

# 2ª GUIA DE POO 1a parte

(Roberto Tecla Parra)

Nombre: Santos Méndez Ulises Jesús

Fecha: 18/06/2021

Grupo:2CM11

## Excepciones

### I. ¿Qué llamadas a métodos se ponen dentro de un bloque try?

R= Llamadas a métodos que pueden lanzar excepciones.

### II. ¿El código dentro del bloque try como se escribe?

R= Como si se tratara de un mundo ideal.

### III. Si una excepción es lanzada en cualquier parte del bloque try ¿qué instrucciones se ejecutan?

R= Las instrucciones contenidas en el bloque catch.

### IV. ¿Para lanzar una excepción que operador se usa?

R= Se utiliza el operador throw.

### V. ¿Cuál es el argumento para el operador throw?

R= Un objeto Throwable.

### VI. ¿Qué instrucción aborta el método y lanza la excepción a lo largo de la pila de llamadas?

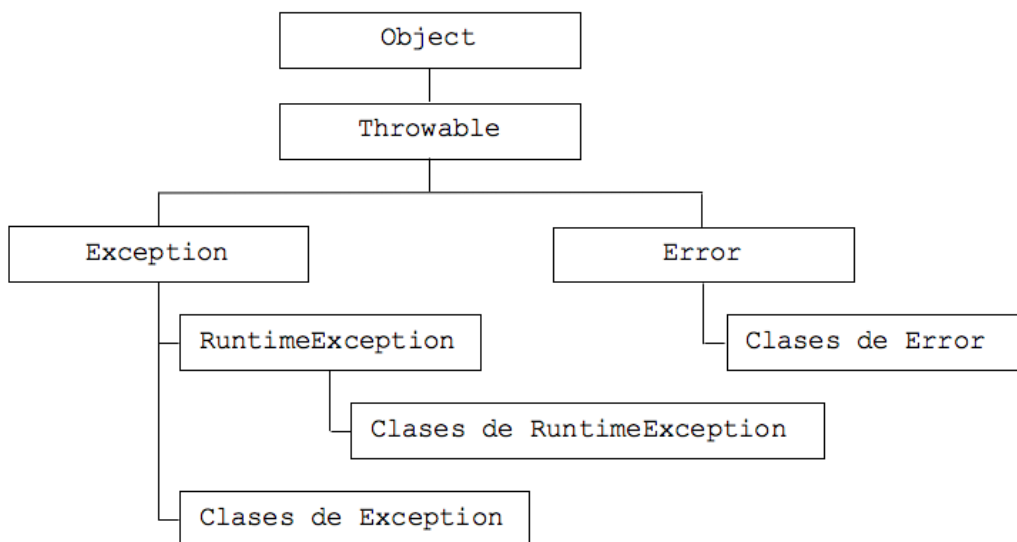
R= La instrucción throwable.

### VII.-¿Qué pasa sin ningún método en la pila de llamadas (ni siquiera main) maneja la excepción?

R= La máquina virtual de Java (JVM) finaliza la ejecución del programa con un mensaje de error.

### a) ¿Cuál es el esquema que usa Java para el manejo de errores?

```
R= try{  
    //código que podría lanzar una excepción  
}  
catch (tipoExcepcion e){  
    //procesar el tipo de excepción  
}
```



### b) ¿Cuál es el esquema que usa Java para sincronizar el acceso a datos compartidos?

```
R= Synchronized (objeto){  
    // En este método solo puede acceder un hilo a la vez  
}
```

**b1) ¿Cuál es el modelo Cliente-Servidor?**

R= El cliente envía una petición, el servidor procesa la petición y es opcional enviar una respuesta, esto depende del protocolo que se tenga, usualmente se debe de conocer la IP del servidor de esta manera se establecerá la conexión.

**c) ¿Cuál es el modelo de componentes en Java?**

R= Los JavaBeans

**d) ¿Cuál es el modelo de eventos en Java?**

R= Awt

**e) ¿En Java cual es el equivalente a RPC (Remote procedure Call)**

R= El equivalente es RMI (Remote Method Invocation)

**Hilos**

**f) ¿Qué es un quantum?**

R= Un Quantum es el tiempo máximo que un proceso (programa) puede hacer uso del procesador (CPU).

**g) ¿Qué es un cambio de contexto?**

R= Cuando finaliza el tiempo de ejecución de un proceso.

**h) ¿Qué es una condición de competencia?**

R= Es la condición que permite una relativa prioridad de un hilo sobre otro.

**i) ¿Qué es una región crítica?**

R= La porción de código de un programa de ordenador en la que se accede a un recurso compartido (estructura de datos o dispositivo) que no debe ser accedido por más de un proceso o hilo en ejecución.

**j) ¿Qué es exclusión mutua?**

R= Un proceso puede un recurso que ya está asignado a otro proceso y la condición de no aparición se debe de cumplir. Consiste en que un solo proceso excluye temporalmente a todos los demás para usar un recurso compartido de forma que garantice la integridad del sistema.

**k) ¿Qué es espera ocupada?**

R= Estado donde un proceso repetidamente verifica la condición, como esperar la entrada por teclado o que el ingreso de una sección crítica este habilitado.

**l) ¿Qué es un multitarea?**

R= Permitir que varios procesos o aplicaciones se ejecuten aparentemente al mismo tiempo.

**m) ¿Qué es un planificador?**

R= Se define como la propiedad de que varios procesos compartan recursos comunes.

**n) ¿En Thread hilo=new Thread(this); this es un objeto de una clase que implementa la interfaz Runnable?**

**o) ¿En (new Button()).addActionListener(this); this es un objeto de una clase que llama al método que se encuentra en this?**

**¿Qué es un método ...**

**p) de clase?**

R= Los métodos de clase pueden operar con las variables de clase, pero no pueden acceder a las variables de instancia declaradas dentro de la clase. Un método de clase es aquel que puede ser invocado sin existir una instancia. (Static).

**q) de instancia?**

R= Los métodos de instancia operan sobre las variables de instancia de los objetos, pero también tienen acceso a las variables de clase. (Void).

**r) abstracto?**

R= Es un método declarado, pero no implementado, es decir, es un método del que sólo se escribe su nombre, parámetros y tipo devuelto, pero no su código.

**s) sincronizado?**

R= Los métodos sincronizados garantizan el acceso en forma de exclusión mutua a los datos encapsulados, siempre que estos datos se accedan sólo a través de estos métodos.

**t) remoto?**

R= Declarado en una interfaz remota y lanza excepciones remotas.

**1.- ¿Dentro de que método está el cuerpo de un hilo?**

R= Dentro del método run().

**2.- ¿Cuáles son los estados por los que puede pasar un hilo durante su ciclo de vida?**

R= Nacido, listo, en ejecución, en espera, dormido, suspendido, bloqueado, muerto.

**3.- ¿Como pasa un hilo del recién nacido estado al estado listo?**

R= Llamando el método start().

**4.- ¿Cómo pasa un hilo del estado listo al estado ejecutable?**

R= Al otorgarle el recurso del procesador para que pueda ejecutarse.

**5.- ¿Cómo pasa un hilo del estado ejecutable al estado en espera?**

R= Llamando al método wait()

**6.- ¿Cómo pasa un hilo del estado ejecutable al estado dormido?**

R= Llamando al método sleep()

**7.- ¿Cómo pasa un hilo del estado ejecutable al estado suspendido?**

R= Con el método suspend().

**8.- ¿Cómo pasa un hilo del estado ejecutable al estado dormido?**

R= Llamando al método sleep()

**9.- ¿Cómo pasa un hilo del estado ejecutable al estado muerto?**

R= Con el método stop()

**10.- ¿Cómo pasa un hilo del estado en espera al estado listo?**

R= Con el método notify() o notifyAll().

**11.- ¿Cómo pasa un hilo del estado dormido al estado listo?**

R= Expira el intervalo de sueño.

**12.- ¿Cómo pasa un hilo del estado suspendido al estado listo?**

R= Con el método resume().

**13.- ¿Cómo pasa un hilo del estado bloqueado al estado listo?**

R= Al completar la solicitud de E/S

**14.- Para qué sirve el método start()?**

R= Pone un hilo en estado listo.

**15.- Para qué sirve el método stop()?**

R= Detiene un hilo poniéndolo en estado “muerto”.

**16.- ¿Para qué sirve el método suspend()?**

R= Pasa un hilo de ejecutable a suspendido. Un hilo no consume recursos en este estado.

**17.- ¿Para qué sirve el método resume()?**

R= Permite cambiar el estado de un hilo de suspendido a ejecutable.

**18.- ¿Para qué sirve el método wait()?**

R= El hilo que invoca se bloquea hasta que ocurre un timeout y otro hilo llame al método notify() ó notifyall() sobre el mismo objetos.

**19.- ¿Para qué sirve el método notify()?**

R= Sirve para despertar o desbloquear un hilo, si lo hay esperando

**20.- ¿Para qué sirve el método notifyall()?**

R= Despierta a todos los hilos que están esperando.

**21.- Si un hilo se lleva del estado ejecutable al estado bloqueado llamando el método suspend. ¿Cómo puede llevarse de nuevo al estado ejecutable?**

R= Invocando el método resume()

**22.- ¿Qué método debe (¿llamar un hilo para pasar al estado bloqueado cierto número de milisegundos y transcurridos estos volver al estado ejecutable?**

R= El método sleep()

**23.- ¿Cuáles son las dos causas por las que un hilo llega al estado muerto?**

R= Porque finalizo el método run() o porque se invocó al método stop()

**24.- ¿Qué método se use para finalizar la vida de un hilo?**

R= El método stop()

**25.- ¿Qué método coloca el primer hilo de la cola de espera de un objeto en el estado ejecutable?**

R= El método notify().

**27.- ¿Bajo qué circunstancias implementa una clase la interfaz Runnable?**

R= Cuando la clase ya ha heredado de otra atributos

**28.- ¿Qué método(s) tiene que implementar una clase que implements la interfaz Runnable?**

R= El método run().

**29.- ¿En Java que es un monitor?**

R= Son un esquema de sincronización que usa java para sincronizar el acceso a datos compartidos.

**30.- ¿Por qué es necesario sincronizar el acceso a datos compartidos?**

R= Para garantizar la integridad de los datos.

**31.- ¿Qué pasa cuando un método declarado como synchronized se está ejecutando en un objeto?**

R= Quiere decir que dos o más hilos necesitan acceso a un recurso compartido que solo puede ser utilizado por un hilo a la vez.

**32.- ¿Por qué se puede presentar un deadlock cuando se usan métodos sincronizados?**

R= Porque pueden cumplirse alguna de las 4 condiciones para que se dé un "deadlock" :

- Los procesos deben reclamar un acceso exclusivo a los recursos.
- Los procesos deben retener los recursos mientras esperan otros.
- Los recursos pueden no ser removidos de los procesos que esperan.
- Existe una cadena circular de procesos donde cada proceso retiene uno o más recursos que el siguiente proceso de la cadena necesita.

## **Red**

**¿Qué es un protocolo de comunicaciones?**

R= Es un sistema de reglas que permiten que dos o más entidades de un sistema de comunicación se comuniquen entre ellas para transmitir información por medio de cualquier tipo de variación de una magnitud física.

**32.1 En comunicaciones ¿qué es un protocolo?**

R= Es un sistema de reglas que permiten que dos o más entidades de un sistema de comunicación se comuniquen entre ellas para transmitir información por medio de cualquier tipo de variación de una magnitud física.

**32.2 ¿Por qué se usan pilas de protocolos?**

R= Porque es una forma de mantener ordenada la colección de protocolos entrantes para poder seguir la secuencia de estos.

**32.3 - ¿Qué es TCP/IP?**

R= Son las siglas de Protocolo de Control de Transmisión/Protocolo de Internet (en inglés Transmission Control Protocol/Internet Protocol), un sistema de protocolos que hacen posibles servicios Telnet, FTP, Email, y otros entre ordenadores que no pertenecen a la misma red.

### **32.4 - ¿Qué es una dirección IP?**

R= Es una etiqueta numérica que identifica, de manera lógica y jerárquica, a una interfaz (elemento de comunicación/conexión) de un dispositivo (habitualmente una computadora) dentro de una red que utilice el protocolo IP.

### **32.5 - ¿Qué es un puerto?**

R= Un puerto es una interfaz a través de la cual se pueden enviar y recibir los diferentes tipos de datos

### **32.6 - ¿Qué es números de puertos están reservados?**

R= Los puertos inferiores al 1024 son puertos reservados por los sistemas operativos y usados por "protocolos bien conocidos" como por ejemplo HTTP. Los comprendidos entre 1024 (0400 en hexadecimal) y 49151 (BFFF en hexadecimal) son denominados "registrados" y pueden ser usados por cualquier aplicación.

## **Sockets**

### **33.-Que es un socket?**

R= Un socket es un punto de comunicación por el cual el proceso puede emitir o recibir información.

### **35.-Cuál es el nombre de la clase de un socket cliente?**

R= Socket

### **36.-Cuáles son los 2 argumentos más usuales del constructor de un socket cliente?**

R= El nombre de la máquina en donde estamos intentando abrir la conexión(host) y el puerto (número) del servidor que está corriendo sobre el cual nos queremos conectar

### **37.-Cuál es el Ciclo de Vida de un socket cliente?**

R=

- a. Un nuevo socket es creado usando un constructor socket().
- b. El socket intenta conectarse a un host remoto.
- c. Una vez que la conexión es establecida los host local y remoto obtienen flujos de E/S para enviarse datos uno a otro. Lo que significa que los datos dependen del protocolo.
- d. Cuando la transmisión de datos esta completa, uno o ambos cierran la conexión.

### **38.-Cuál es el nombre de la clase de un socket servidor?**

R= ServerSocket

### **39.-Cuáles es el argumento más usual del constructor de un socket servidor?**

R= El puerto a que el cliente establecerá la conexión. Cuando el cliente solicite una conexión, el servidor abrirá la conexión socket con el método accept().

### **40.-Cuál es el ciclo de vida de un socket servidor?**

R=

- 1) Un ServerSocket nuevo es creado en un puerto particular, usando el constructor ServerSocket().
- 2) El ServerSocket(), escucha los intentos de conexión entrantes en ese puerto usando su método accept().
- 3) Dependiendo del tipo de servicio el método getInputStream() o el método getOutputStream() del socket será llamado para obtener flujos de E/S que comunican con el cliente.
- 4) El servidor y el cliente interactúan de acuerdo con un protocolo establecido hasta que es tiempo de cerrar la conexión.
- 5) El servidor, el cliente o ambos cierran la conexión.
- 6) El servidor regresa al paso 2 y espera la siguiente conexión.

### **41.-Para qué sirve el método accept?**

R= Accept se bloquea hasta que un cliente

### **42.-Qué tipo de valor regresa el método accept y para qué sirve dicho valor?**

R= Accept() regresa un socket que conecta al cliente con el servidor.

### **43.-Qué métodos se usan para obtener flujos de E/S de un socket?**

R= Los métodos getInputStream() o getOutputStream().

## //NOTA: TEMAS YA NO VISTOS EN CLASE

### **RMI**

- 44.-Para qué sirve RMI?
- 45.-Con que se programa a más alto nivel con sockets o con RMI
- 46.- ¿Con RMI la programación está centrada en la invocación de métodos remotos, en el caso de los sockets en que se centra la programación?
- 47.- Escriba los nombres de las capas del protocolo RMI del lado del cliente.
- 48.-Escriba los nombres de las capas del protocolo RMI del lado del servidor.
- 49.- ¿Para qué sirve la capa de referencia remota?
- 50.- ¿Para qué sirve la capa Stub?
- 51.- ¿Para qué sirve la capa del esqueleto?
- 56.- ¿Para qué sirven los métodos bind y rebind de la clase Naming?
- 57.- ¿Para qué sirve el método lookup de la clase Naming?
- 58.- ¿Cuál es el nombre del programa que se usa para generar los stubs y los esqueletos para los objetos remotos?
- 59.- ¿Cuál es el nombre del programa que se usa para registrar un objeto remoto?

### **HTTP**

- 60.- ¿Qué es HTTP?
- 61.-Para que sirven las formas de html?
- 62.-Que es CGI?
- 63.- Cuando se usa el método POST en una petición http?
- 64.- Cuando se usa el método GET en una petición http?
- 64.1-En CGI que es una cadena de consulta?
- 65.- Que es un Servlet?

### **JavaBeans**

- 71. ¿Cuáles son las tres partes de un JavaBean?
- 72. ¿Qué es una propiedad de un JavaBean?
- 73 ¿Qué es una propiedad indexada?
- 74 ¿Qué es una propiedad encadenada?
- 75 ¿Qué es una propiedad restringida?
- 76 ¿Qué es un un método getter?
- 77 ¿Qué es un un método setter?
- 78.-Por medio de que objetos se comunican los JavaBeans?
- 79.-Que es el mecanismo de reflexión
- 80-Que es el mecanismo de introspección

### **JDBC**

- Que es un SGBD?
- Que es una tabla?
- Que es un registro?
- Que es un campo?
- Que es un metadato?
- Que es un diccionario de datos?
- Que significan las siglas CRUD
- Que significan las siglas SQL
- Para que sirve SQL
- En SQL que es resultado de una consulta

- 81.-¿Que es ODBC?
- 82.-¿Qué es JDBC?
- 83.-¿Cuáles son las partes de URL en JDBC?

- 84.-¿Cómo se realiza la conexión con JDBC a una Base de Datos?
- 85.-¿Cómo se envía una instrucción de SQL a la Base de Datos?
- 86.-¿Cómo se procesan los resultados de una instrucción de SQL?
- 87.-¿Para que sirve la clase DriverManager?
- 88.-¿Para que sirve la clase Statement?
- 89.-¿Para que sirve la clase ResultSet?