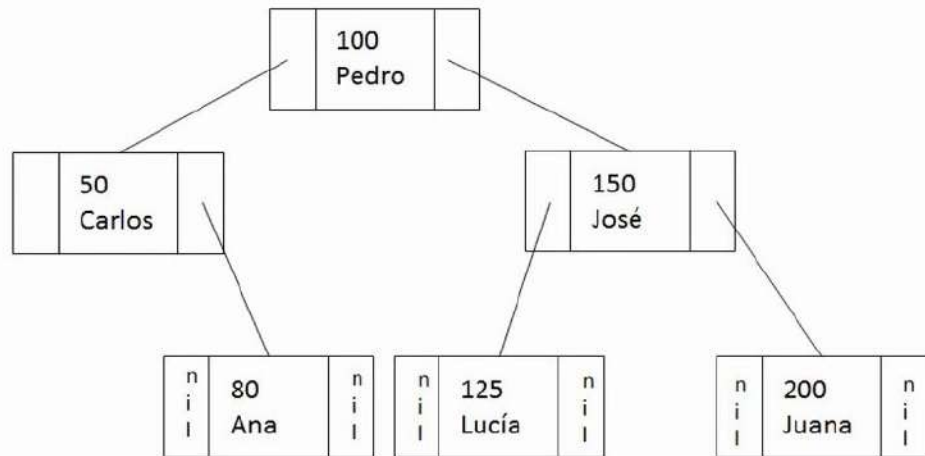


Teórico - Módulo Imperativo

1. Se lee código, tema, descripción y fecha de inicio de los trámites de un ministerio. La lectura se realiza ordenada por código de trámite. Se debe almacenar la información de los trámites de forma tal de optimizar tanto la cantidad de operaciones para buscar un trámite en particular (por código de trámite) como así también la cantidad de memoria dinámica utilizada.

¿Almacenaría la información en una lista o en un árbol binario de búsqueda? Justificar su respuesta.

Dado el siguiente árbol binario de búsqueda ordenado por código, indicar el resultado de imprimir los nombres utilizando un recorrido Post-Orden.



Teórico - Módulo Imperativo

Escribir un enunciado de un problema que su solución pueda plantearse con un algoritmo recursivo con al menos 2 casos base. Implementar la solución. Justificar

Teórico - Módulo Imperativo

Dado el procedure recorrido, que procesa un ABB de enteros.

| | |
|---|--|
| Type arbol= ^nodo; nodo = record dato: integer; HI: arbol; HD: arbol; end; | procedure recorrido(a:arbol); begin if (a <> nil) then begin recorrido(a^.HI); recorrido(a^.HD); write(a^.dato, ' '); end; end; |
|---|--|

Indique para cada afirmación si es Verdadera o Falsa y Justifique.

- a - El procedure hace un recorrido pre-orden.
- b - El procedure no tiene caso base.
- c - El procedure imprime los datos en orden ascendente.

- 1) **Módulo imperativo.** Se posee un ABB de números enteros con cinco nodos y el resultado de imprimir los datos en post-orden es "1 2 3 4 5". Dibuje dicho árbol.
- 2) **Módulo POO.** ¿El siguiente programa compila? ¿Qué imprime? Justifique su respuesta.

| | | |
|---|--|---|
| <pre>public class A { private int a = 5; public A(int aa){ a = aa; } public int uno(){ return this.dos(); } public int dos(){ return 3; } }</pre> | <pre>public class B extends A { private int b = 7; public B(int bb){ super(bb); } public int uno(){ return super.uno(); } public int dos(){ return this.dos(); } }</pre> | <pre>public class Program{ public static void main(String args[]){ A a = new B(1); System.out.println(a.uno()); } }</pre> |
|---|--|---|

Teórico - Módulo Objetos

Indique todos los errores en el siguiente código. Justifique su respuesta. Suponga que las clases tienen los métodos setters implementados.

```
public abstract class Persona {
    private String nombre;
    private int dni;

    public Persona (String unNombre, int unDNI){
        setNombre(unNombre);
        setDNI(unDNI);
    }
    ...
}

public class Empleado extends Persona {
    private int nroSucursal;

    public Empleado (String unNombre, int unDNI, int unNroSucursal){
        setNombre(unNombre);
        setDNI(unDNI);
        setNroSucursal(unNroSucursal);
    }
    ...
}
```

```
public class Cliente extends Persona {
    private String domicilioEnvio;

    public Cliente (String unNombre, int unDNI, String unDomicilio){
        setNombre(unNombre);
        setDNI(unDNI);
        setDomicilio(unDomicilio);
    }
    ...
}

public class Main {
    public static void main(String[] args) {
        Empleado empl = new Empleado ("Raúl", 28114557, 3);
        Cliente cli = new Cliente ("Esteban", 20441774, "Calle 44 nro 481");
        Persona pers = new Persona ("Laura", 37983612);
        ...
    }
}
```

Teórico - Módulo Objetos

Dado el siguiente código

| | |
|---|--|
| <pre>public class B { private int x=1; private int y=2; ... }</pre> | <pre>public class A extends B { private int z=3; ... }</pre> |
|---|--|

Indique para cada afirmación si es Verdadera o Falsa y Justifique.

- a - La clase B puede declarar un método abstracto.
- b - Un objeto instancia de la clase A sólo almacena la variable de instancia "z"
- c- La clase B debe obligatoriamente declararse abstracta por ser la clase padre.

Teórico - Módulo Objetos

-- 1 -- Si una clase B hereda de otra clase A. ¿Cuál es la opción correcta para crear una instancia de la clase B?

- a. `B var = new A();`
- b. `B var = new B();`
- c. `A var = new A();`
- d. `A var = new B(A);`

Teórico - Módulo Objetos

-- 2 -- ¿Cuál es el objetivo del constructor de una clase?

- a. Heredar el estado y comportamiento de la superclase
- b. Definir el estado y el comportamiento del objeto
- c. Inicializar el estado interno del objeto.
- d. Inicializar el comportamiento interno del objeto

Teórico - Módulo Objetos

-- 3 -- ¿A qué se denomina herencia?

- a. A la capacidad de que una clase pueda utilizar el estado y comportamiento definido en su superclase.
- b. A la capacidad de que una clase pueda utilizar el estado y comportamiento definido en su subclase.
- c. A la capacidad de que una clase defina métodos abstractos para que sean implementados por sus subclases.
- d. A la capacidad de que una clase defina, como parte de su estado, atributos

Teórico - Módulo Objetos

-- 4 -- ¿A qué se denomina método abstracto?

- a. A los constructores definidos en una superclase.
- b. A los métodos sin código definidos en una superclase.
- c. A los métodos con código definidos en una superclase.
- d. A cualquier método o constructor definido en una clase abstracta.

Teórico - Módulo Objetos

-- 5 -- ¿A qué se le denomina Instancia de una clase?

- a. A la capacidad de que una clase herede el estado y comportamiento de otra clase.
- b. Al objeto que posee el estado y comportamiento declarado en la clase instanciada.
- c. Al conjunto de palabras reservadas del lenguaje para definir variables y funciones públicas y privadas
- d. Al estado privado definido en la clase

Módulo Concurrente

Escriba el enunciado de un ejercicio (al estilo de los enunciados de la práctica) en el cual en la resolución del mismo sería adecuado utilizar un área parcialmente compartida. JUSTIFIQUE la elección del enunciado

Teórico - Módulo Concurrente

Si se desea que x robots compitan para tomar de a una todas las flores de la esquina 10,10. ¿Cuál es la opción correcta de código que debe utilizarse para lograr el objetivo planteado? Justifique.

| | | |
|---|---|---|
| <pre>robot tipoRobotA ... av:= PosAv ca:= PosCa ok:= V mientras (ok) BloquearEsquina(10,10) Pos(10,10) si(HFELE) tomarFlor sino ok:= F Pos(av,ca) LiberarEsquina(10,10) ... fin</pre> | <pre>robot tipoRobotB ... av:= PosAv ca:= PosCa ok:= V mientras (ok) Pos(10,10) BloquearEsquina(10,10) si(HFELE) tomarFlor sino ok:= F Pos(av,ca) LiberarEsquina(10,10) ... fin</pre> | <pre>robot tipoRobotC ... av:= PosAv ca:= PosCa ok:= V mientras (ok) BloquearEsquina(10,10) Pos(10,10) si(HFELE) tomarFlor sino ok:= F LiberarEsquina(10,10) Pos(av,ca) ... fin</pre> |
|---|---|---|