**Preguntas APX y java examen 02-Septiembre-2024**

**APX**

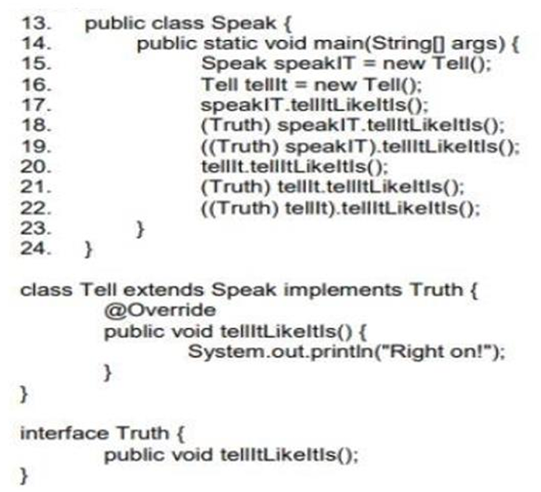
1. Las dependencias circulares son permitidas…\* Respuesta correcta **Están absolutamente prohibidas**\*
2. Son dos de los protocolos físicos que APX admite en una solicitud. **JMS y REST**
3. En caso de que una transacción deba invocar a otra, solo se puede realizar de forma Asincrónica
4. ¿Cuántas veces es posible invocar Mainframe? **2**
5. Una transacción o librería, dentro de su lógica, debe evitar invocar más de \_\_\_ librerías 9
6. No se debe manejar el objeto Datasource directamente
7. En un Batch, ¿Cuál es el step donde el desarrollador introducirá la lógica empresarial necesaria? ¿¿TASK
8. Implementación de PowerCurve en APX? Verdadero falso?
9. Los componentes de seguridad lógica que dan servicio a la arquitectura no están acoplados a la Arquitectura APX. Verdadero
10. DEBUG, sirve para información de muy bajo nivel solo útil para el debug de la aplicación.
11. No es una característica de the **blob**
12. En el caso de Oracle el acceso a los datos es a través de:
    1. JDBC
    2. ORM
    3. JPA
    4. SQL
13. Componente que se encarga de encapsular toda la lógica empresarial y el acceso a datos. Respuesta: Librería.
14. La creación de hilos o su gestión por las aplicaciones está permitida.
    1. Verdadero
    2. Falso
15. El consumo de transacción asincronica pierde el control de la ejecución y solo se puede utilizar cuando la operación realizada de forma asíncrona \_\_\_\_ para el proceso de negocio, ya que el consumidor de esa transacción no sabrá si ha concluido con un commt OK o con rollback. Respuesta: No es crucial
16. Son dos de las características ofrece la Arquitectura Batch para ampliar la funcionalidad que ofrece el framework. Respuesta: Soporte Multi-BBDD, Reading/Writing de multiples formatos de archivo
17. ¿Quié- n es el responsable de iniciar un Job?
    1. JobExecution
    2. JobLauncher
    3. JobInstance
    4. JobBegginer
18. Los archivos objeto del procesamiento Batch de APX son de naturaleza estrictamente \_\_\_\_, y este archivo no es, en ningún caso, un mecanismo de persistencia de información a largo plazo. Respuesta: Temporal.
19. ¿Cuál es la ubicación de archivos temporales utilizados en Batch?
    1. /fichtemporal/datent
    2. /fichtemcomp/datent
    3. /temporalFiles/datent
    4. /datentJobs/datent
20. ¿Cómo solucionamos un Blob?
    1. Mover la funcionalidad a la transacción reduciendo la carga en la librería.
    2. Mover el comportamiento a sus librerías relacionadas, reduciendo el acoplamiento entre librerías y simplificando los mantenimientos.
    3. Moviendo la lógica a diferentes métodos y clases dentro de la misma librería
    4. Mover el comportamiento a DTOs relacionados, reduciendo el acoplamiento entre los componentes y simplificando los mantenimientos
21. Es una contradicción al usar el patrón CRUD
    1. Mantener todas las operaciones sobre la entidad encapsuladas en la librería
    2. Tener varias librerías parciales para una única entidad
    3. Centralizar en una librería de todas las operaciones básicas sobre una entidad
    4. Tener una implementación ligera, evitando extender la lógica más allá del objetivo de la operación
22. Nivel de log que muestra información superior que permite monitorear la ejecución normal. Respuesta: Info
23. En una librería, ¿Cuál es el archivo en donde se define la lógica del negocio?
    1. UUAAR000Impl
    2. UUAAR001-app-osgi.xml y UUAAR001-app.xml
    3. UUAAR001-arq-osgi.xml y UUAAR001-arc.xml
    4. UUAAR000Abstract
24. En primer lugar, APX recibe la solicitud en uno de los protocolos físicos que admite (HTTP, JMS, REST,). Luego, crea un identificador único para la \_\_\_\_\_\_\_\_ y valide si los encabezados de solicitud están bien formateados de acuerdo con el protocolo lógico definido por Arquitectura
    1. Pila de ejecución
    2. Solicitud
    3. Transacción
    4. Ejecución

**Java**

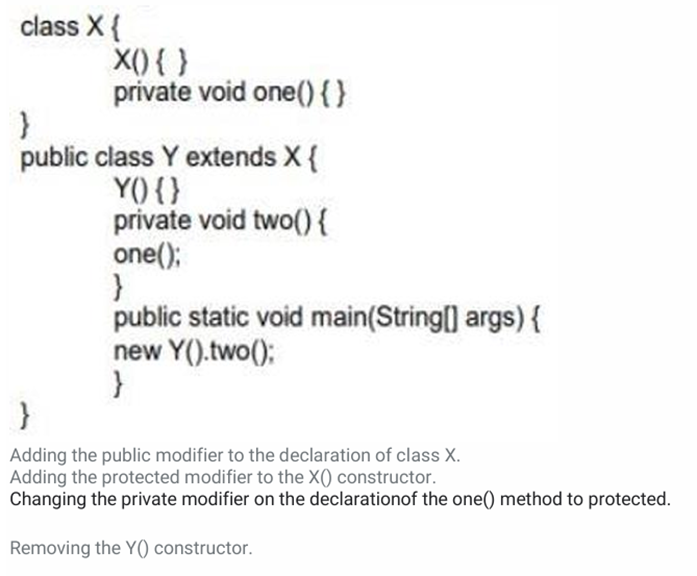
1. The BUILDER pattern is used to: Simplify the creation of complex objects.
2. In Java the difference between throws and throw Is:  
   throws indicates the type of exception that the method does not handle and throw an exception.
3. A method is declared to take three arguments. A program calls this method andpasses only two arguments.What is the result? Compilation fails
4. Which three implementations are valid?



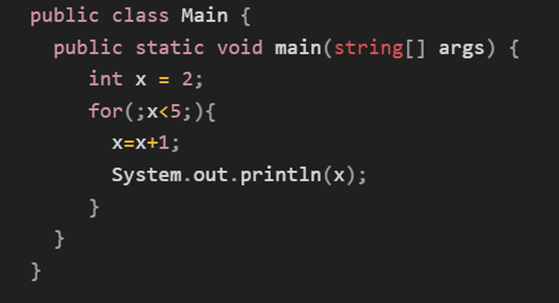
1. Which three lines will compile and output "Right on!"?



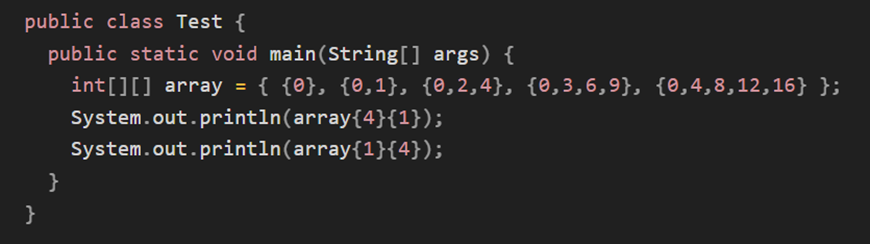
1. What changes will make this code compile? SELECCIONA 2



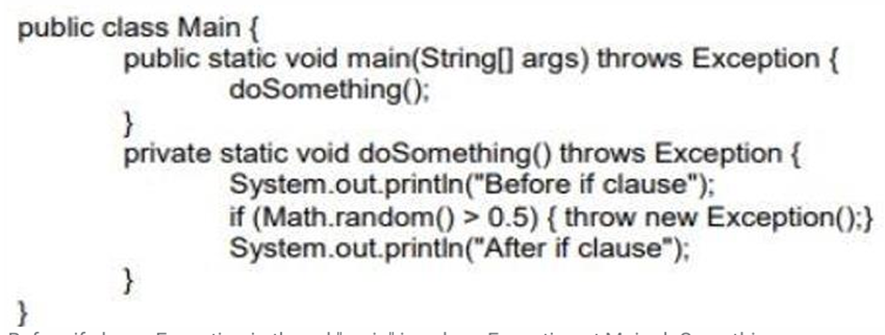
1. ¿Qué imprime?



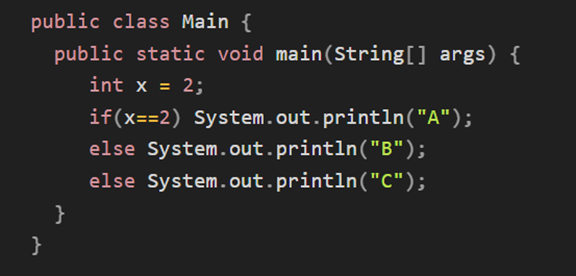
1. ¿Cuál es el resultado?



Which two possible outputs?



1. ¿Cuál es la salida?



1. How many times is 2 printed?

class Menu {

public static void main(String[] args) {

String[] breakfast = {"beans", "egg", "ham", "juice"};

for (String rs : breakfast) {

int dish = 1;

while (dish < breakfast.length) {

System.out.println(rs + "," + dish);

++dish;

}

}

}

}

Resultado: beans,2, beans,3, egg,2, egg,3, ham,2, ham,3, juice,2, juice,3

***4 veces se imprime 2***

1. What is the result?

class Person {

String name = "No name";

public Person (String nm) {name=nm}

}

class Employee extends Person {

String empID = "0000";

public Employee(String id) { empID " //18

}

}

public class EmployeeTest {

public static void main(String[] args) {

Employee e = new Employee("4321");

System.out.printiln(e.empID);

}

}

A. 4321.

B. 0000.

C. An exception is thrown at runtime.

D. Compilation fails because of an error in line 18.

1. What is the result? (No me apareció esta pregunta tal cual, pero era muy parecida)

class Atom {

Atom() {System.out.print("atom ");}

}

class Rock extends Atom {

Rock(String type) {System.out.print(type);}

}

public class Mountain extends Rock {

Mountain(){

super("granite ");

new Rock("granite ");

}

public static void main(String[] a) {new Mountain();}

} //atom granite atom granite

A. Compilation fails.

B. Atom granite.

C. Granite granite.

D. Atom granite granite.

E. An exception is thrown at runtime.

F. Atom granite atom granite.

1. What is the result?

import java.text.\*;

public class Align {

public static void main(String[] args) throws ParseException {

String[] sa = {"111.234", "222.5678"};

NumberFormat nf = NumberFormat.getInstance();

nf.setMaximumFractionDigits(3); //NO HACE NADA

for (String s : sa) {

System.out.println(nf.parse(s));

}

}

}

A. 111.234 222.567

B. 111.234 222.568

C. 111.234 222.5678

D. An exception is thrown at runtime

1. What is the result?

interface Rideable {

String getTicket(); //No recuerdo muy bien el nombre del método, pero era diferente

}

public class Camel implements Rideable {

int weight = 2;

String getGait() {

return mph + ", lope";

}

void go(int speed) {

++speed;

weight++;

int walkrate = speed \* weight;

System.out.print(walkrate + getGait());

}

public static void main(String[] args) {

new Camel().go(8);

}

}

1. Tema: Lambdas con predicate
2. Which two actions, used independently, will permit this class to compile?

import java.io.IOException;

public class Y {

public static void main(String[] args) {

try {

doSomething();

} catch (RuntimeException e) {

System.out.println(e);

}

}

static void doSomething() {

if (Math.random() > 0.5) {

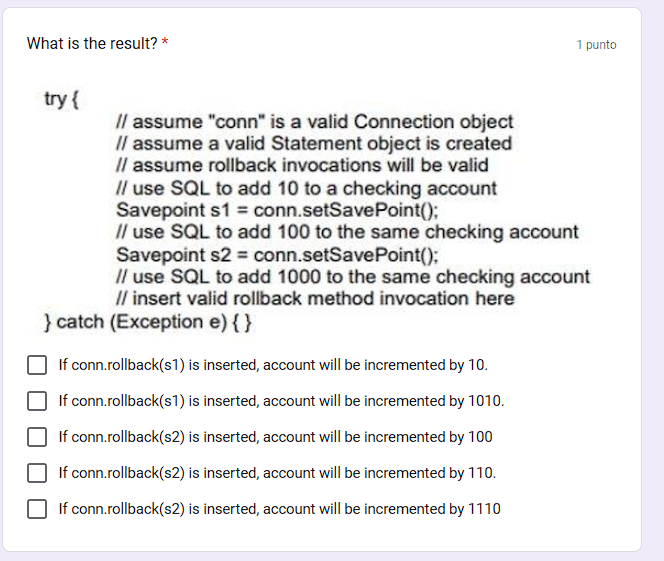
throw new IOException();

}

throw new RuntimeException();

}

}

1. Adding throws IOException to the main() method signature and to the doSomething() method.
2. Adding throws IOException to the main() method ... IOException.
3. Adding throws IOException to the doSomething() method signature.
4. Adding throws IOException to the main() method signature.
5. Adding throws IOException to the doSomething() method signature and changing the catch.
6. ¿Cuál es el resultado?
7. Tema: API java.time - Uso de LocalDateTime y Period para manejar fechas y periodos.
8. ¿Cuáles de las siguientes opciones son válidas?

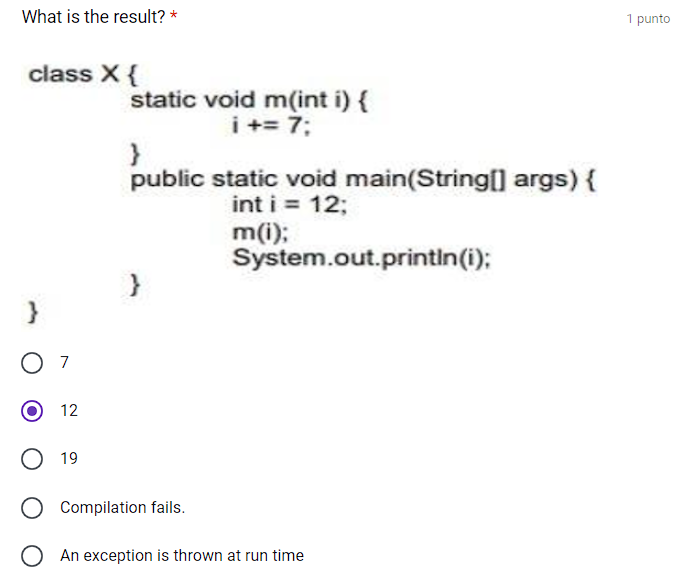
A. El constructor predeterminado proporcionado por el compilador puede ser llamado utilizando this().

B. Un constructor puede ser invocado desde un método de instancia.

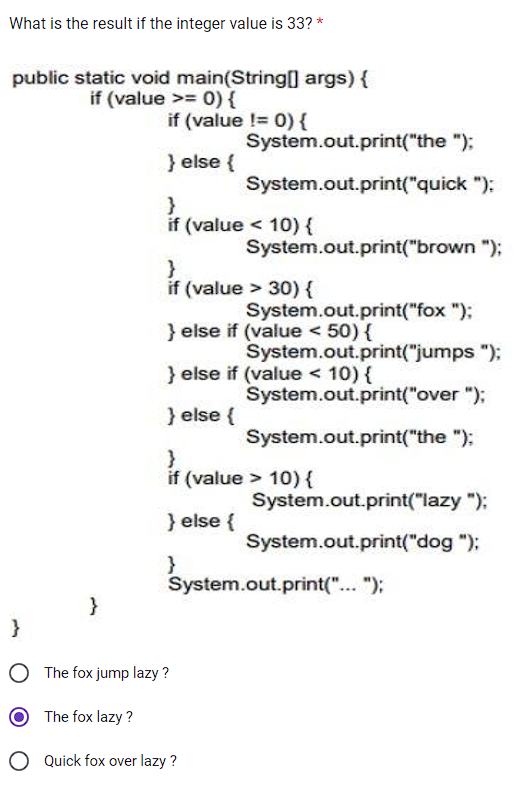
C. Una variable de instancia puede ser accedida dentro de un método de clase (static).

D. Un constructor puede ser llamado dentro de otro constructor utilizando la palabra clave this().

1. Cual es el resultado



1. 33?



What is the result if you try to compile Truthy.java and then run it with assertions enabled?

public class Truthy{

public static void main(String[] args){

int x = 7;

assert (x == 6) ? "x == 6" : "x != 6";

}

}

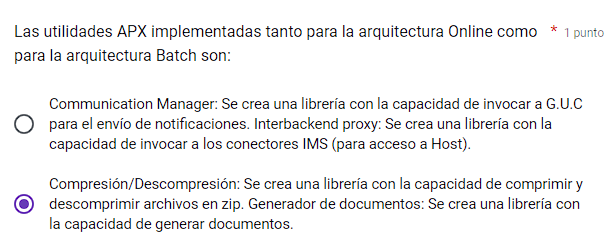
• Truthy.java compiles and the output is x != 6

• Truthy.java compiles and an AssertionError isthrown with x != 6 as additionaloutput.

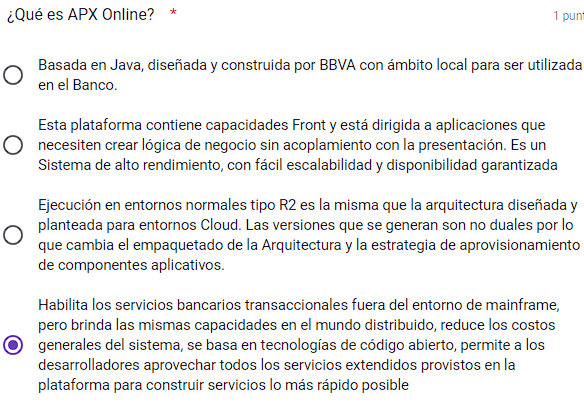
• Truthy.java does NOT compile. ←

• Truthy.java compiles and an AssertionError isthrown with no additional output

1. **APX**

****

1. **APX**

****

1. **APX**

**Definición del singleton**

**Diferencias entre clases abstractas e interfaces**

**Multihilos**