

Programación de Sistemas GITT, GIT, GISA, GISC

Leganés, 30 de mayo de 2019
Duración de la prueba: 40 min

Examen final convocatoria ordinaria (teoría)
Puntuación: 3 puntos sobre 10 del examen

Sólo una opción es correcta en cada pregunta. Cada respuesta correcta suma 0,15 puntos. Cada respuesta incorrecta resta 0,05 puntos. Las preguntas no contestadas no suman ni restan puntos.

Marca:  Anula:  No uses:   

- 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 considera no contestada.
- Rellena **tus datos personales** antes de comenzar a realizar el examen.

Nombre:

Grupo:

Firma:

NIA:

	A	B	C	D		A	B	C	D
1					11				
2					12				
3					13				
4					14				
5					15				
6					16				
7					17				
8					18				
9					19				
10					20				



- 1.- Dado el siguiente código y las variables `array = {1, 2, 3}` y `String b="hello"`, ¿qué valor tienen las variables `array` y `b` después de llamar al método `m(array, b)`?

```
public static void m(int[] array, String b){
    for(int i=0; i<array.length; i++){
        array[i] = array.length;
        b = "bye";
    }
}
```

- (a) *** `array = {3,3,3}` `b = "hello"`
- (b) `array = {1,2,3}` `b = "hello"`
- (c) `array = {3,3,3}` `b = "bye"`
- (d) `array = {1,2,3}` `b = "bye"`

- 2.- Dado el siguiente código, ¿cuál será la salida tras ejecutar el método `main`?

```
public class Test{
    private static int a = 1;
    private static int b = 2;
    public static void main(String[] args){
        int a1 = a + b++;
        int a2 = ++a + 5;
        System.out.println(a1 + " " + a2);
    }
}
```

- (a) *** 3 7
- (b) 4 7
- (c) 3 6
- (d) 4 6

- 3.- Dado el siguiente código, indique qué cobertura de métodos se alcanza en la clase `A` tras ejecutar el método `main`.

```
public class A{
    String a;
    public A(String a){setA(a);}
    public A(){this(null);}
    public setA(String a){this.a=a;}
}
public class B extends A{}
public class Test{
    public static void main(String[] args){
        B test = new B();
    }
}
```

- (a) *** 100 %

- (b) 0 %
- (c) 33 %
- (d) 67 %

4.- Dado el siguiente código, indique por qué sentencia habría que sustituir *XXXX* para modificar el valor de la variable *a* y que pase a tomar el valor 5.

```
public class A{
    private static int a = 3;
    public void setA(int a){this.a =a;}
}
public class B extends A{
    public void modifyA(){
        XXXX
    }
}
```

- (a) ******* setA(5);
- (b) super.a = 5;
- (c) this.a = 5;
- (d) a = 5;

5.- Dado el siguiente código, indique por qué modificadores habría que sustituir *xxxx*, y *yyyy* para que las declaraciones de las clases *A* y *B* fuesen correctas.

```
public interface I1{void m1();}
public interface I2{void m2();}
public xxxx class A implements I1{ }
public yyyy class B extends A implements I2{
    public void m2(){...}
}
```

- (a) ******* xxx = abstract, yyy = abstract.
- (b) xxx = abstract, yyy sin modificador
- (c) xxx sin modificador, yyy = abstract
- (d) xxx sin modificador, yyy sin modificador

6.- Dado el siguiente código indique cuál de las siguientes afirmaciones sobre los constructores de las clases *A* y *B* es correcta.

```
public class A{
    int a;
    public A(int a){this.a = a;}
}
public class B extends A{
    int b;
    public B(int a, int b){
```

```

        super(a);
        this.b = b;
    }
}

```

- (a) *** El constructor de A es invocado por el constructor de B.
- (b) El constructor de A está sobrecargado.
- (c) El constructor de A está sobreescrito.
- (d) Aunque no se indique explícitamente, A tiene un constructor por defecto que asigna al atributo a el valor 0.

7.- Dadas las interfaces I1, e I2 que hereda de la anterior, y la clase abstracta A que implementa la interfaz I2 y su hija no abstracta B, indique cuál de las siguientes sentencias es *INCORRECTA*.

- (a) *** A a = new A();
- (b) A a = new B();
- (c) I1 i1 = new B();
- (d) I2 i2 = new B();

8.- Indique cuál de las siguientes afirmaciones es aplicable a pruebas de caja blanca.

- (a) *** La cobertura de ramas es un tipo de prueba de caja blanca.
- (b) Las clases de equivalencia definen subconjuntos disjuntos de posibles datos de entrada para pruebas de caja blanca.
- (c) Las pruebas de caja blanca también se pueden llamar de entrada/salida.
- (d) Las pruebas de caja blanca no requieren conocer la estructura interna del código.

9.- Indique qué devuelve el siguiente método recursivo para a = 2.

```

public static int m(int a){
    int result = 0;
    if(a<=0){
        result = 3;
    }else{
        result = m(a-1) + m(a-2) - 2;
    }
    return result;
}

```

- (a) *** 5
- (b) 3
- (c) -2
- (d) 4

10.- Indique cuál de las siguientes afirmaciones sobre recursión es *correcta*.

- (a) *** La recursión por la cola se puede convertir en un bucle.
- (b) La recursión anidada también se puede llamar recursión en cascada.
- (c) La recursión en cascada es más eficiente que la recursión por la cola.
- (d) La recursión anidada es un caso de recursión lineal.

11.- Si tenemos un árbol binario no vacío representado como un array, sabiendo que, dado un nodo en una posición i del array, su hijo izquierdo se encuentra en la posición $2*i$ y su hijo derecho en la posición $2*i+1$, ¿en qué posición del array debe situarse el nodo raíz del árbol?

- (a) *** 1
- (b) 0
- (c) $2*i - 1$
- (d) i

12.- El siguiente árbol binario, representado mediante el array {15,13,14,9,10,12,11,1,2,3,4,5,6,7,8}, es:

- (a) *** Árbol binario y *max-heap* (montículo de clave máxima).
- (b) Árbol binario de búsqueda.
- (c) Árbol binario y *min-heap* (montículo de clave mínima).
- (d) Árbol binario sin ser montículo ni de búsqueda

13.- Dada la clase `LinkedListQueue<E>` que implementa una cola sin gestión de excepciones, ¿qué se imprime por pantalla después de ejecutar este programa?

```
public class Main{
    public static void main(String args []){
        LinkedListQueue<Integer> lq = new LinkedListQueue<Integer>();
        lq.enqueue(new Integer(10));
        lq.enqueue(new Integer(15));
        lq.enqueue(new Integer(20));
        Integer i = lq.dequeue();
        i = lq.front();
        i = lq.dequeue();
        i = lq.dequeue();
        i = lq.front();
        System.out.println(i);
    }
}
```

- (a) *** null
- (b) 20
- (c) 10
- (d) 15

14.- ¿Qué estructura de datos debería ser utilizada para implementar una cola que no es de prioridad?

- (a) *** Deque
- (b) Montículo
- (c) Árbol binario
- (d) Pila

15.- En una implementación de una cola doblemente enlazada usando la técnica *dummy nodes* (nodos centinela). Si guardamos N datos en los N nodos con información de la cola, ¿cuántos nodos tiene en total la cola (incluyendo los que almacenan información y los que no)?

- (a) *** $N+2$
- (b) N
- (c) $N+1$
- (d) $N+3$

16.- Una multinacional posee alrededor de 100.000 empleados repartidos a lo largo de todo el planeta. La multinacional dispone de un software propietario que almacena la información de cada uno de los empleados, incluyendo nombre, apellidos, fecha de nacimiento, cuenta de correo electrónico y usuario y contraseña de acceso al ordenador. Sabiendo que la operación de *login* (mediante usuario y contraseña) es la más frecuente y debe realizarse lo más rápido posible, ¿qué estructura de datos debería implementar el software propietario de la multinacional para almacenar la información de sus empleados?

- (a) *** Árbol binario de búsqueda.
- (b) Lista enlazada.
- (c) Montículo como cola de prioridad.
- (d) Pila.

17.- Dado la siguiente interfaz, ¿qué devuelve *top()* si *isEmpty()* devuelve *true*?

```
public interface Stack<E> {  
    boolean isEmpty();  
    int size();  
    E top();  
    void push(E info);  
}
```

- (a) *** *null*
- (b) *new E()*
- (c) *throw new EmptyStackException()*
- (d) *throw new StackOverflowException()*

18.- Se desea ordenar de menor a mayor un array donde $N = 1.000.000$ de elementos. El array es de enteros e inicialmente está ordenado de mayor a menor. ¿Qué algoritmo de ordenación habría que elegir buscando la máxima eficiencia?

- (a) *** HeapSort.

- (b) SelectionSort.
- (c) InsertionSort.
- (d) QuickSort.

19.- Dado el siguiente array de enteros ordenado {6,12,15,16,20,25,30,38,40,50,60,61,67,80,85}, ¿en cuántas iteraciones se localiza el número 61 usando una implementación de búsqueda binaria iterativa?

- (a) *** 2
- (b) 1
- (c) 3
- (d) 4

20.- Dado el siguiente código que imprime por pantalla el contenido de una lista enlazada. Si se llama a este método *print()* sobre una lista de 10 elementos, ¿qué número de elementos tiene la lista después de llamar a *print()*?:

```
public void print(){
    if (first != null){
        Node<E> current = first;
        while (first!=null){
            System.out.println(first.getInfo());
            first = first.getNext();
        }
    }
}
```

- (a) *** 0
- (b) 10
- (c) -1
- (d) 1