



Universidad
Carlos III de Madrid

Programación de Sistemas

Grados en Ingeniería de Sistemas Audiovisuales, Ingeniería de Sistemas de Comunicaciones, Ingeniería en Sistemas de Telecomunicación e Ingeniería Telemática

Leganés, 3 de julio de 2013
Duración de la prueba: 45 min.

Examen final. Convocatoria ordinaria. Teoría.
Puntuación: 5 puntos sobre 10 del examen

Sólo una opción es correcta en cada pregunta. Cada respuesta correcta suma $\frac{1}{4}$ puntos. Cada respuesta incorrecta resta $\frac{1}{12}$ puntos. Las preguntas no contestadas no suman ni restan puntos.

- No se permite el uso de libros o apuntes, ni tener teléfonos móviles u otros dispositivos electrónicos encendidos. Incumplir alguna de estas normas puede ser motivo de expulsión inmediata del examen.
- Marca la respuesta a cada pregunta con una equis (“X”) en la tabla de abajo.
- Si marcas más de una opción, o ninguna opción, la pregunta se cuenta como no contestada (ni suma ni resta).
- Rellena **tus datos personales** antes de comenzar a realizar el examen.

Modelo: A

Nombre:	Grupo:
---------	--------

Firma:

NIA:

--	--	--	--	--	--	--	--	--	--

	A	B	C	D		A	B	C	D
1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	12	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	13	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	14	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	15	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	16	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	17	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	18	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	19	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

■ ■ ■ ■ ■

■ ■ ■ ■

- 1.- Un montículo es una estructura de datos apropiada para:
 - (a) Recuperar de forma eficiente un dato dada su clave.
 - (b) Insertar de forma eficiente un nodo en una lista enlazada.
 - (c) *** Recuperar de forma eficiente el dato con clave menor.
 - (d) Implementar de forma eficiente pilas y colas.

- 2.- Si invocamos el método *add* sobre el *ContentPane* de una ventana, estaremos:
 - (a) *** Añadiendo componentes gráficos a la ventana.
 - (b) Asociando escuchadores al *ContentPane*.
 - (c) Insertando eventos en un panel.
 - (d) Mostrando *layouts* en el programa.

- 3.- El coste de acceso, en términos de tiempo de cómputo, a posiciones intermedias en arrays y en listas:
 - (a) En ambos casos es independiente del tamaño.
 - (b) En ambos casos es dependiente del tamaño.
 - (c) Es independiente del tamaño de la lista pero dependiente del tamaño del array.
 - (d) *** Es independiente del tamaño del array pero dependiente del tamaño de la lista.

- 4.- Dado el siguiente código, se puede afirmar que:

```
Something x;
```

 - (a) *Something* puede ser una clase.
 - (b) *** Todas las demás opciones son correctas.
 - (c) *Something* puede ser una clase abstracta.
 - (d) *Something* puede ser una interfaz.

- 5.- Al recorrer una lista enlazada, ¿cómo se sabe que se ha llegado al último nodo?
 - (a) Depende de si la lista está llena o no.
 - (b) Cuando el primer nodo de la lista coincide con el último.
 - (c) Cuando el nodo actual es null.
 - (d) *** Cuando el siguiente del nodo actual es null.

- 6.- Un método recursivo es aquel que se llama a sí mismo...
 - (a) ... de manera lineal.
 - (b) *** ... directa o indirectamente.
 - (c) ... con una estructura en cascada.
 - (d) ... sin pasar argumentos en la llamada.

7.- Para implementar pilas y colas de forma eficiente con listas:

- (a) *** Para la cola se necesita mantener dos referencias y para la pila se necesita mantener una.
- (b) Para la cola se necesita mantener dos referencias y para la pila se necesita mantener dos también.
- (c) Para la cola se necesita mantener una referencia y para la pila se necesita mantener una también.
- (d) Para la cola se necesita mantener una referencia y para la pila se necesita mantener dos.

8.- Si el método A invoca al método B y este a su vez invoca al método A, estamos hablando de un caso de:

- (a) Cola de ejecución.
- (b) Estructura de datos lineal.
- (c) Métodos no recursivos.
- (d) *** Recursión mutua.

9.- Dadas dos clases A y B tales que *class B extends A*, selecciona cuál de las siguientes instrucciones y declaraciones es incorrecta o incompatible con este contexto:

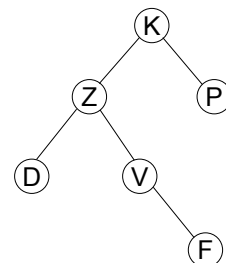
- (a) class C extends B ...
- (b) abstract class A ...
- (c) *** Ninguna de las otras instrucciones es incorrecta o incompatible.
- (d) A ref = new B();

10.- Selecciona la opción *falsa*:

- (a) *** Todos los métodos en una clase abstracta tienen que ser abstractos.
- (b) Una clase abstracta puede tener constructores.
- (c) Ni las clases abstractas ni las interfaces pueden instanciarse.
- (d) Una interfaz Java declara los métodos sin implementarlos.

11.- Un algoritmo recorre el árbol que se muestra con el siguiente orden: K, Z, D, V, F, P. Se trata de un recorrido:

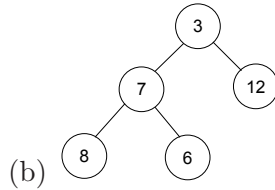
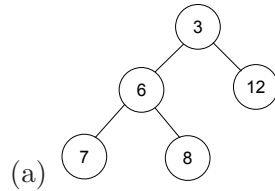
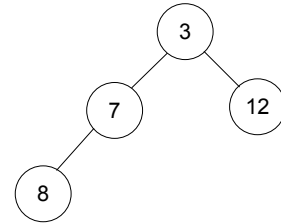
- (a) Inorden.
- (b) Gaussiano.
- (c) Postorden.
- (d) *** Preorden.



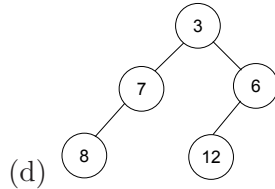
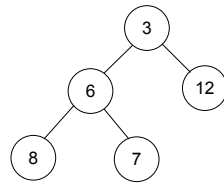
12.- Sea x un método recursivo cuyo código incluye la llamada recursiva $x(n - 1, x(n, m - 1))$. ¿Qué tipo de recursión es?

- (a) Lineal por la cola.
- (b) En cascada.
- (c) *** Anidada.
- (d) Mutua.

13.- El resultado de insertar un nodo con clave 6 en el montículo que se muestra a la derecha es:



(c) ***



14.- El uso de memoria contigua:

- (a) Lo requieren tanto los arrays como las listas.
- (b) No lo requieren ni los arrays ni las listas.
- (c) Lo requieren las listas pero no los arrays.
- (d) *** Lo requieren los arrays pero no las listas.

15.- Para implementar pilas y colas con deque (colas dobles):

- (a) No se pueden implementar pilas y colas con deque, son las deque las que se implementan con pilas y colas.
- (b) No se pueden implementar pilas y colas con deque ni viceversa, es decir, tampoco se pueden implementar deque con colas y pilas.
- (c) Para la cola los datos se extraen por el mismo extremo por donde se insertan y para la pila se extraen por el extremo opuesto.

- (d) *** Para la pila los datos se extraen por el mismo extremo por donde se insertan y para la cola se extraen por el extremo opuesto.

16.- Dado un árbol binario tal y como se ha implementado en clase, el método que se muestra, perteneciente a la clase *BNode*:

- (a) Devuelve la profundidad del nodo al cual pertenece el método.
(b) Devuelve el tamaño del árbol cuya raíz es el nodo al cual pertenece el método.
(c) *** Devuelve la altura del árbol cuya raíz es el nodo al cual pertenece el método.
(d) Realiza un recorrido inorden del árbol cuya raíz es el nodo al cual pertenece el método.

```
public int method() {
    int x = -1;
    if (left != null) {
        x = left.method();
    }
    if (right != null) {
        int y = right.method();
        if (y > x) {
            x = y;
        }
    }
    return x + 1;
}
```

17.- Según el modelo de eventos de Java:

- (a) Los escuchadores se encuentran en el paquete `javax.swing`.
(b) *** Las acciones del usuario sobre la interfaz gráfica son notificadas a los escuchadores.
(c) Todo componente gráfico en la interfaz ha de ser asociado a un escuchador.
(d) Cada escuchador ha de asociarse a un único componente gráfico.

18.- Selecciona la afirmación correcta respecto a programación orientada a objetos en Java:

- (a) *** La *herencia* modela la relación *ES-UN*, en la cual los objetos de la clase hija también son objetos de la clase padre.
(b) El *polimorfismo* es un mecanismo que permite tener varios métodos con el mismo nombre en la misma clase.
(c) Java define un mecanismo de *herencia múltiple* que permite reutilizar métodos de varias clases distintas.
(d) El estado de un objeto se puede recuperar con el método *getState*.

19.- Dado el siguiente código, indica qué se imprimirá:

- (a) anInt=5; aNumber value=-2
- (b) anInt=5; aNumber value=2
- (c) *** anInt=-5; aNumber value=2
- (d) anInt=-5; aNumber value=-2

```
class Number {
    int value;
    Number(int value) {
        this.value = value;
    }
    static void toPositive(int anInt) {
        if (anInt < 0) {
            anInt = - anInt;
        }
    }
    static void toPositive(Number aNumber) {
        if (a.value < 0) {
            a.value = - a.value;
        }
    }
    public static void main(String[] args) {
        int anInt = -5;
        int aNumber = new Number(-2);
        toPositive(anInt);
        toPositive(aNumber)
        System.out.println("anInt=" + anInt +
            "; aNumber value=" + aNumber.value);
    }
}
```

20.- Sea el árbol binario de búsqueda de la derecha, en que las claves son cadenas de texto y se usa el orden alfabético como criterio de ordenación. Si se inserta un nuevo dato con clave "Ruhe", este se añadirá al árbol:

- (a) Como hijo derecho del nodo "Anfreundung".
- (b) *** Como hijo izquierdo del nodo "Sicherheit".
- (c) No se puede insertar, porque el árbol de la figura no cumple las condiciones para ser un árbol binario de búsqueda.
- (d) Como hijo derecho del nodo "Traum".

