



NOMBRE:  
APELLIDOS:  
NIA:  
GRUPO:

## Convocatoria Ordinaria

### 1ª Parte: Teoría (3 puntos sobre 10)

Duración: 30 minutos  
Puntuación máxima: 3 puntos  
Fecha: 28 de mayo de 2021

#### Ejercicio-1

Si el siguiente método recibe un array con los valores 1,2,4,5,6,0,3 ¿qué devuelve este método?

```
public static int method(int[] data) {  
    int a = 0;  
    if(data.length>=1){  
        int b = data[0];  
        for (int i=1; i < data.length; i++){  
            if (b > data[i]){  
                a = i;  
                b = data[i];  
            }  
        }  
    }  
    return a;  
}
```

Seleccione una:

- ☐ a. 6
- ☐ b. 5
- ☐ c. 0
- ☐ d. Se lanza una `ArrayIndexOutOfBoundsException`

#### Ejercicio-2



Dado la siguiente sentencia para la creación de un array, ¿cuál de las siguientes frases es correcta?

```
Object[] objects = {new String("exam"), new Integer(2021), new Boolean(true)};
```

Seleccione una:

- ☐ a. Es un caso de downcasting.
- ☐ b. Los objetos del array no pueden almacenar instancias de clases distintas de Object.
- ☐ c. La creación de los objetos de las clases Integer y Boolean no es correcta.
- ☐ d. La sentencia es correcta.

### Ejercicio-3

La sobrecarga de métodos consiste en:

Seleccione una:

- ☐ a. Programar en una clase métodos con el mismo nombre, el mismo número y tipo de parámetros pero diferentes tipos de retorno.
- ☐ b. Sustituir un método heredado por otro con el mismo nombre en la subclase.
- ☐ c. Usar referencias de clase para apuntar a objetos en clases que heredan de otra clase.
- ☐ d. Programar en una clase métodos con el mismo nombre, pero que reciben un número o tipo de parámetros diferente

### Ejercicio-4



Dadas las siguientes declaraciones de clases e interfaz, ¿cuál de las siguientes asignaciones causaría un error de compilación?

```
public interface Identifiable{...}  
public class Person implements Identifiable{...}  
public class Student extends Person{...}  
public class Intern extends Student{...}
```

Seleccione una:

- ☐ a. Identifiable i = new Person();
- ☐ b. Person p = new Intern();
- ☐ c. Identifiable i = new Intern();
- ☐ d. Student s = new Identifiable();

#### Ejercicio-5

El modificador final aplicado a un método...

Seleccione una:

- ☐ a. El modificador final no puede aplicarse a métodos.
- ☐ b. indica que el valor de retorno del método es una constante.
- ☐ c. indica que el método no puede ser sobrescrito.
- ☐ d. indica que el método utiliza constantes.

#### Ejercicio-6

Un atributo estático...

Seleccione una:

- ☐ a. debe llamarse siempre con this.nombreAtributo.
- ☐ b. indica que sólo se puede acceder al atributo desde la misma clase y sus clases derivadas.
- ☐ c. existe sólo una vez para todas las instancias de la clase donde se define el atributo.
- ☐ d. indica que el atributo es una constante y su valor no puede cambiarse mientras se ejecuta el programa.

#### Ejercicio-7



Dado el siguiente método recursivo:

```
public static int m(int x, int y) {  
    if (x<=1) {  
        return y;  
    }  
    else {  
        return m(x-1, x + m(x-2,y));  
    }  
}
```

Seleccione una:

- ☐ a. Es una recursión anidada.
- ☐ b. Es una recursión lineal no por la cola.
- ☐ c. No es un método recursivo.
- ☐ d. Es una recursión mutua.

#### Ejercicio-8

Dado el siguiente método recursivo, qué frase es correcta si se llama a este método con n igual a 4 y m igual a 3.

```
public static int method(int n, int m){  
    if(n < m){  
        return 3;  
    }else{  
        return 2*method(n--, ++m);  
    }  
}
```

Seleccione una:

- ☐ a. La llamada a method(4,3) devuelve 12 como resultado.
- ☐ b. La llamada a method(4,3) devuelve 6 como resultado.
- ☐ c. La llamada a method(4,3) causa un StackOverflowError
- ☐ d. Este es un caso de recursión lineal no por la cola.



## Ejercicio-9

Nos dan un método que calcula el área de un cuadrado para un lado dado recibido como parámetro. Queremos hacer pruebas de caja negra sobre este método. Selecciona el conjunto de valores que nos permita probar todas las clases de equivalencia y valores frontera de este método.

Seleccione una:

- ☐ a. -1, 1, 2, 3
- ☐ b. 0, 0.5, 1
- ☐ c. -5, 0, 5
- ☐ d. 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

## Ejercicio-10

Dada una cola vacía implementada a través de una lista enlazada, y en la que los siguientes elementos se insertan uno a uno en el orden indicado 3,4,8,2,1,7,9, ¿qué devolvería la llamada al siguiente método considerando que head apunta al primer nodo que se extrae de la cola?

```
public void method(){  
    if(head != null){  
        Node aux = head;  
        while(aux.getNext()!=null){  
            aux = aux.getNext();  
        }  
        System.out.println(aux.getNext().getInfo());  
    }  
}
```

Seleccione una:

- ☐ a. 3
- ☐ b. Se lanzaría una NullPointerException
- ☐ c. 7
- ☐ d. 9

## Ejercicio-11



Tenemos la siguiente serie de sentencias:

```
Stack stack = new Stack();  
Queue queue = new Queue();  
stack.push("A");  
stack.push("B");  
stack.push("C");  
queue.enqueue(stack.pop());  
queue.enqueue(stack.pop());  
stack.push("D");  
queue.enqueue(stack.pop());  
queue.enqueue(stack.pop());
```

Pasando del primer elemento a desencolar al último elemento a desencolar, ¿qué contiene la cola después de ejecutar el conjunto de sentencias anteriormente indicadas?

Seleccione una:

- ☐ a. A,B,C,D
- ☐ b. La cola está vacía
- ☐ c. C,B,D,A
- ☐ d. C,B,A,D

Ejercicio-12



Tenemos una agencia de viajes (agency) situada en Londres (Reino Unido) y creamos un árbol de búsqueda binario para organizar los posibles destinos, donde el primer parámetro del método que inserta es el destino (información) y el segundo la distancia a Londres en kilómetros (utilizado como clave). ¿Cuál es el resultado de llamar a `agency.printInOrder()` después de ejecutar las siguientes sentencias?

```
agency.insert("Madrid", 1262);  
agency.insert("Rome", 1435);  
agency.insert("Paris", 243);  
agency.insert("New York", 5510);  
agency.insert("Cape Town", 9607);  
agency.insert("Amsterdam", 357);  
agency.insert("Vienna", 1235);  
agency.insert("Tehran", 4398);
```

Seleccione una:

- ☐ a. Paris, Amsterdam, Vienna, Madrid, Rome, Tehran, New York, Cape Town
- ☐ b. Madrid, Paris, Rome, Amsterdam, New York, Vienna, Tehran, Cape Town
- ☐ c. Madrid, Paris, Amsterdam, Vienna, Rome, New York, Tehran, Cape Town
- ☐ d. Vienna, Amsterdam, Paris, Tehran, Madrid, Rome, Cape Town, New York

### Ejercicio-13

¿Cuál de las siguientes afirmaciones es VERDADERA?

Seleccione una:

- ☐ a. Todos los subárboles de un montículo son árboles binarios de búsqueda.
- ☐ b. Todos los subárboles de un árbol binario de búsqueda son montículos
- ☐ c. El número de comparaciones para buscar un elemento en un árbol binario de búsqueda depende directamente de la profundidad del nodo que contiene la información deseada.
- ☐ d. Para que un montículo se considere completo debe tener todos los niveles llenos.



### Ejercicio-14

¿Cuántos intercambios necesita el algoritmo Bubble Sort para ordenar este array de String de menor a mayor (es decir, en orden alfabético)?

{"Denmark", "Bulgaria", "Croatia", "Estonia", "Austria", "France", "Germany", "Italy", "Hungary"}

Seleccione una:

- ☐ a. 8
- ☐ b. 9
- ☐ c. 10
- ☐ d. 7

### Ejercicio-15

¿Cuántas iteraciones se necesitan con la búsqueda lineal y con la búsqueda binaria para averiguar que el valor 13 no está en el siguiente array?

{-10, -4, 0, 3, 5, 8, 16}

Seleccione una:

- ☐ a. 7 con la búsqueda lineal y 3 con la búsqueda binaria
- ☐ b. No podemos utilizar la búsqueda binaria en este caso ya que el array no está ordenado
- ☐ c. 6 con la búsqueda lineal y 4 con la búsqueda binaria
- ☐ d. No podemos utilizar la búsqueda binaria en este caso porque hay números negativos

### Soluciones:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
b	d	d	d	c	c	a	a	c	b	c	a	c	d	a