MODELO Examen Programación Servicios y Procesos – Trimestre 1 – 2º DAM O/U

NOMBRE Y APELLIDOS:

Ejercicio 1 (4 puntos)

Responde a las siguientes preguntas tipo test. Dos respuestas incorrectas restan una correcta. Una respuesta incorrecta resta media correcta.

1. Un sistema informático tiene instalado un microprocesador con múltiples núcleos. Podemos decir que:
    a. Se trata de un sistema multiproceso.
    b. Se trata de un sistema monoproceso.
    c. Ninguna de las opciones es correcta.
2. La técnica denominada multiprogramación permite:
    a. Ejecutar múltiples procesos de forma intercalada.
    b. Ejecutar múltiples procesos de forma simultánea.
    c. Las otras dos opciones son correctas.
3. ¿Cuál es la función del Bloque de Control de Proceso (BCP)?
    a. Almacenar datos en disco duro.
    b. Almacenar información relativa al proceso.
    c. Bloquear los procesos con el fin de controlar su acceso.
4. ¿Qué ventajas tiene la programación concurrente?
    a. Facilita la programación de algoritmos.
    b. Se produce un mayor aprovechamiento de la CPU.
    c. Las otras respuestas son correctas.
5. Para lanzar o ejecutar un hilo llamaremos al siguiente método de la clase *Thread*:
    a. *start*.
    b. *launch*.
    c. *run*.
6. La diferencia entre implementar la interfaz *Runnable* y heredar de la clase *Thread* es que:
    a. Si implementamos la interfaz *Runnable* aún podemos heredar de otra clase.
    b. Si heredamos de la clase *Thread* no podemos implementar ninguna interfaz.
    c. No existe ninguna diferencia.
7. El hilo principal *M* ejecuta las siguientes líneas de código:

```
Thread h = new Thread(new H());
h.start();
Thread t = Thread.currentThread();
System.out.println("ID: " + t.getId());
System.out.println("Nombre: " + t.getName());
```

    ¿A qué hilo pertenece la información que se está mostrando por pantalla?
    a. Al hilo principal *M*.
    b. Al hilo *H*.
    c. A ninguno de los dos.

MODELO Examen Programación Servicios y Procesos – Trimestre 1 – 2º DAM O/U

NOMBRE Y APELLIDOS:

8. El método *interrupt*:
    a. Pertenece a la clase *Thread*.
    b. Debemos implementarlo si queremos utilizarlo para detener un hilo.
    c. Pertenece a la interfaz *Runnable*.
9. Si al llamar al método *interrupt* de un hilo éste se encuentra suspendido por estar ejecutando el método *sleep*:
    a. El método *sleep* lanzará una excepción.
    b. El hilo no se interrumpirá hasta que no finalice la llamada *sleep*.
    c. El hilo no se interrumpirá en ningún caso.
10. Si utilizamos *flags* para finalizar un hilo:
    a. No necesitamos esperar su finalización mediante el método *join* ya que finalizará de manera instantánea.
    b. La ejecución del hilo continuará hasta que se compruebe el valor del *flag*.
    c. Es un mecanismo en desuso en favor de las interrupciones.
11. ¿Qué técnica utilizada en clase permite a diferenes hilos comunicarse (intercambiar información) entre sí?
    a. Uso de memoria compartida.
    b. Comunicación interprocesos IPC.
    c. Envío de mensajes asíncronos.
12. Las secciones críticas:
    a. Son aquellas partes del código en las que accedemos y manipulamos recursos compartidos.
    b. Son mecanismos que provee *Java* para compartir recursos.
    c. Son mecanismos que provee *Java* para evitar condiciones de carrera.
13. ¿Qué podemos afirmar dado el siguente fragmento de código?

```
public static void main(String[] args) {
    Contador c;
    for(int i = 1; i <= 5; i++){
        c = new Contador();
        Thread t = new Thread(c);
        t.setName("Hilo " + i);
        t.start();
    }
}
```

    a. Todos los hilos creados comparten un mismo objeto contador.
    b. Cada hilo creado tiene su propio objeto contador.
    c. Los hilos están compartiendo memoria ya que existe un objeto compartido.
14. ¿Para qué sirve el mecanismo denominado monitor?
    a. Para evitar que varios hilos accedan a una misma sección crítica con el objetivo de que así se puedan producir condiciones de carrera.

NOMBRE Y APELLIDOS:

b. Para evitar que varios hilos accedan a una misma sección crítica con el objetivo de que no se produzcan condiciones de carrera.
c. Para que varios hilos puedan acceder a una misma sección crítica y así se puedan producir condiciones de carrera.

15. Dadas estas dos clases:

```
public class HiloIngreso implements Runnable {

   private Cuenta cuenta;

   public HiloIngreso(Cuenta cuenta) {
      this.cuenta = cuenta;
   }

   @Override
   public void run() {
      for(int i = 0; i < 1000; i++) {
         synchronized (cuenta){
            cuenta.ingresar(100);
         }
      }
   }
}

public class HiloReintegro implements Runnable {

   private Cuenta cuenta;

   public HiloReintegro(Cuenta cuenta) {
      this.cuenta = cuenta;
   }
   @Override
   public void run() {
      for(int i = 0; i < 1000; i++) {
         synchronized (cuenta){
            cuenta.reintegrar(100);
         }
      }
   }
}
```

¿Qué podemos afirmar?

a. Nunca se podrá ejecutar el método *ingresar* si se está ejecutando el método *reintegrar*.
b. Se podrá ejecutar el método *reintegrar* aunque se esté ejecutando el método *ingresar*.
c. No tenemos suficiente información para poder asegurar qué ocurrirá.

16. El hilo H_1 ha llamado al método *wait* del objeto *p*. Al hacerlo se ha suspendido su ejecución. Se reanudará cuando:

a. Cualquier otro hilo llame al método *notify* del objeto *p*.
b. El hilo H_1 llame al método *notify* del objeto *p*.
c. El objeto *p* llame al método *notify* del hilo H_1.

MODELO Examen Programación Servicios y Procesos – Trimestre 1 – 2º DAM O/U

NOMBRE Y APELLIDOS:

17. ¿Qué afirmación es cierta sobre el método *shutdown* de un *ExecutorService*?
    a. Finaliza de forma abrupta todos los hilos del *thread pool*.
    b. Es una llamada bloqueante, de forma que el hilo invocante quedará suspendido hasta que todos los hilos finalicen.
    c. Es una llamada no bloqueante que indica al *ExecutorService* que no debe aceptar más tareas.
18. Ejecutamos tres tareas *T1*, *T2* y *T3* utilizando un *thread pool*. Tras realizar múltiples ejecuciones observamos que al principio se suelen ejecutar dos de las tres tareas, pero la tercera no se ejecuta hasta que una de las dos anteriores finaliza. Podemos afirmar que:
    a. Estamos utilizando un *thread pool* de tamaño 1.
    b. Estamos utilizando un *thread pool* de tamaño 2.
    c. Estamos utilizando un *thread pool* de tipo *cached*.
19. Si utilizamos la palabra reservada *async* a la hora de definir un método…
    a. Éste método se ejecutará asíncronamente haciendo uso de un hilo.
    b. Podremos utilizar el operador *await* en el cuerpo del método.
    c. La ejecución del método pausará la ejecución del hilo principal hasta su finalización.
20. Tenemos un método cuya firma es:

```
async void LeerAchivo()
```

    Desde el método *Main* queremos llamar a *LeerArchivo* de la siguiente forma:

```
await LeerAchivo()
```

    ¿Cuál de las siguientes afirmaciones es cierta?
    a. Para poder realizar la operación *await* es necesario que el método *LeerArchivo* devuelva un objeto de tipo *Task*.
    b. Para declarar el método LeerArchivo como *async* es necesario que devuelva un objeto de tipo *Task*.
    c. El código anterior es correcto.
21. ¿Cuál es la función principal del nivel de transporte en el modelo TCP/IP?
    a. Permitir que un datagrama llegue desde un dispositivo origen a otro en cualquier parte del mundo.
    b. Entregar los datos a la aplicación correspondiente y distinguir entre diferentes aplicaciones mediante el uso de puertos.
    c. Transportar datos entre dispositivos que se encuentran dentro de una misma red.
22. ¿Qué protocolo del nivel de transporte no asegura que los paquetes lleguen de forma ordenada y sin errores?
    a. IP.
    b. UDP.
    c. TCP.

MODELO Examen Programación Servicios y Procesos – Trimestre 1 – 2º DAM O/U

NOMBRE Y APELLIDOS:

23. ¿Qué función principal realiza el protocolo DHCP?
    a. Asignar automáticamente direcciones IP en una red.
    b. Enviar correos electrónicos a través de una red.
    c. Convertir nombres de dominio en direcciones IP.
24. Para que un *socket* cliente se conecte a un *socket* servidor ¿Qué necesita conocer?
    a. La IP del servidor y el puerto en el que escucha el *socket* servidor.
    b. Únicamente la IP del servidor.
    c. El puerto del *socket* cliente.
25. ¿Cuál de las siguientes afirmaciones es cierta?
    a. Las clases *DataInputStream* y *DataOutputStream* únicamente disponen de métodos para leer y escribir *bytes*.
    b. Las clases *BufferedInputStream* y *BufferedOutputStream* disponen de métodos para leer y escribir diferentes tipos de datos tales como enteros, *doubles*, cadenas de texto, etc.
    c. Las clases *ObjectInputStream* y *ObjectOutputStream* disponen de métodos para leer y escribir instancias de objetos.

HOJA DE RESPUESTAS

1		11		21	
2		12		22	
3		13		23	
4		14		24	
5		15		25	
6		16			
7		17			
8		18			
9		19			
10		20			