

Diagnóstico APX

Puntos totales 47/81

Online

Se ha registrado el correo del encuestado (**tonalidavid.salcedo.contractor@bbva.com**) al enviar este formulario.

Tipos de Pasos en un proceso Batch ¿Cuáles son?*

0/1

Write

No conozco la respuesta

Chunk

Process

Task

Read

Otro:

Respuesta correcta

Chunk

Task

¿Qué es patrón de diseño?*

1/1

No conozco la respuesta

Patrón de identificación único de procesos

Patrón de identificación de procesos

Identificador de diseños

Proporciona una solución a un problema de diseño

Proporciona información de la ejecución

Batch Execution Banking basada en Spring Batch, ¿ qué Invocación son restringidas ?*

0/1

Utilidades inherentes a la Arquitectura APX.

Servicios web de invocación, servicios de descanso.

No conozco la respuesta

Acceso a recursos externos a través de HTTP.

GUC

Utilice acciones previas y posteriores.

Compresión / Descompresión.

Otro:

Respuesta correcta

Servicios web de invocación, servicios de descanso.

Acceso a recursos externos a través de HTTP.

Utilice acciones previas y posteriores.

Utilidades inherentes a la Arquitectura APX.

Compresión / Descompresión.

GUC

¿Qué es el `LOGGER.warn` ?*

1/1

Imprime información útil que le permite ver el progreso de la aplicación.

Imprime advertencias o fallas no críticas de la aplicación, que no impiden que la aplicación continúe su flujo

Le permite imprimir datos informativos con detalles finos que son útiles para depurar la aplicación.

Imprime información con el máximo nivel de detalle.

Imprime mensajes de error críticos que impiden que la aplicación funcione correctamente.

Estos son los registros habilitados en Producción

No conozco la respuesta

Ejemplo de contenedor mágico*

1/1

Contiene diferentes variables

No conozco la respuesta

Un campo de entrada no debe usarse para que la transacción haga cosas funcionalmente diferentes

Todas

Contiene varios métodos en una sola librería

Un campo de entrada se devuelve a la salida

Para que sirve IMSConnect*

1/1

Permite la posibilidad de invocar una transacción de APX directamente

Permite la posibilidad de invocar una transacción de mainframe directamente

Permite la posibilidad de invocar un batch de APX directamente

Permite la posibilidad de invocar un DTO de APX directamente

Permite la posibilidad de invocar un servicio ASO directamente

No conozco la respuesta

¿Qué es Biblioteca APX?*

1/1

No conozco la respuesta

Encapsula la lógica empresarial y el acceso a los datos. Publica su interfaz para que pueda ser utilizada por terceros.

Es un desarrollo que reemplaza a Mainframe

Representación de una Entidad Comercial en forma de Bean.

Es un software diseñado para crear proyectos tipo APX

Una transacción es la unidad de aplicación que se ejecutará en APX Online. Define la Unidad

Transaccional a nivel de lógica de aplicación.

NO se deben utilizar sentencias que incluyan atributos o acciones específicas de un administrador de base de datos en particular, como lo es*

1/1

SELECT

FROM

OR

No conozco la respuesta

ROWNUM

Niveles más comunes de LOG en APX*

1/1

DEBUG, INFO

DEBUG, INFO, WARNING

TRACE, DEBUG, INFO, WARNING y ERROR
No conozco la respuesta
DEBUG, INFO, WARNING y ERROR

¿Qué es APX CLI ?*

1/1

Una transacción es la unidad de aplicación que se ejecutará en APX Online. Define la Unidad Transaccional a nivel de lógica de aplicación.
Encapsula la lógica empresarial y el acceso a los datos. Publica su interfaz para que pueda ser utilizada por terceros.
Es un software diseñado para crear proyectos tipo APX, así como administrar dependencias con otras bibliotecas APX y administrar utilidades

Representación de una Entidad Comercial en forma de Bean.
Es un desarrollo que reemplaza a Mainframe
No conozco la respuesta

¿Qué es el LOGGER.info ?*

1/1

Imprime información útil que le permite ver el progreso de la aplicación.

No conozco la respuesta
Imprime mensajes de error críticos que impiden que la aplicación funcione correctamente. Estos son los registros habilitados en Producción
Imprime advertencias o fallas no críticas de la aplicación, que no impiden que la aplicación continúe su flujo
Le permite imprimir datos informativos con detalles finos que son útiles para depurar la aplicación.
Imprime información con el máximo nivel de detalle.

Restricciones para imprimir loggers en APX*

1/1

Tampoco está permitido escribir por sysout

No se permite la generación de archivos propios para la escritura de logs.

Solo en España hay restricciones
No hay restricciones
Le permite imprimir datos informativos

No conozco la respuesta

¿Qué es Transacción APX?*

1/1

Es un desarrollo que reemplaza a Mainframe

Representación de una Entidad Comercial en forma de Bean.

Una transacción es la unidad de aplicación que se ejecutará en APX Online. Define la Unidad Transaccional a nivel de lógica de aplicación.

No conozco la respuesta

Encapsula la lógica empresarial y el acceso a los datos. Publica su interfaz para que pueda ser utilizada por terceros.

Es un software diseñado para crear proyectos tipo APX

Declarar una dependencia de compilación como opcional (<optional> true </optional>), ¿es correcto declararla?*

0/1

Dependiendo el desarrollo

Tal vez

Si

No conozco la respuesta

No

Otro:

Respuesta correcta

No

Las dependencias circulares son permitidas...*

0/1

Son permitidas

Algunas son permitidas

Depende de la necesidad del proyecto

Están absolutamente prohibidas

No conozco la respuesta

Respuesta correcta

Están absolutamente prohibidas

Diferencias entre Las invocaciones de APX Batch y invocaciones de APX Online*

0/1

No conozco la respuesta

Horarios de ejecución

Permite la posibilidad de invocar un batch de APX directamente

Permitirá ejecutar transacciones de cics solo por protocolo ps10.

Los encabezados del protocolo lógico a enviar

Ninguno

Respuesta correcta

Los encabezados del protocolo lógico a enviar

¿Por qué se deben utilizar variables BIND?*

1/1

Para enviar muchos datos al mismo tiempo

Para mejorar las sentencias

Para no afectar el rendimiento de la base de datos.

Para una mejor vista

No conozco la respuesta

El multiproceso no debe usarse en APX Batch ¿Por qué?*

1/1

Por la incompatibilidad en la JVM

Fuente de consumo de recursos descontrolado y bloqueos de otros trabajos que se ejecutan al mismo tiempo.

Es una etiqueta no permitida

No conozco la respuesta

No es parte de Spring

Otro:

Excepciones de Spring permitida*

0/1

org.springframework.web.client.RestClientException

java.lang.NumberFormatException

com.bbva.apx.exception.db.NoResultException

com.bbva.elara.utility.interbackend.cics.exceptions.BusinessException
No conozco la respuesta

Respuesta correcta

org.springframework.web.client.RestClientException

APX, ¿protocolos físicos que admite?*

1/1

HTTP, JMS, REST

No conozco la respuesta

REST

JMS

SOAP

HTTP

Batch , se pueden implementar las siguientes interfaces ¿Cuáles?*

1/1

ItemReader, ItemProcessor, ItemWriter

No conozco la respuesta

ItemProcessor

ItemWriter

ItemReader

Otro:

La creación de instancias, abrir o cerrar explícitamente conexiones a la base de datos,
¿de quien depende estas tareas?*

0/1

ASO

Bibliotecas

No conozco la respuesta

Arquitectura

Java

Respuesta correcta

Arquitectura

Agrega un campo de entrada obligatorio. Agregue un campo de salida. Modifique el tipo de un campo de entrada o salida existente. Son características de un cambio ...*

1/1

Retrocompatibles

Versionado

No conozco la respuesta

No retrocompatibles

Cambio en desarrollo

Batch consta de 3 partes ¿Cuáles son?*

1/1

Salida: escribe o realiza alguna acción como resultado del procesamiento.

No conozco la respuesta

Entrada: obtenga y lea la información de entrada al proceso.

Procesamiento: se realiza algún tipo de lógica empresarial sobre estos datos de entrada.

Otro:

Declarar una dependencia de compilación como opcional (<optional> true </optional>), ¿Qué efectos conlleva?*

0/1

Dependiendo el desarrollo

Hace que busque todas las dependencias definida

Hace que los paquetes definidos en la sección import-package también se establezcan como opcionales

No busca ninguna dependencia

No conozco la respuesta

Otro:

Respuesta correcta

Hace que los paquetes definidos en la sección import-package también se establezcan como opcionales

La severidad del aviso que ésta contextual a la transacción debe realizarse desde la misma mediante el método*

1/1

addSeverity
addAdvice
setAdvice
No conozco la respuesta
setSeverity

Los hilos son permitidos...*

1/1

Si
No conozco la respuesta
No

Depende donde se usará
Depende del JVM que se ejecute

Ejemplos de patrones de diseño*

1/1

Paginación en bibliotecas.
Paginación en transacciones.
No conozco la respuesta
CRUD en bibliotecas.
Todas

DTO en componentes APX.

Unicas excepciones que pueden ser capturadas y administradas por las aplicaciones*

0/1

com.bbva.apx.exception.business.BusinessException

com.bbva.apx.exception.io.network.TimeoutException

com.bbva.apx. exception.db.DuplicateKeyException

com.bbva.apx.exception.db.IncorrectResultSizeException

com.bbva.apx.exception.db.NoResultException

com.bbva.apx.exception.db.TimeoutException

com.bbva.elara.utility.interbackend.cics.exceptions.BusinessException

No conozco la respuesta

Respuesta correcta

com.bbva.apx.exception.business.BusinessException

com.bbva.apx.exception.io.network.TimeoutException

com.bbva.apx.exception.db.DuplicateKeyException

com.bbva.apx.exception.db.IncorrectResultSizeException

com.bbva.apx.exception.db.NoResultException

com.bbva.apx.exception.db.TimeoutException

com.bbva.elara.utility.interbackend.cics.exceptions.BusinessException

El código try ... catch (Exception exc) ¿Es permitido en APX?*

1/1

Si

No conozco la respuesta

Tal vez

No

La comunicación de APX a otros servicios NO APX se realizará ÚNICAMENTE a través de...*

1/1

APIConnector

DTOs

Librerías

No conozco la respuesta

Depende de su consumo

Que es El método getRequestHeader*

1/1

Recuperará el valor de los parámetros del encabezado con el que se ejecutó la transacción

Utilidad que Accede a BBDD

Recuperará los encabezados de ASO
No conozco la respuesta
Utilidad para organizar documentos en cabezales
Utilidad para crear encabezados para la ejecución de transacciones

Las etiquetas @SuppressWarnings o etiquetas similares son permitidas...*

0/1

Depende del JVM que se ejecute
Si

Algunas
No conozco la respuesta
No

Respuesta correcta

No

Es obligatorio que los DTOs se implementen dentro de un....*

1/1

empaquetado independiente (Bundle Batch)
empaquetado independiente (Bundle TX)
empaquetado independiente (Bundle DTO)

No conozco la respuesta
empaquetado independiente (Bundle LIB)

Que es un Chunk*

1/1

Conjuntos de datos de entrada se procesan en la misma transacción

Conjunto de readers
Conjunto de procesos
No conozco la respuesta
Realiza una operación dentro del paso (por ejemplo, invocando utilidades de arquitectura)
Otro:

Propósito Bibliotecas del motor de reglas*

1/1

Es un software diseñado para crear proyectos tipo APX, así como administrar dependencias con otras bibliotecas APX y administrar utilidades

Una transacción es la unidad de aplicación que se ejecutará en APX Online. Define la Unidad Transaccional a nivel de lógica de aplicación.

Es un desarrollo que reemplaza a Mainframe

El propósito de esta biblioteca es permitir que el desarrollador invoque las reglas comerciales que necesita a través del motor de reglas Drools

Encapsula la lógica empresarial y el acceso a los datos. Publica su interfaz para que pueda ser utilizada por terceros.

No conozco la respuesta

Chunk esta formado por 3 componentes*

1/1

Step

Procesador

Lector

Task

Escritor

No conozco la respuesta

Otro:

Para mongo ¿ que utilidad debe utilizarse ?*

0/1

Datio

JDBC

APIConnector

JPA

No conozco la respuesta

Respuesta correcta

Datio

Como se Estandarización el código en APX*

0/1

Desarrollo se defina con respecto a las reglas de Sonar
Desarrollo definido en inglés, ya que los componentes se pueden reutilizar en todo el Grupo BBVA.

Desarrollo se defina con respecto a las reglas de APX

No conozco la respuesta

Desarrollo pase por Bitbucket, jenkins, sonar

Respuesta correcta

Desarrollo definido en inglés, ya que los componentes se pueden reutilizar en todo el Grupo BBVA.

En un DTO se desea intercambiar información de forma*

0/1

coherente

organizada

asociada

agrupada

No conozco la respuesta

Respuesta correcta

coherente

organizada

agrupada

Que es Conector CICS*

0/1

Permitirá ejecutar transacciones de cics solo por protocolo ps10.

Permite la posibilidad de invocar una transacción de APX directamente

No conozco la respuesta

Permite la posibilidad de invocar una transacción de mainframe directamente

Permite la posibilidad de invocar un servicio ASO directamente

Permite la posibilidad de invocar un batch de APX directamente

Respuesta correcta

Permitirá ejecutar transacciones de cics solo por protocolo ps10.

Capacidades de APX*

0/1

Implementaciones en caliente de acceso a datos
Interoperabilidad de backends
Monitoreo de procesamiento por lotes de servicios bancarios
No conozco la respuesta

Procesamiento transaccional

Integración con servicios de seguridad

Respuesta correcta

Procesamiento transaccional
Integración con servicios de seguridad
Implementaciones en caliente de acceso a datos
Interoperabilidad de backends
Monitoreo de procesamiento por lotes de servicios bancarios

¿Qué es el LOGGER.trace ?*

0/1

No conozco la respuesta
Imprime información útil que le permite ver el progreso de la aplicación.
Imprime advertencias o fallas no críticas de la aplicación, que no impiden que la aplicación continúe su flujo
Imprime información con el máximo nivel de detalle.
Le permite imprimir datos informativos con detalles finos que son útiles para depurar la aplicación.

Imprime mensajes de error críticos que impiden que la aplicación funcione correctamente.
Estos son los registros habilitados en Producción

Respuesta correcta

Imprime información con el máximo nivel de detalle.

Ejecución de procesos Monostep. ¿Se puede usar?*

0/1

No conozco la respuesta
Tal vez
Si

Depende el desarrollo
No se recomienda la ejecución de trabajos con un solo paso.

Otro:

Respuesta correcta

No se recomienda la ejecución de trabajos con un solo paso.

Existe un API de código de barras*

1/1

Tal vez

No

No conozco la respuesta

Solo en España

SI

Su uso es restringido

@Component, @Service, @Controller y @Repository, estas anotaciones ¿que causan en las clases de implementación de las bibliotecas APX ?*

1/1

Tienen el efecto de que la biblioteca no expone su servicio OSGi.

Oculta los bundles donde se usan

Mejorar la visibilidad a los métodos a usar

No conozco la respuesta

Tira la arquitectura

El acceso a los datos debe realizarse utilizando las utilidades que Arquitectura proporciona desde el asistente de biblioteca. En el caso de bases de datos relacionales, ¿Qué utilidad se debe usar?*

1/1

APIConector

Elastic

No conozco la respuesta

Datio

JDBC

Catracteristicas del Batch*

0/1

Rendimiento / procesamiento de alta velocidad.

trazabilidad de la información y ejecución.

Sin interacción (directa) del usuario.

No conozco la respuesta

Procesamiento normal en una ventana por lotes durante fuera de línea.

Requisitos complejos no funcionales:

Generalmente procesa grandes volúmenes de información.

Aplicación por lotes: estrategias de procesamiento

Otro:

Respuesta correcta

Generalmente procesa grandes volúmenes de información.

Sin interacción (directa) del usuario.

Procesamiento normal en una ventana por lotes durante fuera de línea.

Requisitos complejos no funcionales:

trazabilidad de la información y ejecución.

Rendimiento / procesamiento de alta velocidad.

Aplicación por lotes: estrategias de procesamiento

¿ Que se considera retro compatible ?*

1/1

Modificar el código para que sea compatible

Agregar un nuevo método de ejecución a la biblioteca, Modificar la lógica interna de un método de ejecución

No conozco la respuesta

Versionar el código

Que es Identificador único (UID)*

0/1

Identificador único de cada DTO

Identificador único de cada de cada componente en APX

No conozco la respuesta

Identificador único de cada Batch

Identificador único de cada transacción
Identifica de forma única una ejecución

Respuesta correcta

Identifica de forma única una ejecución

¿Qué sucede cuando se implementa una nueva versión de una dependencia en una biblioteca?*

0/1

No encuentra las dependencias
No tiene impacto en nuestra biblioteca en el momento de la ejecución
No conozco la respuesta
Errores en tiempo de ejecución

No carga las dependencias correctas

Respuesta correcta

No tiene impacto en nuestra biblioteca en el momento de la ejecución

¿Está permitida invocación sincrónica?*

0/1

Dependiendo el desarrollo
No conozco la respuesta
SI

No se
No se permite la invocación sincrónica

Respuesta correcta

No se permite la invocación sincrónica

Máximo de resultados por paginación*

0/1

5000
1000

10000
2000

1

No conozco la respuesta

Respuesta correcta

5000

¿Qué es APX?*

1/1

Es una extensión de Mainframe sustituyendola

La Arquitectura APX o Arquitectura Backend Java Extendida fue creada con el objetivo de ser una extensión de la Arquitectura Backend proporcionando las mismas capacidades en el mundo distribuido.

No conozco la respuesta

Es una forma de poder sustituir al Mainframe para ejecutar procesos mas rapidos

Ofrece una alternativa confiable para el desarrollo de transacciones que no está acoplada al canal pero depende del Mainframe

Agregue un campo de entrada opcional. Modifique un campo de entrada obligatorio y configúrelo como opcional. Modifique un campo de salida opcional y hágalo obligatorio. Lógica interna. son características de un cambio ...*

0/1

Versionado

No retrocompatibles

Cambio en desarrollo

No conozco la respuesta

Retrocompatibles

Respuesta correcta

Retrocompatibles

¿Qué es Batch?*

1/1

Es un patrón de ejecución, ejecución de un programa sin el control o supervisión directa del usuario

No conozco la respuesta

Proceso que varia con respecto a sus dependencias

Proceso que reemplaza al mainframe

Proceso idéntico a online

Otro:

¿Qué es el `LOGGER.debug` ?*

0/1

Imprime advertencias o fallas no críticas de la aplicación, que no impiden que la aplicación continúe su flujo

Imprime mensajes de error críticos que impiden que la aplicación funcione correctamente. Estos son los registros habilitados en Producción

Imprime información con el máximo nivel de detalle.

No conozco la respuesta

Imprime información útil que le permite ver el progreso de la aplicación.

Le permite imprimir datos informativos con detalles finos que son útiles para depurar la aplicación.

Respuesta correcta

Le permite imprimir datos informativos con detalles finos que son útiles para depurar la aplicación.

¿Por que no se recomiendan las transacciones asíncronas?*

1/1

No hay implementación de transacciones asincronas

Se pierde el control de la ejecución

No se permite en Mexico

OSGI no reconoce las transacciones como dependencias

No conozco la respuesta

Su lentitud de ejecución

En caso de que una transacción deba invocar a otra, solo se puede realizar de forma*

0/1

Dependiendo el desarrollo

Sincrónica

Con un conector

Asincrónica

No conozco la respuesta

Respuesta correcta

Asincrónica

El árbol de dependencia en la invocación a las bibliotecas debe ser finito, eviten profundidades superiores a*

1/1

2

9

1

No conozco la respuesta

3

El acceso a datos se hace mediante ¿Qué componentes en online ?*

0/1

DTOs

No conozco la respuesta

Transacciones

Bibliotecas

Jobs

Respuesta correcta

Bibliotecas

Consideraciones para tener en cuenta para la gestión de errores*

0/1

APX no permite la gestión de errores

No hay consideraciones

Permite lanzar errores en un bloque try .. catch

No conozco la respuesta

Cada UUAA tiene un rango de código de error asignado.

El manejo de errores se lleva a cabo a través de la consola de operación APX actual

Respuesta correcta

Cada UUAA tiene un rango de código de error asignado.

El manejo de errores se lleva a cabo a través de la consola de operación APX actual

Excepciones de Java permitida*

1/1

org.springframework.web.client.RestClientException

No conozco la respuesta

java.lang.NumberFormatException

com.bbva.apx.exception.db.NoResultException

com.bbva.elara.utility.interbackend.cics.exceptions.BusinessException

En que ayuda un DTO*

1/1

Ayuda a identificar los métodos que se repiten en componentes APX

El uso de este patrón ayuda a identificar las agrupaciones de datos simples y/o complejos, y evita duplicidad de código.

No conozco la respuesta

Ayuda a agrupar métodos de un componente

En tiempo de ejecución, las dependencias con otras bibliotecas se resuelven en función de los paquetes declarados en la sección*

1/1

No conozco la respuesta

Test

Abstract

dependency

import-package

La Arquitectura es la encargada de realizar la gestión de commit para garantizar la transaccionalidad*

0/1

No conozco la respuesta

No

Es posible desde la consola de configuracion

Si

Es posible desde las librerias

Respuesta correcta

Si

Que es un DTO*

1/1

Un objeto de transferencia de datos es, transporta datos entre procesos

Es un objeto de agrupación de componentes APX

Agrupar métodos de un componente

No conozco la respuesta

Una UUAA, ¿ puede acceder a los datos que posee y datos de terceros a través de bibliotecas APX ?*

1/1

Sí

Tal vez

No

No conozco la respuesta

Cuántas veces es posible invocar Mainframe*

0/1

No conozco la respuesta

Sin restricción

2

No es posible invocar Mainframe

1

Respuesta correcta

2

Las dependencias circulares que ocasionan ...*

1/1

Emporar la comunicación de los Bundles

Que Osgi no cargue estas dependencias correctamente

Mejorar el acceso a los componentes

Depende de la necesidad del proyecto

No conozco la respuesta

¿Qué es el LOGGER.error ?*

1/1

Imprime información útil que le permite ver el progreso de la aplicación.

Imprime información con el máximo nivel de detalle.

No conozco la respuesta

Le permite imprimir datos informativos con detalles finos que son útiles para depurar la aplicación.

Imprime advertencias o fallas no críticas de la aplicación, que no impiden que la aplicación continúe su flujo

Imprime mensajes de error críticos que impiden que la aplicación funcione correctamente.

Estos son los registros habilitados en Producción

Batch Execution Banking basada en Spring Batch, ¿características que proporciona ?*

0/1

Soporte multi-DB usando fuentes de datos configurables

Opción de compromisos parciales.

No conozco la respuesta

Control del segmento de ejecución.

No permite la ejecución de ciertos comandos de eliminación del sistema.

Lectores y escritores listos para usar y personalizables.

El proceso por lotes se reinicia en caso de error.

Acceso a utilidades y recursos del sistema.

Lectura / Escritura de múltiples formatos de archivo (TXT, CSV, etc.).

Integración con planificadores: BMC Control-M, Tivoli OPC, Quartz.

Los lectores acceden a las bases de datos en modo "solo lectura".

Otro:

Respuesta correcta

Integración con planificadores: BMC Control-M, Tivoli OPC, Quartz.

Opción de compromisos parciales.

El proceso por lotes se reinicia en caso de error.

Soporte multi-DB usando fuentes de datos configurables

Lectores y escritores listos para usar y personalizables.
Acceso a utilidades y recursos del sistema.
Lectura / Escritura de múltiples formatos de archivo (TXT, CSV, etc.).
Control del segmento de ejecución.
Los lectores acceden a las bases de datos en modo "solo lectura".
No permite la ejecución de ciertos comandos de eliminación del sistema.

El multiproceso no debe usarse en APX Batch ¿Que se puede hacer como opción?*

0/1

No es parte de Spring
No conozco la respuesta
Varios procesos por lotes en diferentes tiempos

No Ejecutar procesos
Varios procesos por lotes al mismo tiempo
Otro:

Respuesta correcta

Varios procesos por lotes al mismo tiempo

Las bibliotecas pueden definir su propia _____ para determinar qué UUAA externas pueden usar sus bibliotecas.*

1/1

Autoconsumo
De quien las consume
De quien serán dependencias
Visibilidad

Gestión de errores
No conozco la respuesta

¿Para que se usa el JPA?*

0/1

Permite manejar las utilidades de APX (Datio, JDBC,...)
Permite usar componentes APX con cualquier BBDD
Permite realizar un almacenamiento en caché de tablas de BBDD
Permite insertar, actualiza y consultar en BBDD

No conozco la respuesta

Respuesta correcta

Permite realizar un almacenamiento en caché de tablas de BBDD

JPA, ¿ es permitido usarlo en APX ?*

1/1

No conozco la respuesta

No

Si

Tal vez

¿Que es Document Manager?*

1/1

Utilidad para manejar Documentos

No conozco la respuesta

Utilidad para crear Documentos

Utilidad que esta restringida para su uso

Utilidad que proporciona acceso a los servicios de gestión de documentos externos a las aplicaciones

Utilidad para organizar documentos

Una transacción o librería, dentro de su lógica, debe evitar invocar más de ____ librerías*

0/1

3

1

9

2

No conozco la respuesta

Respuesta correcta

9

¿Por que no se recomiendan las transacciones asíncronas?*

1/1

Su lentitud de ejecución

No hay implementación de transacciones asincronas

No conozco la respuesta
OSGI no reconoce las transacciones como dependencias
No se permite en Mexico
No sabrá si ha concluido con un commit OK o con un OK retroceso.

APX, ¿Qué es Escritura ROP?*

1/1

Es un Protocolo HTTP
Es un Protocolo JMS
Es un Protocolo REST
Escritura en Archivos
No conozco la respuesta
Un LOG interno utilizado con fines de Auditoría, Fraude o Informativo que indica que la transacción ha finalizado

¿Qué método se usa para agregar avisos desde una Librería o transacción?*

0/1

addAvisosAPX
addSeverity

addAdvice
No conozco la respuesta
setSeverity

Respuesta correcta
addAdvice

Examen teórico APX

¿Que causa no corresponde al antipatron Contenedor Magico?*

1/1

Reparto de responsabilidades incorrecto
Falta de cumplimiento de la arquitectura
Intervención demasiado limitada
Desacoplamiento entre librerías

¿Quién es el responsable de iniciar un Job?

*

1/1

JobLauncher

JobInstance

JobBegginer

JobExecution

Selecciona las tres respuestas que describen las mejores practicas establecidas para acceder a Datos*

1/1

En el caso de procesos Batch no deben existir cesiones de ficheros entre UUAA's

Una UUAA puede acceder a datos de terceros a través de las librerías APX que el dueño del dato proporcione

No se debe manejar el objeto Datasource directamente

Una UUAA solo puede acceder a los datos que ella misma custodia

Las partes de un campo variable definidos por la cadena NMM son:*

1/1

N es el orden de la variable literal (1 o 4) , MM la longitud maxima del campo informado en la variable (15 es el maximo)

N es el orden de la variable literal (1 y 4) , MM la longitud maxima del campo informado en la variable (15 es el maximo)

N es el orden de la variable literal (1 o 2) , MM la longitud maxima del campo informado en la variable (15 es el maximo)

N es el orden de la variable literal (1 o 2) , MM la longitud maxima del campo informado en la variable (20 es el maximo)

El objeto _____ no debe manejarse directamente. No debe crear instancias , abrir o cerrar explicitamente conexiones a la base de datos*

1/1

Datasource

JPA
JDBC
Database

¿Cuál es el prefijo que debe utilizarse en la nomenclatura de los métodos que exponen las librerías para ser consumidos?

*

1/1

dependency[[Nombre_elegido_por_app] donde [Nombre_elegido_por_app] es opcional
execute[[Nombre_elegido_por_app] donde [Nombre_elegido_por_app] es opcional

execute[[Nombre_elegido_por_app] donde [Nombre_elegido_por_app] es obligatorio
dependency[[Nombre_elegido_por_app] donde [Nombre_elegido_por_app] es obligatorio

Para garantizar el comportamiento transaccional de APX, la arquitectura se basa en el estándar ____, por lo que los recursos administrados dentro de la transacción deben ser del tipo XA.

*

1/1

JT3
JTX
JPA
JTA

En las consultas a BBDD debe usar variables _____ para no afectar el rendimiento de la base de datos

*

1/1

Miembro
Bind

Locales
Dynamic

En un Batch, ¿Cual es el step donde el desarrollador introducirá la logica empresarial necesaria?*

0/1

itemWriter

Task

Chunk

JobLauncher

Respuesta correcta

Task

El identificador unico para la ejecución de transacciones/Job se genera automaticamente como ultimo paso cuando se invoca las transacciones en linea/Batch. El resultado de la transaccion se puede propagar para la correlación en las otras capas.*

1/1

Falso

Verdadero

Un _____ proporciona una solución a un problema de diseño. Debe cumplir con diferentes características, como la efectividad al resolver problemas similares en ocasiones anteriores. Por lo tanto, debe de ser re-utilizable, es decir, aplicable a diferentes problemas en diferentes circunstancias.

*

1/1

Anti-patrón

Metodología

Patrón de diseño

Reutilizando código

Para enviar un aviso desde una biblioteca o desde la transacción, cualquiera que sea el tipo, el método utilizado es _____.

*

0/1

addError
addWarning

addAdvice
addSeverity

Respuesta correcta

addAdvice

Cualquier documento utilizado (.dll, .txt, etc.) se almacenará en el sistema de archivos compartidos del nodo Batch (generalmente _____) y luego se eliminará periódicamente.

*

1/1

/datentJobs/datent.
/temporalFiles/datent.
/fichtemcomp/datent.

/fichtemporal/datent.

Los componentes de seguridad lógica que dan servicio a la arquitectura no están acoplados a la Arquitectura APX.

*

1/1

Verdadero

Falso

La información utilizada en la lógica de negocio de la transacción debe incluirse en

_____.

*

1/1

Los archivos locales.
Las variables globales.
Los parámetros de entrada.

Las variables locales.

¿Cuál definición de APX es incorrecta?

*

0/1

Extended Programming Architecture
Backend Extended Platform
Extended Java Backend Architecture

Extended Platform Architecture

Respuesta correcta

Extended Platform Architecture

Las utilidades APX implementadas tanto para la arquitectura Online como para la arquitectura Batch son:

*

0/1

Communication Manager: Se crea una librería con la capacidad de invocar a G.U.C para el envío de notificaciones. Interbackend proxy: Se crea una librería con la capacidad de invocar a los conectores IMS (para acceso a Host).

Compresión/Descompresión: Se crea una librería con la capacidad de comprimir y descomprimir archivos en zip. Generador de documentos: Se crea una librería con la capacidad de generar documentos.

Respuesta correcta

Compresión/Descompresión: Se crea una librería con la capacidad de comprimir y descomprimir archivos en zip. Generador de documentos: Se crea una librería con la capacidad de generar documentos.

De la lista siguiente marca 2 funcionalidades de servicios persistencia incluidas en APX

*

0/1

SFTP

PS9

JDBC

JMS/RABBIT

Two-Phase Commit

JNDI

Respuesta correcta

JDBC

Two-Phase Commit

De la lista siguiente marca 2 funcionalidades de servicios conectores incluidos en APX

*

0/1

Gestor de Eventos

HTTP/HTTPs

JDBC

JMS/RABBIT

Two-Phase Commit

JNDI

Respuesta correcta

Gestor de Eventos

HTTP/HTTPs

Patrones de diseño en APX

*

0/1

Paginación en Librerías

Paginación en Transacciones

CRUD en Librerías

DTO en componentes APX

Respuesta correcta

Paginación en Librerías

Paginación en Transacciones
CRUD en Librerías
DTO en componentes APX

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

*

1/1

Pila de ejecución
Solicitud
Transacción

Ejecución

¿Cuáles son los campos que definen a una transacción en APX?

*

0/1

Aplicación, versión, país, paquete descripción
Aplicación, dominio, país, paquete descripción
Aplicación, transacción, versión, país, descripción
Aplicación, transacción, dominio, versión, país, descripción

Respuesta correcta

Aplicación, transacción, versión, país, descripción

¿Cuáles son los protocolos físicos disponibles en APX Online?

Selecciona 2

*

1/1

HTTP/ SOAP

HTTP/ REST / RESTFul (A través de Servicios Backend)

AQPG (Formato Arquitectura Plataforma QP05: síncrono o asíncrono)

PS9

Este tipo de paginación es el que establece Arquitectura y Calidad como el más eficiente, tanto en consumo de recursos como en el tiempo de proceso para la consecución de la información, lo que redunda en un menor tiempo de espera por parte del usuario final.

*

1/1

Paginación en Librería

Paginación en Transaccion

Paginación en Batch

Paginación en DTO

Este patrón suele representarse como una única librería que contiene la mayoría de la funcionalidad y que otras librerías utilizan incorrectamente de apoyo. La clave del problema es que puede contener un porcentaje muy elevado de responsabilidades dentro de la solución a diseñar

*

0/1

Contenedor Magico

Blob

CRUD

Respuesta correcta

Blob

¿Cómo solucionamos un Blob?

*

1/1

Mover la funcionalidad a la transacción reduciendo la carga en la librería.

Mover el comportamiento a sus librerías relacionadas, reduciendo el acoplamiento entre librerías y simplificando los mantenimientos.

Moviendo la lógica a diferentes métodos y clases dentro de la misma librería

Mover el comportamiento a DTOs relacionados, reduciendo el acoplamiento entre los componentes y simplificando los mantenimientos

Selecciona dos aplicaciones que justifican el uso del patrón Data Transfer Object (DTO) es obligatorio cuando:

*

1/1

Se necesitan mapear demasiadas variables

Se necesita una agrupación simple de datos basada en una funcionalidad asociada

Desea intercambiar información entre diferentes componentes en APX de manera coherente, organizada y agrupada

Se hace conexión a Bases de datos relacionales

Este patrón suele representarse en una librería que contiene un solo método “execute” con parámetros que actúan como filtros y que engloba diversas funcionalidades dispares en función de sus parámetros de entrada

*

1/1

Contenedor Mágico

CRUD

Paginacion

Blob

¿Qué causas no corresponden al patrón Contenedor Mágico?

*

1/1

Reparto de responsabilidades incorrecto

Falta de cumplimiento de la arquitectura

Intervención demasiado limitada

Desacoplamiento entre librerías

La aplicación de este anti patrón de diseño en los desarrollos, causará la búsqueda y análisis de una mejor solución a un problema por medio de reingeniería

*

1/1

Paginación en transacciones
Contenedor mágico

Paginación Mágica
Generador de Logs

¿Qué nivel de profundidad es permitido al realizar invocaciones de librería a librería?

*

0/1

Profundidades a 5 niveles: Transacción > Librería > Librería > Librería> Librería > Librería

Profundidades a 7 niveles: Transacción > Librería > Librería > Librería> Librería > Librería> Librería > Librería

Profundidades a 3 niveles: Transacción > Librería > Librería > Librería

Profundidades a 9 niveles: Transacción > Librería > Librería > Librería> Librería > Librería> Librería > Librería> Librería > Librería

Respuesta correcta

Profundidades a 3 niveles: Transacción > Librería > Librería > Librería

¿Cuál es el número máximo de librerías a invocar desde una transacción o librería?

*

0/1

3 librerías

7 librerías

9 librerías

12 librerías

Respuesta correcta

9 librerías

No puede incluir _____ o etiquetas similares. Estas anotaciones no reciben errores ni advertencias en el compilador en el momento de la compilación.

*

1/1

@BlockWarnings

@SuppressWarnings

@LeaveWarnings
@ForbiddenWarnings

Está prohibido poner las siguientes anotaciones de primavera en las clases de implementación de las bibliotecas APX: @Component, @Service, @Controller y @Repository, porque estas anotaciones causan conflictos en la resolución de classpath del paquete de la biblioteca y tienen el efecto de que la biblioteca no expone su servicio OSGI.

*

1/1

@Component, @Service, @Controller & @Bean
@Component, @Override, @Controller & @Repository
@Component, @Service @Controller & @Repository

@Component, @Override, @Controller & @Configuration

Niveles de Logs más comunes dentro de APX

*

0/1

INFO, WARNING & ERROR.
TRACE, DEBUG, INFO, WARNING & ERROR.

DEBUG, TRACE WARNING & ERROR
DEBUG, INFO, WARNING & ERROR

Respuesta correcta

DEBUG, INFO, WARNING & ERROR

Para información de muy bajo nivel, solo útil para depurar la aplicación tanto en el desarrollo como en el análisis de incidentes ¿A qué tipo de nivel de Log se refiere?

*

0/1

Trace

Info
Debug

Respuesta correcta

Debug

Este nivel de Log tiene las siguientes características:

- Paradas e inicios de servicios y sistemas.
- Parámetros críticos o relevantes de configuración.
- Inicio y fin de transacciones y operaciones completas.
- Cambios del estado de las operaciones

*

1/1

Warning

Error

Info

Trace

DEBUG, sirve para información de muy bajo nivel solo útil para el debug de la aplicación, tanto en el desarrollo como en el análisis de incidencias

- Paradas y arranques de servicios y sistemas
- Parámetros críticos o relevantes de configuración
- Comienzo y fin de transacciones y operaciones completas
- Cambios de estado de operaciones

*

0/1

Verdadero

Falso

Respuesta correcta

Falso

WARN, son situaciones que impiden la ejecución correcta de una operación o transacción, pero sin afectar a otras operaciones o transacciones (error aislado o contenido)

*

1/1

Verdadero

Falso

ERROR, información de situaciones, que aún sin ser de error, si son anómalas o no previstas, aunque el aplicativo tiene alternativas para solventarlas

*

1/1

Verdadero

Falso

INFO, información de nivel superior que permita hacer un seguimiento de la ejecución normal

*

1/1

Verdadero

Falso

Las partes de un campo variable definidos por la cadena _\$NMM\$ son:

*

1/1

(N es el orden de la variable literal (1 o 2), MM la longitud máxima del campo informado en la variable (20 es el máximo)

(N es el orden de la variable literal (1 a 4), MM la longitud máxima del campo informado en la variable (15 es el máximo)

(N es el orden de la variable literal (1 o 2), MM la longitud máxima del campo informado en la variable (15 es el máximo)

_____ fue creada para brindar una herramienta a los desarrolladores y administradores del entorno para gestionar las operaciones en la plataforma

*

1/1

Consola de Operaciones APX

Monitor APX
Atenea
Kibana

La comunicación de APX a otros servicios NO APX se realiza UNICAMENTE a través de la utilidad APX _____

*

1/1

Certificados
Conector API genérico (APIConnector)

Conectores
JDBC

No está permitido generar _____ para escribir de Logs. Tampoco se permite la escritura por _____, debe ser la arquitectura quien los escriba.

*

1/1

Archivos propios / sysout

Archivos de control / system
Archivos de lectura / sysout

¿Cuáles son los protocolos lógicos disponibles en APX?

Selecciona 2

*

0/1

HTTP/ SOAP

PS9

HTTP/ REST / RESTFul (A través de Servicios Backend)

AQPG (Formato Arquitectura Plataforma QP05: síncrono o asíncrono)

Respuesta correcta

PS9

AQPG (Formato Arquitectura Plataforma QP05: síncrono o asíncrono)

¿Cuáles son las acciones posteriores a la ejecución de la lógica de negocio en la capa de Arquitectura de control?

*

1/1

Validation of Exit Code, Commit Control, Electronic Journal, Business Event Notification, Monitorization

Validation of Exit Code, Commit Control, ROP Writing, Business Event Notification, Monitorization

Validation of Exit Code, Commit Control, ROP Writing, Operations Registry, Monitorization
Validation of Exit Code, Commit Control, ROP Writing, Business Event Notification, Unique Identifier.

Selecciona las únicas excepciones que pueden ser capturadas y administradas por las aplicaciones, típicas de APX:

*

0/1

com.bbva.apx.exception.business.NullPointerException

com.bbva.apx.exception.business.IOException

com.bbva.apx.exception.io.network.TimeoutException

com.bbva.apx.exception.business.BusinessException

Respuesta correcta

com.bbva.apx.exception.io.network.TimeoutException

com.bbva.apx.exception.business.BusinessException

¿Cuáles son los dos tipos de step que existen en Batch APX?

*

1/1

Creator, Processor

Processor,Task

Chunk, Reader

Chunk,Task

Agrega un campo de entrada obligatorio. Agregue un campo de salida. Modifique el tipo de un campo de entrada o salida existente. Son características de un cambio...

*

1/1

Cambio en desarrollo
No retrocompatibles

Retrocompatibles
Versionado

Es obligatorio que los DTOs se implementen dentro de un....

*

1/1

empaquetado independiente (Bundle Batch)
empaquetado independiente (Bundle DTO)

empaquetado independiente (Bundle TX)
empaquetado independiente (Bundle LIB)

Las dependencias circulares son permitidas...

*

1/1

Algunas son permitidas
Depende de la necesidad del proyecto
Son permitidas
Están absolutamente prohibidas

El multiproceso no debe usarse en APX Batch ¿Que se puede hacer como opción?

*

1/1

Varios procesos por lotes en diferentes tiempos
No conozco la respuesta
No ejecutar procesos
Varios procesos por lotes al mismo tiempo

Excepción de Spring permitida

*

1/1

java.lang.NumberFormatException
org.springframework.web.client.RestClientException

com.bbva.elara.utility.interbackend.cics.exceptions.BusinessException
org.springframework.web.client.NumberFormatException

Diferencias entre Las invocaciones de APX Batch e invocaciones de APX Online

*

1/1

Los encabezados del protocolo lógico a enviar

Ninguno
Permitirá ejecutar transacciones de cics solo por protocolo ps10.
Permite la posibilidad de invocar un batch de APX directamente
Horarios de ejecución

APX, ¿Qué es Escritura ROP?

*

1/1

Es un Protocolo HTTP
Es un Protocolo JMS
Un LOG interno utilizado con fines de Auditoría, Fraude o Informativo que indica que la transacción ha finalizado

Es un Protocolo REST
Escritura en Archivos

En caso de que una transacción deba invocar a otra, solo se puede realizar de forma...

*

1/1

Dependiendo el desarrollo
Sincrónica
Con un conector
Asincrónica

Capacidades de APX

*

0/1

Implementaciones en caliente de acceso a datos

Procesamiento transaccional

Integración con servicios de seguridad

Monitoreo de procesamiento por lotes de servicios bancarios

Interoperabilidad de backends

Respuesta correcta

Implementaciones en caliente de acceso a datos

Procesamiento transaccional

Integración con servicios de seguridad

Monitoreo de procesamiento por lotes de servicios bancarios

Interoperabilidad de backends

¿Qué característica NO pertenece al BATCH?

*

1/1

Procesamiento normal en una ventana de lote fuera de línea

Interacción con usuario (directo)

Procesamiento generalmente grande volúmenes de información

Requisitos no funcionales complejos

Existe un API de código de barras

*

1/1

Solo en España

Su uso es restringido

Si

No

NO se deben utilizar sentencias que incluyan atributos o acciones específicas de un administrador de base de datos en particular, como lo es...

*

1/1

OR
SELECT
FROM
ROWNUM

¿Qué es un Chunk?

*

1/1

Realiza una operación dentro del paso (por ejemplo, invocando utilidades de arquitectura)
Conjunto de procesos
Conjuntos de datos de entrada se procesan en la misma transacción

Conjunto de readers

JAVA

15 de 23 puntos

Examen práctico Java

Which three are bad practices?

*

1/1

Checking for an IOException and ensuring that the program can recover if one occurs.

Checking for ArrayIndexOutOfBoundsException and ensuring that the program can recover if one occurs.

Checking for FileNotFoundException to inform a user that a filename entered is not valid.

Checking for Error and, if necessary, restarting the program to ensure that users are unaware problems.

Checking for ArrayIndexOutOfBoundsException when iterating through an array to determine when all elements have been visited.

Indica a JUnit que la propiedad que usa esta anotación es una simulación y, por lo tanto, se inicializa como tal y es susceptible de ser inyectada por @InjectMocks.

*

1/1

Mockito
Mock

Inject
InjectMock

*

1/1

Given

```
public static void main(String[] args){  
    int[][] array2D = {{0,1,2}, {3,4,5,6}};  
    System.out.print(array2D[0].length + "");  
    System.out.print(array2D[1].getClass().isArray() + "");  
    System.out.print(array2D[0][1]);  
}
```

What is the result?

3false3
3false1
2false1
3true1

2true3

Which two statments are true?

*

1/1

An interface CANNOT be extended by another interface.

An abstract class can be extended by a concrete class.

An abstract class CANNOT be extended by an abstract class.

An interface can be extended by an abstract class.

An abstract class can implement an interface.

An abstract class can be extended by an interface.

*

1/1

Given:

```
class Alpha{ String getType(){ return "alpha";}}
class Beta extends Alpha{String getType(){ return "beta";}}
public class Gamma extends Beta { String getType(){ return "gamma";}
    public static void main(String[] args) {
        Gamma g1 = (Gamma) new Alpha();
        Gamma g2 = (Gamma) new Beta();
        System.out.print(g1.getType()+ " " +g2.getType());
    }
}
```

What is the result?

Gamma gamma
Beta beta
Alpha beta
Compilation fails

Which five methods, inserted independently at line 5, will compile? (Choose five)

*

1/1

```
1 public class Blip{
2     protected int blipvert(int x){ return 0
3 }
4 class Vert extends Blip{
5     //insert code here
6 }
```

Private int blipvert(long x) { return 0; }

Protected int blipvert(long x) { return 0; }

Protected long blipvert(int x, int y) { return 0; }

```
Public int blipvert(int x) { return 0; }
```

```
Private int blipvert(int x) { return 0; }
```

```
Protected long blipvert(int x) { return 0; }
```

```
Protected long blipvert(long x) { return 0; }
```

*

0/1

Given:

```
1. class Super{  
2.     private int a;  
3.     protected Super(int a){ this.a = a; }  
4. }
```

...

```
11. class Sub extends Super{  
12.     public Sub(int a){ super(a);}  
13.     public Sub(){ this.a = 5;}  
14. }
```

Which two independently, will allow Sub to compile? (Choose two)

Change line 2 to: public int a;

Change line 13 to: public Sub(){ super(5);}

Change line 2 to: protected int a;

Change line 13 to: public Sub(){ this(5);}

Change line 13 to: public Sub(){ super(a);}

Respuesta correcta

Change line 13 to: public Sub(){ super(5);}

Change line 13 to: public Sub(){ this(5);}

Whats is true about the class Wow?

*

1/1


```

public abstract class Wow {
    private int wow;
    public Wow(int wow) { this.wow = wow; }
    public void wow() { }
    private void wowza() { }
}

```

It compiles without error.

It does not compile because an abstract class cannot have private methods

It does not compile because an abstract class cannot have instance variables.

It does not compile because an abstract class must have at least one abstract method.

It does not compile because an abstract class must have a constructor with no arguments.

What is the result?

*

1/1

```

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(); }
}

```

Compilation fails.

Atom granite.

Granite granite.

Atom granite granite.

An exception is thrown at runtime.

Atom granite atom granite.

What is printed out when the program is excuted?

*

1/1

```
public class MainMethod {  
    void main() {  
        System.out.println("one");  
    }  
    static void main(String args) {  
        System.out.println("two");  
    }  
    public static final void main(String[] args) {  
        System.out.println("three");  
    }  
    void mina(Object[] args) {  
        System.out.println("four");  
    }  
}
```

one
two
three

four
There is no output.

What is the result?

*

0/1

```

class Feline {
    public String type = "f ";
    public Feline() {
        System.out.print("feline ");
    }
}
public class Cougar extends Feline {
    public Cougar() {
        System.out.print("cougar ");
    }
    void go() {
        type = "c ";
        System.out.print(this.type + super.type);
    }
    public static void main(String[] args) {
        new Cougar().go();
    }
}

```

Cougar c f.
 Feline cougar c c.
 Feline cougar c f.

Compilation fails.

Respuesta correcta

Feline cougar c c.

What is the result?

*

1/1

```

class Alpha { String getType() { return "alpha"; } }
class Beta extends Alpha { String getType() { return "beta"; } }
public class Gamma extends Beta { String getType() { return "gamma"; }
    public static void main(String[] args) {
        Gamma g1 = new Alpha();
        Gamma g2 = new Beta();
        System.out.println(g1.getType() + " " + g2.getType());
    }
}

```

Alpha beta
 Beta beta.
 Gamma gamma.
 Compilation fails.

What is the result?

*

1/1

```

import java.util.*;
public class MyScan {
    public static void main(String[] args) {
        String in = "1 a 10 . 100 1000";
        Scanner s = new Scanner(in);
        int accum = 0;
        for (int x = 0; x < 4; x++) {
            accum += s.nextInt();
        }
        System.out.println(accum);
    }
}

```

11
 111

1111

An exception is thrown at runtime.

What is the result?

*

1/1

```
public class Bees {  
    public static void main(String[] args) {  
        try {  
            new Bees().go();  
        } catch (Exception e) {  
            System.out.println("thrown to main");  
        }  
    }  
    synchronized void go() throws InterruptedException {  
        Thread t1 = new Thread();  
        t1.start();  
        System.out.print("1 ");  
        t1.wait(5000);  
        System.out.print("2 ");  
    }  
}
```

The program prints 1 then 2 after 5 seconds.

The program prints: 1 thrown to main.

The program prints: 1 2 thrown to main.

The program prints: 1 then t1 waits for its notification.

Which statement is true?

*

1/1

```

class ClassA {
    public int numberOfInstances;
    protected ClassA(int numberOfInstances) {
        this.numberOfInstances = numberOfInstances;
    }
}
public class ExtendedA extends ClassA {
    private ExtendedA(int numberOfInstances) {
        super(numberOfInstances);
    }
    public static void main(String[] args) {
        ExtendedA ext = new ExtendedA(420);
        System.out.print(ext.numberOfInstances);
    }
}

```

420 is the output.

An exception is thrown at runtime.
 All constructors must be declared public.
 Constructors CANNOT use the private modifier.
 Constructors CANNOT use the protected modifier.

The SINGLETON pattern allows:

*

0/1

Have a single instance of a class and this instance cannot be used by other classes
 Having a single instance of a class, while allowing all classes have access to that instance.
 Having a single instance of a class that can only be accessed by the first method that calls it.

Respuesta correcta

Having a single instance of a class, while allowing all classes have access to that instance.

What is the result?

*

0/1

```
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);
        for (String s : sa) { System.out.println(nf.parse(s)); }
    }
}
```

111.234 222.567
111.234 222.568

111.234 222.5678
An exception is thrown at runtime.

Respuesta correcta
111.234 222.5678

What is the result?

*

0/1

Given

```
public class SuperTest {
    public static void main(String[] args) {
        //statement1
        //statement2
        //statement3
    }
}

class Shape {
    public Shape() {
        System.out.println("Shape: constructor");
    }
    public void foo() {
        System.out.println("Shape: foo");
    }
}

class Square extends Shape {
    public Square() {
        super();
    }
    public Square(String label) {
        System.out.println("Square: constructor");
    }
    public void foo() {
        super.foo();
    }
    public void foo(String label) {
        System.out.println("Square: foo");
    }
}
```

What should statement1, statement2, and statement3, be respectively, in order to produce the result?

Shape: constructor
Shape: foo
Square: foo

Square square = new Square ("bar"); square.foo ("bar"); square.foo();
Square square = new Square ("bar"); square.foo ("bar"); square.foo ("bar");
Square square = new Square (); square.foo (); square.foo(bar);
Square square = new Square (); square.foo (); square.foo("bar");
Square square = new Square (); square.foo (); square.foo ();

Respuesta correcta

Square square = new Square (); square.foo (); square.foo("bar");

Which three implementations are valid?

*

0/1


```
interface SampleCloseable {  
    public void close() throws java.io.IOException;  
}
```

```
class Test implements SampleCloseable { public void close() throws java.io.IOException { // do something } }
```

```
class Test implements SampleCloseable { public void close() throws Exception { // do something } }
```

```
class Test implements SampleCloseable { public void close() throws FileNotFoundException { // do something } }
```

```
class Test extends SampleCloseable { public void close() throws java.io.IOException { // do something } }
```

```
class Test implements SampleCloseable { public void close() { // do something } }
```

Respuesta correcta

```
class Test implements SampleCloseable { public void close() throws java.io.IOException { // do something } }
```

```
class Test implements SampleCloseable { public void close() throws FileNotFoundException { // do something } }
```

```
class Test implements SampleCloseable { public void close() { // do something } }
```

What is the result?

*

1/1

```

class MyKeys {
    Integer key;
    MyKeys(Integer k) { key = k; }
    public boolean equals(Object o) {
        return ((MyKeys) o).key == this.key;
    }
}

```

And this code snippet:

```

Map m = new HashMap();
MyKeys m1 = new MyKeys(1);
MyKeys m2 = new MyKeys(2);
MyKeys m3 = new MyKeys(1);
MyKeys m4 = new MyKeys(new Integer(2));
m.put(m1, "car");
m.put(m2, "boat");
m.put(m3, "plane");
m.put(m4, "bus");
System.out.print(m.size());

```

2
3
4

Compilation fails.

What value of x, y, z will produce the following
result?

1234,1234,1234 -----, 1234, -----

*

0/1

```

public static void main(String[] args) {
    // insert code here
    int j = 0, k = 0;
    for (int i = 0; i < x; i++) {
        do {
            k = 0;
            while (k < z) {
                k++;
                System.out.print(k + " ");
            }
            System.out.println(" ");
            j++;
        } while (j < y);
        System.out.println("---");
    }
}

```

int x = 4, y = 3, z = 2;
 int x = 3, y = 2, z = 3;
 int x = 2, y = 3, z = 3;
 int x = 2, y = 3, z = 4;
 int x = 4, y = 2, z = 3;

Respuesta correcta

int x = 2, y = 3, z = 4;

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

*

0/1

```

13. public class Speak {
14.     public static void main(String[] args) {
15.         Speak speakIT = new Tell();
16.         Tell tellIt = new Tell();
17.         speakIT.tellItLikeltIs();
18.         (Truth) speakIT.tellItLikeltIs();
19.         ((Truth) speakIT).tellItLikeltIs();
20.         tellIt.tellItLikeltIs();
21.         (Truth) tellIt.tellItLikeltIs();
22.         ((Truth) tellIt).tellItLikeltIs();
23.     }
24. }

```

```

class Tell extends Speak implements Truth {
    @Override
    public void tellItLikeltIs() {
        System.out.println("Right on!");
    }
}

```

```

interface Truth {
    public void tellItLikeltIs();
}

```

Line 17

Line 18

Line 19

Line 20

Line 21

Line 22

Respuestas correctas

Line 19

Line 20

Line 22

What is the result?

*

1/1

```

class Feline {
    public String type = "f";
    public Feline() {
        System.out.print(s: "feline ");
    }
}

public class Cougar extends Feline{
    public Cougar() {
        System.out.print(s: "cougar ");
    }
    void go(){
        String type = "c";
        System.out.print(this.type + super.type);
    }
}

Run | Debug
public static void main(String[] args) {
    new Cougar().go();
}
}

```

Feline cougar c f
 Feline cougar c c
 Feline cougar f f

No compila

¿Que acción realiza el archivo de dependencia pom.xml?*

1/1

Revisa que versiones de dependencias se tienen con otros proyectos
 Elimina las dependencias con otros proyectos

Recupera todas las dependencias con otros proyectos

Modifica las dependencias que se tienen con otros proyectos

¿Que causa no corresponde al antipatron Contenedor Magico?*

1/1

Reparto de responsabilidades incorrecto

Falta de cumplimiento de la arquitectura

Intervención demasiado limitada

Desacoplamiento entre librerías

Selecciona las tres respuestas que describen las mejores practicas establecidas para acceder a Datos*

0/1

En el caso de procesos Batch no deben existir cesiones de ficheros entre UUAA's

Una UUAA puede acceder a datos de terceros a través de las librerías APX que el dueño del dato proporcione

No se debe manejar el objeto Datasource directamente

Una UUAA solo puede acceder a los datos que ella misma custodia

Respuesta correcta

Una UUAA puede acceder a datos de terceros a través de las librerías APX que el dueño del dato proporcione

No se debe manejar el objeto Datasource directamente

Una UUAA solo puede acceder a los datos que ella misma custodia

Las partes de un campo variable definidos por la cadena NMM son:*

1/1

N es el orden de la variable literal (1 o 4) , MM la longitud maxima del campo informado en la variable (15 es el maximo)

N es el orden de la variable literal (1 y 4) , MM la longitud maxima del campo informado en la variable (15 es el maximo)

N es el orden de la variable literal (1 o 2) , MM la longitud maxima del campo informado en la variable (15 es el maximo)

N es el orden de la variable literal (1 o 2) , MM la longitud maxima del campo informado en la variable (20 es el maximo)

El objeto _____ no debe manejarse directamente. No debe crear instancias , abrir o cerrar explicitamente conexiones a la base de datos*

1/1

Datasource

JPA
JDBC
Database

En un Batch, ¿Cual es el step donde el desarrollador introducirá la logica empresarial necesaria?*

0/1

itemWriter

Task
Chunk
JobLauncher

Respuesta correcta

Task

El identificador unico para la ejecución de transacciones/Job se genera automaticamente como ultimo paso cuando se invoca las transacciones en linea/Batch. El resultado de la transaccion se puede propagar para la correlación en las otras capas.*

1/1

Falso
Verdadero

Un _____ proporciona una solución a un problema de diseño. Debe cumplir con diferentes características, como la efectividad al resolver problemas similares en ocasiones anteriores. Por lo tanto, debe de ser re-utilizable, es decir, aplicable a diferentes problemas en diferentes circunstancias.

*

1/1

Reutilizando código
Anti-patrón
Metodología
Patrón de diseño

¿Cuál es el resultado?

*

1/1

```
import java.util.*;  
public class App {  
    public static void main(String[] args) {  
        List p = new ArrayList();  
        p.add(7);  
        p.add(1);  
        p.add(5);  
        p.add(1);  
        p.remove(1);  
        System.out.println(p);  
    }  
}
```

[7, 5]

[7, 1]

[7, 5, 1]

[7, 1, 5, 1]

Para enviar un aviso desde una biblioteca o desde la transacción, cualquiera que sea el tipo, el método utilizado es _____.

*

0/1

addSeverity
addError
addAdvice
addWarning

Respuesta correcta

addAdvice

Cualquier documento utilizado (.dll, .txt, etc.) se almacenará en el sistema de archivos compartidos del nodo Batch (generalmente _____) y luego se eliminará periódicamente.

*

0/1

/fichtemporal/datent.
/datentJobs/datent.
/fichtemcomp/datent.
/temporalFiles/datent.

Respuesta correcta

/fichtemcomp/datent.

Which five methods, inserted independently at line 5, will compile? (Choose five)

*

0/1

```
1 public class Blip{  
2     protected int blipvert(int x){ return 0  
3 }  
4 class Vert extends Blip{  
5     //insert code here  
6 }
```

Public int blipvert(int x) { return 0; }

Protected long blipvert(int x) { return 0; }

```
Protected int blipvert(long x) { return 0; }
```

```
Private int blipvert(long x) { return 0; }
```

```
Protected long blipvert(int x, int y) { return 0; }
```

```
Private int blipvert(int x) { return 0; }
```

```
Protected long blipvert(long x) { return 0; }
```

Respuesta correcta

```
Protected long blipvert(long x) { return 0; }
```

```
Public int blipvert(int x) { return 0; }
```

```
Private int blipvert(long x) { return 0; }
```

```
Protected long blipvert(int x, int y) { return 0; }
```

```
Protected int blipvert(long x) { return 0; }
```

Los componentes de seguridad lógica que dan servicio a la arquitectura no están acoplados a la Arquitectura APX.

*

1/1

Verdadero

Falso

La información utilizada en la lógica de negocio de la transacción debe incluirse en

_____.

*

1/1

Los archivos locales.

Las variables globales.

Los parámetros de entrada.

Las variables locales.

¿Cuáles son las acciones posteriores a la ejecución de la lógica de negocio en la capa de Arquitectura de control?

*

0/1

Validation of Exit Code, Commit Control, Electronic Journal, Business Event Notification, Monitorization

Validation of Exit Code, Commit Control, ROP Writing, Business Event Notification, Monitorization

Validation of Exit Code, Commit Control, ROP Writing, Operations Registry, Monitorization

Validation of Exit Code, Commit Control, ROP Writing, Business Event Notification, Unique Identifier.

Respuesta correcta

Validation of Exit Code, Commit Control, ROP Writing, Business Event Notification, Monitorization

*

0/1

Given:

```
1. class Super{
2.     private int a;
3.     protected Super(int a){ this.a = a; }
4. }
...
11. class Sub extends Super{
12.     public Sub(int a){ super(a);}
13.     public Sub(){ this.a = 5;}
14. }
```

Which two independently, will allow Sub to compile? (Choose two)

Change line 2 to: public int a;

Change line 13 to: public Sub(){ super(5);}

Change line 2 to: protected int a;

Change line 13 to: public Sub(){ this(5);}

Change line 13 to: public Sub(){ super(a);}

Respuesta correcta

Change line 13 to: public Sub(){ super(5);}

Change line 13 to: public Sub(){ this(5);}

Selecciona las únicas excepciones que pueden ser capturadas y administradas por las aplicaciones, típicas de APX:

*

1/1

com.bbva.apx.exception.business.NullPointerException
com.bbva.apx.exception.business.IOException
com.bbva.apx.exception.io.network.TimeoutException
com.bbva.apx.exception.business.BusinessException

*

1/1

Given

```
public static void main(String[] args){  
    int[][] array2D = {{0,1,2}, {3,4,5,6}};  
    System.out.print(array2D[0].length + "");  
    System.out.print(array2D[1].getClass().isArray() + "");  
    System.out.print(array2D[0][1]);  
}
```

What is the result?

3false3
3false1
2false1
3true1

2true3

Which two statments are true?

*

1/1

An interface CANNOT be extended by another interface.

An abstract class can be extended by a concrete class.

An abstract class CANNOT be extended by an abstract class.

An interface can be extended by an abstract class.

An abstract class can implement an interface.

An abstract class can be extended by an interface.

*

1/1

Given:

```
class Alpha{ String getType(){ return "alpha";}}
class Beta extends Alpha{String getType(){ return "beta";}}
public class Gamma extends Beta { String getType(){ return "gamma";}
    public static void main(String[] args) {
        Gamma g1 = (Gamma) new Alpha();
        Gamma g2 = (Gamma) new Beta();
        System.out.print(g1.getType()+ " " +g2.getType());
    }
}
```

What is the result?

Gamma gamma
Beta beta
Alpha beta
Compilation fails

Which three are bad practices?

*

0/1

Checking for an IOException and ensuring that the program can recover if one occurs.

Checking for ArrayIndexOutOfBoundsException and ensuring that the program can recover if one occurs.

Checking for FileNotFoundException to inform a user that a filename entered is not valid.

Checking for Error and, if necessary, restarting the program to ensure that users are unaware problems.

Checking for ArrayIndexOutOfBoundsException when iterating through an array to determine when all elements have been visited.

Respuesta correcta

Checking for ArrayIndexOutOfBoundsException and ensuring that the program can recover if one occurs.

Checking for Error and, if necessary, restarting the program to ensure that users are unaware problems.

Checking for ArrayIndexOutOfBoundsException when iterating through an array to determine when all elements have been visited.

La arquitectura batch publica un conector WebService que no admite solicitudes para la ejecución de procesos batch ejecutadas a través del portal de desarrollo. Las aplicaciones que requieran la ejecución de procesos batch bajo demanda, también solicitan la ejecución a través de esta interfaz WebService. Este último escenario únicamente se está produciendo en entornos BBVA CIB.

*

0/1

Verdadero

Falso

Respuesta correcta

Falso

¿Quién es el responsable de iniciar un Job?

*

1/1

JobExecution

JobLauncher

JobInstance

JobBegginer

¿Cuáles son los dos tipos de step que existen en Batch APX?

*

1/1

Creator, Processor

Processor, Task
Chunk, Reader
Chunk, Task

Las 3 principales partes de un task

*

1/1

Chunk, processing, output
Input, processing, output

Input, load, processing, Output
Input, Load, Output

En batch , cada step tiene

*

1/1

itemInput, itemProcessor y itemWriter
itemReader, itemProcessor y itemWriter

itemReader, itemProcessor y item Output

¿Cuál es la ubicación de archivos temporales utilizados en Batch?

*

1/1

/fichtemporal/datent
/fichtemcomp/datent

/temporalFiles/datent
/datentJobs/datent

¿Qué característica NO pertenece al BATCH?

*

1/1

Procesamiento generalmente grande volúmenes de información
Interacción con usuario (directo)

Procesamiento normal en una ventana de lote fuera de línea
Requisitos no funcionales complejos

Describe las Clases que implementan la lógica de cada uno de los steps en un proceso

Batch

*

1/1

OPC, Processors, Readers
Readers, Writers, Processors

Processors, UID, OPC

De la lista siguiente, indique cuales son las invocaciones restringidas en la Arquitectura Batch:

*

0/1

Invocación WebServices, servicios REST
Acceso a recursos externos vía HTTP
Uso de Pre y Post Acciones
Escritura en Registro de Operaciones
Escritura en el Diario Electrónico.

a, b y c

a, c y d

Todas las anteriores

Respuesta correcta

Invocación WebServices, servicios REST
Acceso a recursos externos vía HTTP
Uso de Pre y Post Acciones

¿Cuál es el prefijo que debe utilizarse en la nomenclatura de los métodos que exponen las librerías para ser consumidos?

*

0/1

execute[Nombre_elegido_por_app] donde [Nombre_elegido_por_app] es opcional
dependency[[Nombre_elegido_por_app] donde [Nombre_elegido_por_app] es opcional

execute[Nombre_elegido_por_app] donde [Nombre_elegido_por_app] es obligatorio

dependency[[Nombre_elegido_por_app] donde [Nombre_elegido_por_app] es obligatorio

Respuesta correcta

execute[Nombre_elegido_por_app] donde [Nombre_elegido_por_app] es opcional

Estos tipos de librerías se pueden combinar entre sí para tener una librería con diferentes capacidades o no elegir ninguno de las opciones anteriores.

★

1/1

Online y Batch

Shell y JDBC

Encapsula la lógica empresarial y el acceso a los datos. Publica su interfaz para que pueda ser utilizada por terceros.

★

1/1

DTO

Transacción APX

Unidad funcional

Librería APX

Capa de Acceso a Datos

Para garantizar el comportamiento transaccional de APX, la arquitectura se basa en el estándar ____, por lo que los recursos administrados dentro de la transacción deben ser del tipo XA.

★

1/1

JT3

JTX

JPA

JTA

El objeto _____ no debe manejarse directamente. No debe crear instancias, abrir o cerrar explícitamente conexiones a la base de datos.

*

1/1

Database
Conection
Datasource

JDBC

En las consultas a BBDD debe usar variables _____ para no afectar el rendimiento de la base de datos

*

1/1

Miembro
Bind

Locales
Dynamic

Indica a JUnit que la propiedad que usa esta anotación es una simulación y, por lo tanto, se inicializa como tal y es susceptible de ser inyectada por @InjectMocks.

*

1/1

Mockito
Mock

Inject
InjectMock

¿Cuál de las siguientes opciones son las consideraciones para tener en cuenta en la instalación en Entorno de Ejecución Local?

*

0/1

Descomprimir la carpeta arquitectura en la ruta raíz /, es probable que se requieran permisos de administrador (sudo).

Dar permisos de ejecución a la carpeta y archivos del ENTORNO_LOCAL_APX: sudo chmod -R 777 /ENTORNO_LOCAL_APX

La instalación debe realizarse sobre un ordenador de sobremesa del BBVA con sistema operativo Windows o Linux.

Las rutas y urls proporcionadas serán exclusivamente para un entorno local

Respuesta correcta

La instalación debe realizarse sobre un ordenador de sobremesa del BBVA con sistema operativo Windows o Linux.

Las rutas y urls proporcionadas serán exclusivamente para un entorno local

En un Batch cual es el step donde el desarrollador introdujera la logica empresarial necesaria

*

1/1

Chunk
JobLauncher
ItemWriter
Task

¿Cuál definición de APX es incorrecta?

0/1

Extended Platform Architecture
Extended Programming Architecture
Backend Extended Platform
Extended Java Backend Architecture

Respuesta correcta

Extended Platform Architecture

¿Qué es APX Online?

*

1/1

Basada en Java, diseñada y construida por BBVA con ámbito local para ser utilizada en el Banco.

Esta plataforma contiene capacidades Front y está dirigida a aplicaciones que necesiten crear lógica de negocio sin acoplamiento con la presentación. Es un Sistema de alto rendimiento, con fácil escalabilidad y disponibilidad garantizada

Ejecución en entornos normales tipo R2 es la misma que la arquitectura diseñada y planteada para entornos Cloud. Las versiones que se generan son no duales por lo que cambia el empaquetado de la Arquitectura y la estrategia de aprovisionamiento de componentes aplicativos.

Habilita los servicios bancarios transaccionales fuera del entorno de mainframe, pero brinda las mismas capacidades en el mundo distribuido, reduce los costos generales del sistema, se basa en tecnologías de código abierto, permite a los desarrolladores aprovechar todos los servicios extendidos provistos en la plataforma para construir servicios lo más rápido posible

La Plataforma Extendida BackEnd Java APX nació con el requerimiento que pudiera funcionar en colaboración con el MainFrame o stand-alone (independientemente).

Igualmente se pueden hacer uso de sus capacidades de Arquitectura Bancaria o no utilizarlas cuando las aplicaciones se salgan de este modelo

*

1/1

Verdadero

Falso

Dentro de los casos de uso de la Arquitectura de APX y con el fin de estandarizar las aplicaciones del banco se plantearon dos paradigmas compatibles con BBVA:

1. Backend Banca Tradicional

2. Servicios Distribuidos Backend

*

1/1

Verdadero

Falso

Las utilidades APX implementadas tanto para la arquitectura Online como para la arquitectura Batch son:

*

0/1

Communication Manager: Se crea una librería con la capacidad de invocar a G.U.C para el envío de notificaciones. Interbackend proxy: Se crea una librería con la capacidad de invocar a los conectores IMS (para acceso a Host).

Compresión/Descompresión: Se crea una librería con la capacidad de comprimir y descomprimir archivos en zip. Generador de documentos: Se crea una librería con la capacidad de generar documentos.

Respuesta correcta

Compresión/Descompresión: Se crea una librería con la capacidad de comprimir y descomprimir archivos en zip. Generador de documentos: Se crea una librería con la capacidad de generar documentos.

Las utilidades APX implementadas únicamente para la arquitectura Online son:

*

0/1

Communication Manager: Se crea una librería con la capacidad de invocar a G.U.C para el envío de notificaciones. Interbackend proxy: Se crea una librería con la capacidad de invocar a los conectores IMS (para acceso a Host).

Compresión/Descompresión: Se crea una librería con la capacidad de comprimir y descomprimir archivos en zip. Generador de documentos: Se crea una librería con la capacidad de generar documentos.

Respuesta correcta

Communication Manager: Se crea una librería con la capacidad de invocar a G.U.C para el envío de notificaciones. Interbackend proxy: Se crea una librería con la capacidad de invocar a los conectores IMS (para acceso a Host).

La descripción de la Capa de Control es: Componente recibe la información necesaria para hacer el dispatch de la ejecución de transacciones. Se crea a la vez que la sesión de contexto transaccional, para permitir que la arquitectura pueda ejecutar la lógica de Negocio. Proporciona el ámbito transaccional por lo que es capaz de confirmar (commit) o deshacer (roll-back) ejecuciones. Este componente es el que gestiona la definición de transacción de acuerdo con la petición y ejecuta las pre y post acciones relacionadas con la lógica de negocio.

*

1/1

Verdadero

Falso

Estos módulos encapsulan toda la funcionalidad relacionada con las operaciones bancarias como el diario o registro de operaciones. Hay dos puntos claves en relación con estos servicios:

1. No deben afectar al rendimiento ejecución de la lógica de negocio por lo que el diario y el registro de las operaciones han de usar colas internas para desacoplar su propia ejecución de la ejecución de la lógica de negocio

2. Estos módulos son responsables de datos que pueden ser almacenados en el diario del MainFrame o en el propio de la arquitectura. Por lo tanto, el servicio de diario tiene que ser capaz de almacenar los datos del diario en ambos sitios.

*

1/1

Verdadero

Falso

De la lista siguiente marca 2 funcionalidades de servicios persistencia incluidas en APX

*

1/1

SFTP

PS9

JDBC

JMS/RABBIT

Two-Phase Commit

JNDI

De la lista siguiente marca 2 funcionalidades de servicios conectores incluidos en APX

*

0/1

Gestor de Eventos

HTTP/HTTPs

JDBC

JMS/RABBIT

Two-Phase Commit

JNDI

Respuesta correcta

Gestor de Eventos

HTTP/HTTPs

De la lista siguiente marca 2 funcionalidades de servicios Core incluidos en APX

*

0/1

Soporte transaccional

Ejecución de reglas de negocio

Casos restringidos

Thread Affinity

Soporte Transaccional JTA/XA

Autorización y Perfilado

Respuesta correcta

Thread Affinity

Soporte Transaccional JTA/XA

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

*

0/1

Pila de ejecución

Solicitud

Transacción
Ejecución

Respuesta correcta

Transacción

¿Cuáles son los campos que definen a una transacción en APX?

*

0/1

Aplicación, versión, país, paquete descripción
Aplicación, dominio, país, paquete descripción
Aplicación, transacción, versión, país, descripción
Aplicación, transacción, dominio, versión, país, descripción

Respuesta correcta

Aplicación, transacción, versión, país, descripción

Son dos de los protocolos físicos que APX admite en una solicitud.

*

0/1

JMS y REST
MQ y TCP
QP05 y MQ
AQPG y PS9

Respuesta correcta

JMS y REST

¿Cuáles son los protocolos físicos disponibles en APX Online?

Selecciona 2

*

1/1

HTTP/ SOAP

HTTP/ REST / RESTFul (A través de Servicios Backend)

AQPG (Formato Arquitectura Plataforma QP05: síncrono o asíncrono)

PS9

¿Cuáles son los protocolos lógicos disponibles en APX?

Selecciona 2

*

0/1

HTTP/ SOAP

PS9

HTTP/ REST / RESTFul (A través de Servicios Backend)

AQPG (Formato Arquitectura Plataforma QP05: síncrono o asíncrono)

Respuesta correcta

PS9

AQPG (Formato Arquitectura Plataforma QP05: síncrono o asíncrono)

La invocación síncrona no está permitida. En caso de que una transacción deba invocar a otra, solo se puede hacer de forma asíncrona

*

1/1

Verdadero

Falso

Selecciona al menos dos restricciones que se tienen en APX en la invocación entre transacciones.

*

0/1

No se permite la invocación síncrona entre transacciones.

No se permite la invocación asíncrona entre transacciones

La invocación asíncrona entre transacciones es en casos de transacciones críticas

La invocación asíncrona entre transacciones es casos de transacciones NO críticas

Desde una transacción asíncrona no se puede invocar nuevamente a otra transacción asíncrona

Desde una transacción asíncrona se puede invocar nuevamente a otra transacción asíncrona

Respuesta correcta

No se permite la invocación síncrona entre transacciones.

La invocación asíncrona entre transacciones es casos de transacciones NO críticas

Desde una transacción asíncrona no se puede invocar nuevamente a otra transacción asíncrona

Un _____ proporciona una solución a un problema de diseño. Debe cumplir con diferentes características, como la efectividad al resolver problemas similares en ocasiones anteriores. Por lo tanto, debe ser reutilizable, es decir, aplicable a diferentes problemas en diferentes circunstancias

*

1/1

Metodología

Patrón de Diseño

Anti patrón

Reutilización de código

Un anti patrón es un diseño que conduce a una mala solución de un problema. Como lo son el Blob y el contenedor mágico en librerías.

*

1/1

Verdadero

Falso

Se utilizará este patrón siempre que se requiera dividir un conjunto de datos en varios subconjuntos y adicionalmente se necesite acceder a ellas de forma aislada. El uso de este patrón es conveniente siempre para mejorar la eficiencia de consultas a orígenes de datos variados, obteniendo únicamente lo que se necesita.

*

1/1

Blob
Contenedor magico
Paginacion de librerias

CRUD

Este tipo de paginación es el que establece Arquitectura y Calidad como el más eficiente, tanto en consumo de recursos como en el tiempo de proceso para la consecución de la información, lo que redunda en un menor tiempo de espera por parte del usuario final.

*

1/1

Paginación en Librería

Paginación en Transaccion
Paginación en Batch
Paginación en DTO

Este patrón suele representarse como una única librería que contiene la mayoría de la funcionalidad y que otras librerías utilizan incorrectamente de apoyo. La clave del problema es que puede contener un porcentaje muy elevado de responsabilidades dentro de la solución a diseñar

*

0/1

Contenedor Magico

Blob
Lava Flow
CRUD

Respuesta correcta

Blob

¿Cómo solucionamos un Blob?

*

1/1

Mover la funcionalidad a la transacción reduciendo la carga en la librería.

Mover el comportamiento a sus librerías relacionadas, reduciendo el acoplamiento entre librerías y simplificando los mantenimientos.

Moviendo la lógica a diferentes métodos y clases dentro de la misma librería
Mover el comportamiento a DTOs relacionados, reduciendo el acoplamiento entre los componentes y simplificando los mantenimientos

¿Cuáles son ejemplos de Patrones de diseño para APX?

*

1/1

Paginación en transacciones, paginación en librerías, paginación en componentes APX
Paginación en librerías, CRUD en librerías, DTOs en componentes APX

Paginación en transacciones, CRUD en librerías, CRUD en componentes APX, DTOs en componentes APX
Paginación en componentes APX. CRUD en librerías, DTOs en componentes APX

¿Qué patrones de la siguiente lista son obligatorios del Data Transfer Object (DTO) en APX?

*

0/1

Se necesite una agrupación de datos simples en una clase con cierta funcionalidad asociada
Se requiera intercambiar información entre diferentes componentes en APX de forma coherente, organizada y agrupada.

Implementar un Servicio Backend
Todas las anteriores
Ninguna de las anteriores

Respuesta correcta
Todas las anteriores

Selecciona dos aplicaciones que justifiquen el uso del patrón Data Transfer Object (DTO) es obligatorio cuando:

*

1/1

Se necesitan mapear demasiadas variables
Se necesita una agrupación simple de datos basada en una funcionalidad asociada

Desea intercambiar información entre diferentes componentes en APX de manera coherente, organizada y agrupada

Se hace conexión a Bases de datos relacionales

¿Cuál es la restricción al generar una jerarquía de clases DTO?

*

1/1

Que las librerías solo tengan parámetros de entrada

Que las clases no tengan dependencias circulares entre ellas.

Que las agrupaciones no dependan entre ellas mismas

Que los patrones sean simples

Los patrones de diseño sirven de apoyo para la búsqueda de soluciones a problemas comunes a la hora del desarrollo de software. De la siguiente lista, ¿cuál de los siguientes es un patrón de diseño?

*

1/1

Paginación en transacciones

Paginación en librerías

CRUD en librerías

DTOs en componentes APX

A, B y C

Todas las anteriores

Ninguna de las anteriores

¿Qué son los Patrones de diseño en APX?

*

1/1

Diseño que sirve de apoyo para la búsqueda de soluciones a problemas comunes a la hora del desarrollo de software.

Patrón de diseño que invariablemente conduce a una mala solución para un problema

Diseño de software que tiene como objetivo el desarrollo único de código, es decir, debe ser aplicable únicamente a una solución específica.

¿Cuál es el objetivo de trabajar con Patrones?

*

1/1

Los patrones de diseño sirven de apoyo para la búsqueda de soluciones a problemas comunes a la hora del desarrollo de software.

Brindar una solución a un problema de diseño. Para que una solución se pueda considerar un patrón debe cumplir con diferentes características, como la efectividad a la hora de resolver problemas similares en ocasiones anteriores. Por ende, debe ser reutilizable, es decir, aplicable a diferentes problemas en distintas circunstancias.

Estos componentes contienen información necesaria para la ejecución de la Transacción en APX y cualquier modificación al respecto podría alterar su comportamiento

a y b

a y c

Este patrón suele representarse en una librería que contiene un solo método “execute” con parámetros que actúan como filtros y que engloba diversas funcionalidades dispares en función de sus parámetros de entrada

*

1/1

Contenedor Mágico

CRUD

Paginacion

Blob

¿Qué causas no corresponden al patrón Contenedor Mágico?

*

1/1

Reparto de responsabilidades incorrecto

Falta de cumplimiento de la arquitectura

Intervención demasiado limitada

Desacoplamiento entre librerías

La aplicación de este anti patrón de diseño en los desarrollos, causará la búsqueda y análisis de una mejor solución a un problema por medio de reingeniería

*

1/1

Paginación en transacciones
Contenedor mágico

Paginación Mágica
Generador de Logs

¿Cuáles de las siguientes oraciones forman parte de las mejores prácticas establecidas para acceder a Datos?

*

0/1

Una UUA puede acceder a datos de terceros a través de librerías APX que el dueño del dato proporcione.

En el caso de procesos batch no deben existir cesiones de ficheros entre UUAs.

En el caso de bases de datos relacionales, deberá utilizarse JPA.

No se debe manejar el objeto Datasource directamente.

No se deben instanciar, ni abrir, ni cerrar explícitamente conexiones a Base de datos

Una UUA solo puede acceder a los datos que ella misma custodia

Respuesta correcta

Una UUA puede acceder a datos de terceros a través de librerías APX que el dueño del dato proporcione.

No se debe manejar el objeto Datasource directamente.

No se deben instanciar, ni abrir, ni cerrar explícitamente conexiones a Base de datos

Una UUA solo puede acceder a los datos que ella misma custodia

¿Qué nivel de profundidad es permitido al realizar invocaciones de librería a librería?

*

1/1

Profundidades a 3 niveles: Transacción > Librería > Librería > Librería

Profundidades a 5 niveles: Transacción > Librería > Librería > Librería> Librería > Librería

Profundidades a 7 niveles: Transacción > Librería > Librería > Librería> Librería > Librería> Librería > Librería

Profundidades a 9 niveles: Transacción > Librería > Librería > Librería> Librería > Librería> Librería > Librería> Librería > Librería

Se debe evitar en una transacción o biblioteca, dentro de su lógica, invocar más de _____ librerías a la vez

*

0/1

7

15

5

9

12

Respuesta correcta

9

¿Cuál es el número máximo de librerías a invocar desde una transacción o librería?

*

0/1

7 librerías

3 librerías

9 librerías

12 librerías

Respuesta correcta

9 librerías

No puede incluir _____ o etiquetas similares. Estas anotaciones no reciben errores ni advertencias en el compilador en el momento de la compilación.

*

1/1

@SuppressWarnings

@BlockWarnings

@LeaveWarnings

@ForbbidenWarnings

La creación de hilos o su gestión por las aplicaciones está permitida.

*

1/1

Verdadero

Falso

Está prohibido poner las siguientes anotaciones de primavera en las clases de implementación de las bibliotecas APX: @Component, @Service, @Controller y @Repository, porque estas anotaciones causan conflictos en la resolución de classpath del paquete de la biblioteca y tienen el efecto de que la biblioteca no expone su servicio OSGI.

*

1/1

@Component, @Service, @Controller & @Bean
@Component, @Service @Controller & @Repository

@Component, @Override, @Controller & @Repository
@Component, @Override, @Controller & @Configuration

Está prohibido usar la propiedad "resolution: = 'optional' " en la definición de los paquetes en el paquete de importación, ya que puede causar errores en la implementación de la biblioteca y provocar fallas en otras bibliotecas que dependen de ella.

*

1/1

Verdadero

Falso

Las únicas excepciones que las aplicaciones pueden capturar y administrar son las siguientes, típicas de APX:

*

0/1

com.bbva.apx.exception.business.BusinessException
com.bbva.apx.exception.business.NullPointerException

com.bbva.apx.exception.business.IOException
com.bbva.apx.exception.io.network.TimeoutException

Respuesta correcta

com.bbva.apx.exception.business.BusinessException
com.bbva.apx.exception.io.network.TimeoutException

¿Que excepción no está permitida en APX?

*

1/1

com.bbva.apx.exception.business.BusinessException
com.bbva.apx.exception.io.network.TimeoutException
com.bbva.apx.exception.db.NullPointerException

com.bbva.apx.exception.db.IncorrectResultSizeException
com.bbva.apx.exception.db.NoResultException
com.bbva.apx.exception.db.TimeoutException

La gestión de errores debe seguir el modelo definido por la Arquitectura. En caso de querer mandar un aviso desde una librería o transacción, sea del tipo que sea, se usará el método_____.

*

0/1

addAdvice
addError
addSeverity
addWarning

Respuesta correcta

addAdvice

Método para fijar la severidad de un error.

*

1/1

setSeverity

setLevel

setAdvice

setError

¿Qué niveles de Logs existen dentro de APX?

*

0/1

INFO, WARNING & ERROR.

TRACE, DEBUG, INFO, WARNING & ERROR.

DEBUG, TRACE WARNING & ERROR

DEBUG, INFO, WARNING & ERROR

Respuesta correcta

DEBUG, INFO, WARNING & ERROR

¿Los niveles más comunes son DEBUG, INFO, WARNING y ERROR?

*

1/1

Verdadero

Falso

Para información de muy bajo nivel, solo útil para depurar la aplicación tanto en el desarrollo como en el análisis de incidentes ¿A qué tipo de nivel de Log se refiere?

*

0/1

Trace

Info

Debug

Low-Level

Respuesta correcta

Debug

Este nivel de Log tiene las siguientes características:

- Paradas e inicios de servicios y sistemas.
- Parámetros críticos o relevantes de configuración.
- Inicio y fin de transacciones y operaciones completas.
- Cambios del estado de las operaciones

★

1/1

Warning

Error

Info

Trace

DEBUG, sirve para información de muy bajo nivel solo útil para el debug de la aplicación, tanto en el desarrollo como en el análisis de incidencias

- Paradas y arranques de servicios y sistemas
- Parámetros críticos o relevantes de configuración
- Comienzo y fin de transacciones y operaciones completas
- Cambios de estado de operaciones

★

0/1

Verdadero

Falso

Respuesta correcta

Falso

WARN, son situaciones que impiden la ejecución correcta de una operación o transacción, pero sin afectar a otras operaciones o transacciones (error aislado o contenido)

★

1/1

Verdadero

Falso

ERROR, información de situaciones, que aún sin ser de error, si son anómalas o no previstas, aunque el aplicativo tiene alternativas para solventarlas

*

1/1

Verdadero

Falso

INFO, información de nivel superior que permita hacer un seguimiento de la ejecución normal

1/1

Verdadero

Falso

La arquitectura APX maneja mensajes de errores de tipo fijos y variables

*

1/1

Verdadero

Falso

Las partes de un campo variable definidos por la cadena _\$NMM\$ son:

*

0/1

(N es el orden de la variable literal (1 o 2), MM la longitud máxima del campo informado en la variable (20 es el máximo)

(N es el orden de la variable literal (1 a 4), MM la longitud máxima del campo informado en la variable (15 es el máximo)

(N es el orden de la variable literal (1 o 2), MM la longitud máxima del campo informado en la variable (15 es el máximo)

Respuesta correcta

(N es el orden de la variable literal (1 o 2), MM la longitud máxima del campo informado en la variable (15 es el máximo)

¿Qué características ofrece la Arquitectura Batch?

*

0/1

Soporte Multi-BBDD, usando fuentes de datos configurables

Acceso a conectores y utilidades APX: acceso DB MongoDB, Elasticsearch, Neo4J, motor de reglas, creador de documentos, etc.

Ejecución de múltiples transacciones

Invocación a Servicios Web

Respuesta correcta

Soporte Multi-BBDD, usando fuentes de datos configurables

Acceso a conectores y utilidades APX: acceso DB MongoDB, Elasticsearch, Neo4J, motor de reglas, creador de documentos, etc.

_____ fue creada para brindar una herramienta a los desarrolladores y administradores del entorno para gestionar las operaciones en la plataforma*

1/1

Atenea

Monitor APX

Consola de Operaciones APX

Kibana

Una de las acciones que realiza la Arquitectura APX previo a la ejecución de la lógica de negocio es: *

0/1

Validar errores

Escribir ROP

Escribe Log de ejecución

Se envía un evento con datos comerciales

Respuesta correcta

Escribe Log de ejecución

Para mandar un aviso desde una librería o transacción, cualquiera que sea el tipo, el método utilizado es ____.*

0/1

addAdvice
addError
addSeverity
addWarning

Respuesta correcta

addAdvice

Es una mala práctica capturar excepciones genéricas y totalmente prohibido sus tratamiento*

1/1

Verdadero

Falso

¿Quién gestiona la ejecución del Commit o Rollback para garantizar la transaccionalidad?*

1/1

La Base de Datos
La Lógica Aplicativa
La Arquitectura APX

La Arquitectura OSGI

La comunicación de APX a otros servicios NO APX se realiza ÚNICAMENTE a través de la utilidad APX ____.*

1/1

Certificados
Conector APO genérico (APIConnector)

Conectores
JDBC

La descripción de una transacción debe tener un significado técnico*

0/1

Verdadero

Falso

Respuesta correcta

Falso

Es una contradicción al usar el patrón CRUD*

1/1

Mantener todas las operaciones sobre la entidad encapsuladas en la librería

Tener varias librerías parciales para una única entidad

Centralizar en una librería de todas las operaciones básicas sobre una entidad

Tener una implementación ligera, evitando extender la lógica más allá del objetivo de la operación

_____, información de nivel superior que permita hacer un seguimiento de la ejecución normal*

1/1

WARNING

INFO

ERROR

TRACE

Selecciona dos comunicaciones que deben ser validadas por el equipo de Arquitectura antes de.*

1/1

Comunicación de APX a servicios NO APX

Comunicación con Elastic Search

Comunicación con Kibana

Si la transacción APX debe invocar a Mainframe más de dos veces

Una librería que no se despliega correctamente impide que los componentes que pueden consumirla se instalen correctamente. Cuando mayor sea el número de módulos dependientes, mayor será la probabilidad de falla*

1/1

Verdadero

Falso

En cualquier caso se permite la creación de hilos o su gestión por parte de las aplicaciones*

1/1

Verdadero

Falso

¿Qué características NO pertenecen al Batch?*

1/1

Interacción (directa) con el usuario

Generalmente procesa grandes volúmenes de información

Requisitos complejos no funcionales

Trazabilidad de la información y ejecución

¿En una transacción cuál es la estructura que contiene los paquetes que definen las clases utilizadas para las pruebas unitarias de la transacción?*

1/1

scr\test\java

src\main\java

scr\main\resources

scr\test\resources

Está permitido que la aplicación modifique o informe una variable de entorno de la JVM*

1/1

Verdadero

Falso

En una librería, ¿Cuál es el archivo en donde se define la lógica del negocio?*

1/1

UUAAR000Impl

UUAAR001-app-osgi.xml y UUAAR001-app.xml

UUAAR001-arq-osgi.xml y UUAAR001-arc.xml

UUAAR000Abstract

Tomada de forma aislada, una aplicación Batch es una aplicación independiente que se ejecuta periódicamente para realizar una task/step (repitiendo n veces) y luego se cierra. Esta única tarea suele tener 3 partes.*

1/1

Chunk, processing, output

Input, processing, output

Input, processing, writer

Input, load, output

No está permitido generar _____ para escribir de Logs. Tampoco se permite la escritura por _____, debe ser la arquitectura quien los escriba.*

0/1

Archivos propios / sysout

Archivos de control / system

Archivos propios / sysin

Archivos de lectura / sysout

Respuesta correcta

Archivos propios / sysout

Los patrones de diseño apoyan la búsqueda de soluciones comunes en el desarrollo de software. Los patrones adoptados al ecosistema APX son Blob in libraries y Magic container in Libraries*

1/1

Verdadero
Falso

¿Qué se considera retro compatible?*

0/1

Modificar el código para que sea compatible
No conozco la respuesta
Agregar un nuevo método de ejecución a la biblioteca, Modificar la lógica
Versionar el código

Respuesta correcta

Agregar un nuevo método de ejecución a la biblioteca, Modificar la lógica

@Component, @Service, @Controller y @Repository, estas anotaciones ¿que causan en las clases de implementación de las bibliotecas APX?

*

1/1

Ocultar los bundles donde se usan
Tienen el efecto de que la biblioteca no expone su servicio OSGi.

Mejorar la visibilidad a los métodos a usar
Tira la arquitectura
No conozco la respuesta

Máximo de resultados por paginación

*

1/1

1000
1
10000
No conozco la respuesta

2000
5000

La severidad del aviso que ésta contextual a la transacción debe realizarse desde la misma mediante el método...

*

0/1

setSeverity
addSeverity

No conozco la respuesta
setAdvice
addAdvice

Respuesta correcta
setSeverity

Agrega un campo de entrada obligatorio. Agregue un campo de salida. Modifique el tipo de un campo de entrada o salida existente. Son características de un cambio...

*

1/1

Cambio en desarrollo
Versionado
Retrocompatibles
No retrocompatibles

No conozco la respuesta

¿Qué es Biblioteca APX?

*

1/1

Representación de una Entidad Comercial en forma de Bean.
Es un desarrollo que reemplaza a Mainframe
Una transacción es la unidad de aplicación que se ejecutará en APX Online. Define la Unidad Transaccional a nivel de lógica de aplicación.
Encapsula la lógica empresarial y el acceso a los datos. Publica su interfaz para que pueda ser utilizada por terceros.

Es un software diseñado para crear proyectos tipo APX
No conozco la respuesta

Consideraciones para tener en cuenta para la gestión de errores

*

0/1

No conozco la respuesta

APX no permite la gestión de errores

El manejo de errores se lleva a cabo a través de la consola de operación APX actual

Permite lanzar errores en un bloque try ... catch

Cada UUAA tiene un rango de código de error asignado.

No hay consideraciones

Respuesta correcta

Cada UUAA tiene un rango de código de error asignado.

El manejo de errores se lleva a cabo a través de la consola de operación APX actual

Es obligatorio que los DTOs se implementen dentro de un....

*

1/1

empaquetado independiente (Bundle TX)

No conozco la respuesta

empaquetado independiente (Bundle LIB)

empaquetado independiente (Bundle DTO)

empaquetado independiente (Bundle Batch)

El acceso a los datos debe realizarse utilizando las utilidades que Arquitectura proporciona desde el asistente de biblioteca. En el caso de bases de datos relacionales, ¿Qué utilidad se debe usar?

*

1/1

Datio

APIConector

No conozco la respuesta

JDBC

Elastic

Las dependencias circulares son permitidas...

*

1/1

No conozco la respuesta

Están absolutamente prohibidas

Son permitidas

Algunas son permitidas

Depende de la necesidad del proyecto

Los hilos son permitidos...

*

1/1

Si

Depende del JVM que se ejecute

No

Depende donde se usará

No conozco la respuesta

El multiproceso no debe usarse en APX Batch ¿Que se puede hacer como opción?

*

1/1

Varios procesos por lotes al mismo tiempo

Varios procesos por lotes en diferentes tiempos

No conozco la respuesta

No es parte de Spring

No ejecutar procesos

Otro:

¿Qué es Batch?

*

1/1

No conozco la respuesta

Proceso que varía con respecto a sus dependencias

Proceso idéntico a online

Proceso que reemplaza al mainframe

Es un patrón de ejecución, ejecución de un programa sin el control o supervisión directa del usuario

Otro:

Excepción de Spring permitida

*

0/1

No conozco la respuesta

com.bbva.elara.utility.interbackend.cics.exceptions.BusinessException

java.lang.NumberFormatException

com.bbva.apx.exception.db.NoResultException

org.springframework.web.client.RestClientException

Respuesta correcta

org.springframework.web.client.RestClientException

El acceso a datos se hace mediante ¿Qué componentes en online?

*

1/1

Bibliotecas

No conozco la respuesta

Jobs

DTOs

Transacciones

Diferencias entre Las invocaciones de APX Batch e invocaciones de APX Online

*

0/1

Permite la posibilidad de invocar un batch de APX directamente

Los encabezados del protocolo lógico a enviar

Ninguno

Horarios de ejecución

Permitirá ejecutar transacciones de cics solo por protocolo ps10.

No conozco la respuesta

Respuesta correcta

Los encabezados del protocolo lógico a enviar

¿Qué método se usa para agregar avisos desde una Librería o transacción?

*

1/1

addAvisosAPX

No conozco la respuesta

addAdvice

setSeverity

addSeverity

APX, ¿Qué es Escritura ROP?

*

1/1

Es un Protocolo JMS

Es un Protocolo REST

Un LOG interno utilizado con fines de Auditoría, Fraude o Informativo que indica que la transacción ha finalizado

No conozco la respuesta

Escritura en Archivos

Es un Protocolo HTTP

Batch Execution Banking basada en *Spring Batch* ¿qué Invocación son restringidas?

*

0/1

GUC

Utilice acciones previas y posteriores.

Servicios web de invocación, servicios de descanso.

No conozco la respuesta

Acceso a recursos externos a través de HTTP.

Utilidades inherentes a la Arquitectura APX.

Compresión / Descompresión.

Otro:

Respuesta correcta

Utilice acciones previas y posteriores.

Compresión / Descompresión.

Servicios web de invocación, servicios de descanso.

Acceso a recursos externos a través de HTTP.

Utilidades inherentes a la Arquitectura APX.

GUC

El árbol de dependencia en la invocación a las bibliotecas debe ser finito, evitando profundidades superiores a

*

1/1

2

9

No conozco la respuesta

1

3

¿Qué es el *LOGGER.info*?

*

1/1

No conozco la respuesta

Le permite imprimir datos informativos con detalles finos que son útiles para depurar la aplicación. Imprime mensajes de error críticos que impiden que la aplicación funcione correctamente. Estos son los registros habilitados en Producción

Imprime advertencias o fallas no críticas de la aplicación, que no impiden que la aplicación continúe su flujo

Imprime información con el máximo nivel de detalle.

Imprime información útil que le permite ver el progreso de la aplicación.

¿Qué es Conector CICS?

*

0/1

Permite la posibilidad de invocar una transacción de APX directamente

Permitirá ejecutar transacciones de cics solo por protocolo ps10.

Permite la posibilidad de invocar una transacción de mainframe directamente

Permite la posibilidad de invocar un servicio ASO directamente

No conozco la respuesta

Permite la posibilidad de invocar un batch de APX directamente

Respuesta correcta

Permitirá ejecutar transacciones de cics solo por protocolo ps10.

En tiempo de ejecución, las dependencias con otras bibliotecas se resuelven en función de los paquetes declarados en la sección...

*

1/1

Abstract

dependency

Test

No conozco la respuesta

import-package

JPA ¿es permitido usarlo en APX?

*

0/1

Si

Tal vez

No conozco la respuesta

No

Respuesta correcta

No

Agregue un campo de entrada opcional. Modifique un campo de entrada obligatorio y configúrelo como opcional. Modifique un campo de salida opcional y hágalo obligatorio. Lógica interna. son características de un cambio...

*

1/1

No conozco la respuesta

Cambio en desarrollo

Versionado

Retrocompatibles

No retrocompatibles

El código *try ... catch* (Exception exc) ¿Es permitido en APX?

*

1/1

Tal vez
No

No conozco la respuesta
Si

En un DTO se desea intercambiar información de forma

*

0/1

No conozco la respuesta

Coherente

Agrupada
Asociada
Organizada

Respuesta correcta

Agrupada
Organizada
Coherente

¿Está permitida invocación sincrónica?

*

1/1

Si
No se permite la invocación sincrónica

No se
Dependiendo el desarrollo
No conozco la respuesta

Ejemplos de patrones de diseño

*

1/1

No conozco la respuesta
DTO en componentes APX.
Todas

Paginación en bibliotecas.
Paginación en transacciones.

CRUD en bibliotecas.

La comunicación de APX a otros servicios NO APX se realizará ÚNICAMENTE a través de...

*

1/1

DTOs

APIConnector

Librerías

Depende de su consumo

No conozco la respuesta

Ejemplo de contenedor mágico

*

0/1

No conozco la respuesta

Un campo de entrada se devuelve a la salida

Un campo de entrada no debe usarse para que la transacción haga cosas funcionalmente diferentes

Todas

Contiene varios métodos en una sola librería

Contiene diferentes variables

Respuesta correcta

Un campo de entrada no debe usarse para que la transacción haga cosas funcionalmente diferentes

Para mongo ¿qué utilidad debe utilizarse?

*

0/1

JPA

JDBC

No conozco la respuesta

APIConnector

Datio

Respuesta correcta

Datio

Chunk está formado por 3 componentes

*

1/1

Task

Escritor

Lector

Procesador

Step

No conozco la respuesta

Otro:

En caso de que una transacción deba invocar a otra, solo se puede realizar de forma...

*

1/1

No conozco la respuesta

Dependiendo el desarrollo

Sincrónica

Con un conector

Asincrónica

La Arquitectura es la encargada de realizar la gestión de *commit* para garantizar la transaccionalidad

*

1/1

No

Si

No conozco la respuesta

Es posible desde las librerías

Es posible desde la consola de configuración

¿Qué es el *LOGGER.debug*?

*

1/1

Le permite imprimir datos informativos con detalles finos que son útiles para depurar la aplicación.

No conozco la respuesta

Imprime advertencias o fallas no críticas de la aplicación, que no impiden que la aplicación continúe su flujo

Imprime mensajes de error críticos que impiden que la aplicación funcione correctamente. Estos son los registros habilitados en Producción

Imprime información con el máximo nivel de detalle.

Imprime información útil que le permite ver el progreso de la aplicación.

¿Por qué no se recomiendan las transacciones asíncronas?

*

1/1

Su lentitud de ejecución

No hay implementación de transacciones asíncronas

OSGI no reconoce las transacciones como dependencias

Se pierde el control de la ejecución

No se permite en México

No conozco la respuesta

Batch consta de 3 partes ¿Cuáles son?

*

1/1

Salida: escribe o realiza alguna acción como resultado del procesamiento.

No conozco la respuesta

Procesamiento: se realiza algún tipo de lógica empresarial sobre estos datos de entrada.

Entrada: obtenga y lea la información de entrada al proceso.

Otro:

Ejecución de procesos Monostep. ¿Se puede usar?

*

1/1

Si

No se recomienda la ejecución de trabajos con un solo paso.

Depende el desarrollo

No conozco la respuesta

Tal vez
Otro:

Niveles más comunes de LOG en APX

*

1/1

DEBUG, INFO, WARNING y ERROR

No conozco la respuesta
DEBUG, INFO, WARNING
DEBUG, INFO
TRACE, DEBUG, INFO, WARNING y ERROR

¿Qué es APX CLI?

*

1/1

Encapsula la lógica empresarial y el acceso a los datos. Publica su interfaz para que pueda ser utilizada por terceros.

No conozco la respuesta

Representación de una Entidad Comercial en forma de Bean.

Una transacción es la unidad de aplicación que se ejecutará en APX Online. Define la Unidad Transaccional a nivel de lógica de aplicación.

Es un software diseñado para crear proyectos tipo APX, así como administrar dependencias con otras bibliotecas APX y administrar utilidades

Es un desarrollo que reemplaza a Mainframe

¿Qué es Identificador único (UID)?

*

1/1

Identificador único de cada de cada componente en APX

Identificador único de cada Batch

Identificador único de cada transacción

Identificador único de cada DTO

No conozco la respuesta

Identifica de forma única una ejecución

¿Cuántas veces es posible invocar Mainframe?

*

1/1

Sin restricción

No es posible invocar Mainframe

No conozco la respuesta

2

1

¿Qué es el *LOGGER.warn*?

*

1/1

Imprime información útil que le permite ver el progreso de la aplicación.

Imprime información con el máximo nivel de detalle.

No conozco la respuesta

Imprime advertencias o fallas no críticas de la aplicación, que no impiden que la aplicación continúe su flujo

Le permite imprimir datos informativos con detalles finos que son útiles para depurar la aplicación.

Imprime mensajes de error críticos que impiden que la aplicación funcione correctamente. Estos son los registros habilitados en Producción

La creación de instancias, abrir o cerrar explícitamente conexiones a la base de datos, ¿de quién depende estas tareas?

*

1/1

Java

Bibliotecas

ASO

No conozco la respuesta

Arquitectura

Declarar una dependencia de compilación como opcional (*true*), ¿Qué efectos conlleva?

*

1/1

Hace que busque todas las dependencias definidas

Dependiendo el desarrollo

No conozco la respuesta

Hace que los paquetes definidos en la sección import-package también se establezcan como opcionales

No busca ninguna dependencia
Otro:

Únicas excepciones que pueden ser capturadas y administradas por las aplicaciones

*

0/1

No conozco la respuesta

com.bbva.apx.exception.io.network.TimeoutException

com.bbva.apx.exception.db.TimeoutException

com.bbva.apx.exception.db.IncorrectResultSizeException

com.bbva.apx.exception.db.DuplicateKeyException

com.bbva.apx.exception.business.BusinessException

com.bbva.apx.exception.db.NoResultException

com.bbva.elara.utility.interbackend.cics.exceptions.BusinessException

Respuesta correcta

com.bbva.apx.exception.business.BusinessException

com.bbva.apx.exception.io.network.TimeoutException

com.bbva.apx.exception.db.DuplicateKeyException

com.bbva.apx.exception.db.IncorrectResultSizeException

com.bbva.apx.exception.db.NoResultException

com.bbva.apx.exception.db.TimeoutException

com.bbva.elara.utility.interbackend.cics.exceptions.BusinessException

¿Qué es APX?

*

1/1

Es una extensión de Mainframe sustituyéndola

Es una forma de poder sustituir al Mainframe para ejecutar procesos más rápidos

Ofrece una alternativa confiable para el desarrollo de transacciones que no está acoplada al canal pero depende del Mainframe

No conozco la respuesta

La Arquitectura APX o Arquitectura Backend Java Extendida fue creada con el objetivo de ser una extensión de la Arquitectura Backend proporcionando las mismas capacidades en el mundo distribuido.

¿En qué ayuda un DTO?

*

1/1

Ayuda a agrupar métodos de un componente

Ayuda a identificar los métodos que se repiten en componentes APX

El uso de este patrón ayuda a identificar las agrupaciones de datos simples y/o complejos, y evita duplicidad de código.

No conozco la respuesta

Las bibliotecas pueden definir su propia ____ para determinar qué UUAAs externas pueden usar sus bibliotecas.

*

1/1

Autoconsumo

Visibilidad

De quien serán dependencias

Gestión de errores

De quien las consuma

No conozco la respuesta

¿Qué sucede cuando se implementa una nueva versión de una dependencia en una biblioteca?

*

0/1

No conozco la respuesta

No carga las dependencias correctas

No encuentra las dependencias

Errores en tiempo de ejecución

No tiene impacto en nuestra biblioteca en el momento de la ejecución

Respuesta correcta

No tiene impacto en nuestra biblioteca en el momento de la ejecución

Capacidades de APX

*

0/1

Interoperabilidad de backends

Implementaciones en caliente de acceso a datos

Integración con servicios de seguridad

Procesamiento transaccional

Monitoreo de procesamiento por lotes de servicios bancarios

No conozco la respuesta

Respuesta correcta

Implementaciones en caliente de acceso a datos

Procesamiento transaccional

Integración con servicios de seguridad

Monitoreo de procesamiento por lotes de servicios bancarios

Interoperabilidad de backends

¿Por qué se deben utilizar variables BIND?

*

1/1

Para no afectar el rendimiento de la base de datos.

Para una mejor vista

Para enviar muchos datos al mismo tiempo

Para mejorar las sentencias

No conozco la respuesta

¿Qué es patrón de diseño?

*

1/1

Patrón de identificación de procesos

Proporciona una solución a un problema de diseño

Identificador de diseños

Proporciona información de la ejecución

No conozco la respuesta

Patrón de identificación único de procesos

Las dependencias circulares que ocasionan...

*

1/1

No conozco la respuesta

Depende de la necesidad del proyecto
Emporar la comunicación de los Bundles
Que OSGI no cargue estas dependencias correctamente

Mejorar el acceso a los componentes

¿Por qué no se recomiendan las transacciones asíncronas?

*

1/1

No sabrá si ha concluido con un commit OK o con un OK retroceso.

Su lentitud de ejecución
OSGI no reconoce las transacciones como dependencias
No se permite en Mexico
No hay implementación de transacciones asíncronas
No conozco la respuesta

El multiproceso no debe usarse en APX Batch ¿por qué?

*

1/1

No es parte de Spring
Por la incompatibilidad en la JVM
Fuente de consumo de recursos descontrolado y bloqueos de otros trabajos que se ejecutan al mismo tiempo.

Es una etiqueta no permitida
No conozco la respuesta
Otro:

Batch, se pueden implementar las siguientes interfaces ¿cuáles?

*

1/1

ItemProcessor
ItemReader
ItemWriter
No conozco la respuesta

ItemReader, ItemProcessor, ItemWriter

Otro:

Excepciones de Java permitida

*

0/1

No conozco la respuesta

`org.springframework.web.client.RestClientException`

`com.bbva.apx.exception.db.NoResultException`

`java.lang.NumberFormatException`

`com.bbva.elara.utility.interbackend.cics.exceptions.BusinessException`

Respuesta correcta

`java.lang.NumberFormatException`

Batch Execution Banking basada en Spring Batch ¿características que proporciona?

*

0/1

Soporte multi-DB usando fuentes de datos configurables

Lectura / Escritura de múltiples formatos de archivo (TXT, CSV, etc.).

No permite la ejecución de ciertos comandos de eliminación del sistema.

Acceso a utilidades y recursos del sistema.

Lectores y escritores listos para usar y personalizables.

Integración con planificadores: BMC Control-M, Tivoli OPC, Quartz.

No conozco la respuesta

Opción de compromisos parciales.

El proceso por lotes se reinicia en caso de error.

Los lectores acceden a las bases de datos en modo "solo lectura".

Control del segmento de ejecución.

Otro:

Respuesta correcta

Lectores y escritores listos para usar y personalizables.

Acceso a utilidades y recursos del sistema.

Control del segmento de ejecución.

No permite la ejecución de ciertos comandos de eliminación del sistema.

Opción de compromisos parciales.

Lectura / Escritura de múltiples formatos de archivo (TXT, CSV, etc.).

Integración con planificadores: BMC Control-M, Tivoli OPC, Quartz.

Soporte multi-DB usando fuentes de datos configurables

El proceso por lotes se reinicia en caso de error.

Los lectores acceden a las bases de datos en modo "solo lectura".

Propósito Bibliotecas del motor de reglas

*

1/1

No conozco la respuesta

Encapsula la lógica empresarial y el acceso a los datos. Publica su interfaz para que pueda ser utilizada por terceros.

El propósito de esta biblioteca es permitir que el desarrollador invoque las reglas comerciales que necesita a través del motor de reglas Drools

Una transacción es la unidad de aplicación que se ejecutará en APX Online. Define la Unidad Transaccional a nivel de lógica de aplicación.

Es un software diseñado para crear proyectos tipo APX, así como administrar dependencias con otras bibliotecas APX y administrar utilidades

Es un desarrollo que reemplaza a Mainframe

Características del Batch

*

0/1

No conozco la respuesta

Generalmente procesa grandes volúmenes de información.

Requisitos complejos no funcionales

Procesamiento normal en una ventana por lotes durante fuera de línea.

Aplicación por lotes: estrategias de procesamiento

Sin interacción (directa) del usuario.

Trazabilidad de la información y ejecución.

Rendimiento / procesamiento de alta velocidad.

Otro:

Respuesta correcta

Procesamiento normal en una ventana por lotes durante fuera de línea.

Trazabilidad de la información y ejecución.

Requisitos complejos no funcionales

Rendimiento / procesamiento de alta velocidad.

Sin interacción (directa) del usuario.

Aplicación por lotes: estrategias de procesamiento

Generalmente procesa grandes volúmenes de información.

¿Qué es el *LOGGER.trace*?

*

1/1

Imprime información con el máximo nivel de detalle.

Imprime advertencias o fallas no críticas de la aplicación, que no impiden que la aplicación continúe su flujo

No conozco la respuesta

Le permite imprimir datos informativos con detalles finos que son útiles para depurar la aplicación.

Imprime información útil que le permite ver el progreso de la aplicación.

Imprime mensajes de error críticos que impiden que la aplicación funcione correctamente. Estos son los registros habilitados en Producción

¿Qué es Transacción APX?

*

1/1

Es un software diseñado para crear proyectos tipo APX

Representación de una Entidad Comercial en forma de Bean.

Una transacción es la unidad de aplicación que se ejecutará en APX Online. Define la Unidad Transaccional a nivel de lógica de aplicación.

No conozco la respuesta

Encapsula la lógica empresarial y el acceso a los datos. Publica su interfaz para que pueda ser utilizada por terceros.

Es un desarrollo que reemplaza a Mainframe

Una transacción o librería, dentro de su lógica, debe evitar invocar más de __ librerías

*

0/1

9

No conozco la respuesta

2

1

3

Respuesta correcta

9

Existe un API de código de barras

*

0/1

Si

No conozco la respuesta

No

Solo en España

Tal vez

Su uso es restringido

Respuesta correcta

Si

Como es la estandarización del código en APX

*

1/1

No conozco la respuesta

Desarrollo pase por Bitbucket, jenkins, sonar

Desarrollo se defina con respecto a las reglas de APX

Desarrollo definido en inglés, ya que los componentes se pueden reutilizar en todo el Grupo BBVA.

Desarrollo se defina con respecto a las reglas de Sonar

NO se deben utilizar sentencias que incluyan atributos o acciones específicas de un administrador de base de datos en particular, como lo es...

*

1/1

ROWNUM

OR

SELECT

No conozco la respuesta

FROM

Declarar una dependencia de compilación como opcional (true), ¿es correcto declararla?

*

0/1

No conozco la respuesta

Dependiendo el desarrollo

No

Si

Tal vez

Otro:

Respuesta correcta

No

¿Qué es Document Manager?

*

1/1

Utilidad que proporciona acceso a los servicios de gestión de documentos externos a las aplicaciones

Utilidad para organizar documentos

Utilidad para crear Documentos

No conozco la respuesta

Utilidad para manejar Documentos

Utilidad que está restringida para su uso

Restricciones para imprimir loggers en APX

*

0/1

Le permite imprimir datos informativos

No se permite la generación de archivos propios para la escritura de logs.

Tampoco está permitido escribir por sysout

No hay restricciones

Solo en España hay restricciones

No conozco la respuesta

Respuesta correcta

Tampoco está permitido escribir por sysout

No se permite la generación de archivos propios para la escritura de logs.

¿Qué es El método getRequestHeader?

*

1/1

Utilidad para organizar documentos en cabezales
Utilidad para crear encabezados para la ejecución de transacciones
No conozco la respuesta
Recuperará el valor de los parámetros del encabezado con el que se ejecutó la transacción

Utilidad que Accede a BBDD
Recuperará los encabezados de ASO

Para que sirve IMSConnect

*

1/1

Permite la posibilidad de invocar una transacción de APX directamente
No conozco la respuesta
Permite la posibilidad de invocar una transacción de mainframe directamente

Permite la posibilidad de invocar un batch de APX directamente
Permite la posibilidad de invocar un DTO de APX directamente
Permite la posibilidad de invocar un servicio ASO directamente

¿Para qué se usa el JPA?

*

0/1

Permite insertar, actualiza y consultar en BBDD

No conozco la respuesta
Permite realizar un almacenamiento en caché de tablas de BBDD
Permite manejar las utilidades de APX (Datio, JDBC,...)
Permite usar componentes APX con cualquier BBDD

Respuesta correcta

Permite realizar un almacenamiento en caché de tablas de BBDD

APX, ¿protocolos físicos que admite?

*

1/1

SOAP
No conozco la respuesta
REST
JMS
HTTP
HTTP, JMS, REST

¿Qué es un DTO?*

1/1

Agrupar métodos de un componente

Un objeto de transferencia de datos es, transporta datos entre procesos

Es un objeto de agrupación de componentes APX

No conozco la respuesta

Una UUA ¿puede acceder a los datos que posee y datos de terceros a través de bibliotecas APX?

*

1/1

No

Tal vez

No conozco la respuesta

Si

¿Qué es un Chunk?*

1/1

Realiza una operación dentro del paso (por ejemplo, invocando utilidades de arquitectura)

Conjuntos de datos de entrada se procesan en la misma transacción

Conjunto de procesos

Conjunto de readers

No conozco la respuesta

Otro:

Tipos de pasos en un proceso Batch ¿Cuáles son?

*

1/1

Write

Task

Process

Read

No conozco la respuesta

Chunk

Otro:

Las etiquetas @SuppressWarnings o etiquetas similares son permitidas...

*

1/1

Depende del JVM que se ejecute

No

Algunas

Si

No conozco la respuesta

¿Qué es el *LOGGER.error*?

*

1/1

Le permite imprimir datos informativos con detalles finos que son útiles para depurar la aplicación.

No conozco la respuesta

Imprime información útil que le permite ver el progreso de la aplicación.

Imprime información con el máximo nivel de detalle.

Imprime advertencias o fallas no críticas de la aplicación, que no impiden que la aplicación continúe su flujo

Imprime mensajes de error críticos que impiden que la aplicación funcione correctamente. Estos son los registros habilitados en producción

What is the result?*

1/1

```
public class Test {  
    public static void main(String[] args) {  
        int b = 4;  
        b--;  
        System.out.print(--b);  
        System.out.println(b);  
    }  
}
```

22

12

32

33

In Java the difference between throws and throw is:*

1/1

Throws throws an exception and throw indicates the type of exception that the method.

Throws is used in methods and throw in constructors.

Throws indicates the type of exception that the method does not handle and throw an exception.

What is the result?*

0/1

```

class Feline {
    public String type = "f ";
    public Feline() {
        System.out.print("feline ");
    }
}
public class Cougar extends Feline {
    public Cougar() {
        System.out.print("cougar ");
    }
    void go() {
        type = "c ";
        System.out.print(this.type + super.type);
    }
    public static void main(String[] args) {
        new Cougar().go();
    }
}

```

Cougar c f.
 Feline cougar c c.
 Feline cougar c f.

Compilation fails.

Respuesta correcta

Feline cougar c c.

Which statement, when inserted into line " // TODO code application logic here", is valid in compilation time change?*

1/1

```

public class SampleClass {
    public static void main(String[] args) {
        AnotherSampleClass asc = new AnotherSampleClass();
        SampleClass sc = new SampleClass();
        // TODO code application logic here
    }
}
class AnotherSampleClass extends SampleClass { }

```

```

asc = sc;
sc = asc;

```

```

asc = (Object) sc;
asc= sc.clone();

```

What is the result?*

1/1

```

public class Test {
    public static void main(String[] args) {
        int[][] array = { {0}, {0,1}, {0,2,4}, {0,3,6,9}, {0,4,8,12,16} };
        System.out.println(array[4][1]);
        System.out.println(array[1][4]);
    }
}

```

4 Null.

Null 4.

An IllegalArgumentException is thrown at run time.

4 An ArrayIndexOutOfBoundsException is thrown at run time.

What is the result?*

1/1

```

import java.util.*;
public class App {
    public static void main(String[] args) {
        List p = new ArrayList();
        p.add(7);
        p.add(1);
        p.add(5);
        p.add(1);
        p.remove(1);
        System.out.println(p);
    }
}

```

[7, 1, 5, 1]
[7, 5, 1]

[7, 5]
[7, 1]

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

0/1


```

13. public class Speak {
14.     public static void main(String[] args) {
15.         Speak speakIT = new Tell();
16.         Tell tellIt = new Tell();
17.         speakIT.tellItLikeltIs();
18.         (Truth) speakIT.tellItLikeltIs();
19.         ((Truth) speakIT).tellItLikeltIs();
20.         tellIt.tellItLikeltIs();
21.         (Truth) tellIt.tellItLikeltIs();
22.         ((Truth) tellIt).tellItLikeltIs();
23.     }
24. }

```

```

class Tell extends Speak implements Truth {
    @Override
    public void tellItLikeltIs() {
        System.out.println("Right on!");
    }
}

```

```

interface Truth {
    public void tellItLikeltIs();
}

```

Line 17

Line 18

Line 19

Line 20

Line 21

Line 22

Respuesta correcta

Line 19

Line 20

Line 22

What is the result?*

1/1

```

public class Bees {
    public static void main(String[] args) {
        try {
            new Bees().go();
        } catch (Exception e) {
            System.out.println("thrown to main");
        }
    }
    synchronized void go() throws InterruptedException {
        Thread t1 = new Thread();
        t1.start();
        System.out.print("1 ");
        t1.wait(5000);
        System.out.print("2 ");
    }
}

```

The program prints 1 then 2 after 5 seconds.

The program prints: 1 thrown to main.

The program prints: 1 2 thrown to main.

The program prints: 1 then t1 waits for its notification.

Which three are valid? (Choose three)*

0/1

```

class ClassA {}
class ClassB extends ClassA {}
class ClassC extends ClassA {}
And:
ClassA p0 = new ClassA();
ClassB p1 = new ClassB();
ClassC p2 = new ClassC();
ClassA p3 = new ClassB();
ClassA p4 = new ClassC();

```

```
p0 = p1;  
p1 = p2;  
p2 = p4;  
p2 = (ClassC)p1;
```

```
p1 = (ClassB)p3;  
p2 = (ClassC)p4;
```

Respuesta correcta

```
p0 = p1;  
p1 = (ClassB)p3;  
p2 = (ClassC)p4;
```

Which three options correctly describe the relationship between the classes?*

0/1

```
class Class1 { String v1; }  
class Class2 {  
    Class1 c1;  
    String v2;  
}  
class Class3 { Class2 c1; String v3; }
```

Class2 has-a v3.

Class1 has-a v2.

Class2 has-a v2.

Class3 has-a v1.

Class2 has-a Class3.

Class2 has-a Class1.

Respuesta correcta

```
Class2 has-a v2.  
Class3 has-a v1.  
Class2 has-a Class1.
```

Which three implementations are valid?*

0/1

```
interface SampleCloseable {
    public void close() throws java.io.IOException;
}
```

```
class Test implements SampleCloseable { public void close() throws java.io.IOException { // do something }}
```

```
class Test implements SampleCloseable { public void close() throws Exception { // do something }}
```

```
class Test implements SampleCloseable { public void close() throws FileNotFoundException { // do something }}
```

```
class Test extends SampleCloseable { public void close() throws java.io.IOException { // do something }}
```

```
class Test implements SampleCloseable { public void close() { // do something }}
```

Respuesta correcta

```
class Test implements SampleCloseable { public void close() throws java.io.IOException { // do something }}
```

```
class Test implements SampleCloseable { public void close() throws FileNotFoundException { // do something }}
```

```
class Test implements SampleCloseable { public void close() { // do something }}
```

What is the result?*

0/1

```
class MySort implements Comparator<Integer> {
    public int compare(Integer x, Integer y) {
        return y.compareTo(x);
    }
}
```

And the code fragment:

```
Integer[] primes = {2, 7, 5, 3};
MySort ms = new MySort();
Arrays.sort(primes, ms);
for (Integer p2 : primes) { System.out.print(p2 + " "); }
```

2 3 5 7

2 7 5 3

7 5 3 2

Compilation fails.

Respuesta correcta

7 5 3 2

Which two possible outputs?*

0/1

```
public class Main {  
    public static void main(String[] args) throws Exception {  
        doSomething();  
    }  
    private static void doSomething() throws Exception {  
        System.out.println("Before if clause");  
        if (Math.random() > 0.5) { throw new Exception();}  
        System.out.println("After if clause");  
    }  
}
```

Before if clause Exception in thread "main" java.lang.Exception at Main.doSomething (Main.java:21) at Main.main (Main.java:15).

Before if clause Exception in thread "main" java.lang.Exception at Main.doSomething (Main.java:21) at Main.main (Main.java:15) After if clause.

Exception in thread "main" java.lang.Exception at Main.doSomething (Main.java:21) at Main.main (Main.java:15).

Before if clause After if clause.

Respuesta correcta

Before if clause Exception in thread "main" java.lang.Exception at Main.doSomething (Main.java:21) at Main.main (Main.java:15).

Before if clause After if clause.

What is the result?*

0/1

```

class MyKeys {
    Integer key;
    MyKeys(Integer k) { key = k; }
    public boolean equals(Object o) {
        return ((MyKeys) o).key == this.key;
    }
}

```

And this code snippet:

```

Map m = new HashMap();
MyKeys m1 = new MyKeys(1);
MyKeys m2 = new MyKeys(2);
MyKeys m3 = new MyKeys(1);
MyKeys m4 = new MyKeys(new Integer(2));
m.put(m1, "car");
m.put(m2, "boat");
m.put(m3, "plane");
m.put(m4, "bus");
System.out.print(m.size());

```

2

3

4

Compilation fails.

Respuesta correcta

4

What is the result?*

0/1


```

public static void main(String[] args) {
    String color = "Red";
    switch (color) {
        case "Red":
            System.out.println("Found Red");
        case "Blue":
            System.out.println("Found Blue");
        case "White":
            System.out.println("Found White");
            break;
        Default:
            System.out.println("Found Default");
    }
}

```

Found Red.

Found Red Found Blue.

Found Red Found Blue Found White.

Found Red Found Blue Found White Found Default.

Respuesta correcta

Found Red Found Blue Found White.

Which two statements are true?*

0/1

An abstract class can implement an interface.

An abstract class can be extended by an interface.

An interface CANNOT be extended by another interface.

An interface can be extended by an abstract class.

An abstract class can be extended by a concrete class.

An abstract class CANNOT be extended by an abstract class

Respuesta correcta

An abstract class can implement an interface.

An abstract class can be extended by a concrete class.

The pom.xml file is the configuration file For:*

0/1

Ant
Gradle

Maven

Respuesta correcta

Maven

The SINGLETON pattern allows:*

0/1

Have a single instance of a class and this instance cannot be used by other classes

Having a single instance of a class, while allowing all classes have access to that instance.
Having a single instance of a class that can only be accessed by the first method that calls it.

Respuesta correcta

Having a single instance of a class, while allowing all classes have access to that instance.

What is the result?*

0/1

```
class X {  
    static void m(int i) {  
        i += 7;  
    }  
    public static void main(String[] args) {  
        int i = 12;  
        m(i);  
        System.out.println(i);  
    }  
}
```

7

12

19

Compilation fails.

An exception is thrown at run time

Respuesta correcta

12

Which statement is true?*

0/1

```
class ClassA {
    public int numberOfInstances;
    protected ClassA(int numberOfInstances) {
        this.numberOfInstances = numberOfInstances;
    }
}
public class ExtendedA extends ClassA {
    private ExtendedA(int numberOfInstances) {
        super(numberOfInstances);
    }
    public static void main(String[] args) {
        ExtendedA ext = new ExtendedA(420);
        System.out.print(ext.numberOfInstances);
    }
}
```

420 is the output.

An exception is thrown at runtime.

All constructors must be declared public.

Constructors CANNOT use the private modifier.

Constructors CANNOT use the protected modifier.

Respuesta correcta

420 is the output.

Which is true?*

1/1

```

5. class Building { }
6.     public class Barn extends Building {
7.         public static void main(String[] args) {
8.             Building build1 = new Building();
9.             Barn barn1 = new Barn();
10.            Barn barn2 = (Barn) build1;
11.            Object obj1 = (Object) build1;
12.            String str1 = (String) build1;
13.            Building build2 = (Building) barn1;
14.        }
15.    }

```

Which is true?

If line 10 is removed, the compilation succeeds.

If line 11 is removed, the compilation succeeds.

If line 12 is removed, the compilation succeeds.

If line 13 is removed, the compilation succeeds.

More than one line must be removed for compilation to succeed.

What is the result?*

0/1

```

public static void main(String[] args) {
    int [][] array2D = { {0, 1, 2}, {3, 4, 5, 6} };
    System.out.print(array2D[0].length + "" );
    System.out.print(array2D[1].getClass().isArray() + "" );
    System.out.println(array2D[0][1]);
}

```

3false1

2true3

2false3

3true1

3false3

2true1

2false1

Respuesta correcta

3true1

What is the DTO pattern used for?*

0/1

- To implement the data access layer
- To exchange data between processes
- To implement the presentation layer.

Respuesta correcta

To implement the data access layer

What is the result if the integer value is 33?*

1/1

```

public static void main(String[] args) {
    if (value >= 0) {
        if (value != 0) {
            System.out.print("the ");
        } else {
            System.out.print("quick ");
        }
        if (value < 10) {
            System.out.print("brown ");
        }
        if (value > 30) {
            System.out.print("fox ");
        } else if (value < 50) {
            System.out.print("jumps ");
        } else if (value < 10) {
            System.out.print("over ");
        } else {
            System.out.print("the ");
        }
        if (value > 10) {
            System.out.print("lazy ");
        } else {
            System.out.print("dog ");
        }
        System.out.print("... ");
    }
}

```

The fox jump lazy ?
The fox lazy ?

Quick fox over lazy ?

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);
        for (String s : sa) { System.out.println(nf.parse(s)); }
    }
}
```

111.234 222.567
 111.234 222.568
 111.234 222.5678

An exception is thrown at runtime.

What is the result?*

0/1

```
11. class Person {
12.     String name = "No name";
13.     public Person (String nm) { name = nm}
14. }
15.
16. class Employee extends Person {
17.     String empID = "0000";
18.     public Employee (String id) { empID =
19.         id; }
20. }
21. public class EmployeeTest {
22.     public static void main(String[] args)
23.     {
24.         Employee e = new Employee("4321");
25.         System.out.println(e.empID);
26.     }
27. }
```

4321.
 0000.

An exception is thrown at runtime.

Compilation fails because of an error in line 18.

Respuesta correcta

Compilation fails because of an error in line 18.

What is the result?*

0/1

Given

```
public class SuperTest {
    public static void main(String[] args) {
        //statement1
        //statement2
        //statement3
    }
}
class Shape {
    public Shape() {
        System.out.println("Shape: constructor");
    }
    public void foo() {
        System.out.println("Shape: foo");
    }
}
class Square extends Shape {
    public Square() {
        super();
    }
    public Square(String label) {
        System.out.println("Square: constructor");
    }
    public void foo() {
        super.foo();
    }
    public void foo(String label) {
        System.out.println("Square: foo");
    }
}
```

What should statement1, statement2, and statement3, be respectively, in order to produce the result?

Shape: constructor
Shape: foo
Square: foo

Square square = new Square ("bar"); square.foo ("bar"); square.foo();
Square square = new Square ("bar"); square.foo ("bar"); square.foo ("bar");
Square square = new Square (); square.foo (); square.foo(bar);

Square square = new Square (); square.foo (); square.foo("bar");
Square square = new Square (); square.foo (); square.foo ();

Respuesta correcta

Square square = new Square (); square.foo (); square.foo("bar");

Which code fragment is illegal?*

0/1

```
Class Base1 { abstract class Abs1 {} }  
Abstract class Abs2 { void doit() {} }
```

```
class Base2 { abstract class Abs3 extends Base2 {} }  
class Base3 { abstract int var1 = 89; }
```

Respuesta correcta

```
class Base3 { abstract int var1 = 89; }
```

What is the result?*

1/1

```
public static void main(String[] args) {  
    System.out.println("Result: " + 2 + 3 + 5);  
    System.out.println("Result: " + 2 + 3 * 5);  
}
```

Result: 10 Result: 30

Result: 25 Result: 10

Result: 235 Result: 215

Result: 215 Result: 215

Compilation fails.

What is the result?*

0/1

```

public class MyStuff {
    String name;
    MyStuff (String n) { name = n; }
    public static void main (String[] args) {
        MyStuff m1 = new MyStuff ("guitar");
        MyStuff m2 = new MyStuff ("tv");
        System.out.println (m2.equals(m1));
    }
    public boolean equals (Object o) {
        MyStuff m = (MyStuff) o;
        if (m.name != null) { return true; }
        return false;
    }
}

```

The output is true and MyStuff fulfills the Object.equals() contract

The output is false and MyStuff fulfills the Object.equals() contract

The output is true and MyStuff does NOT fulfill the Object.equals() contract.

The output is false and MyStuff does NOT fulfill the Object.equals() contract

Respuesta correcta

The output is true and MyStuff does NOT fulfill the Object.equals() contract.

Which one is valid as a replacement for foo?*

0/1

```

public static void main(String[] args) {
    Boolean b1 = true;
    Boolean b2 = false;
    int i = 0;
    while (foo) { }
}

```

b1.compareTo(b2)

i = 1

i == 2? -1:0

foo.equals("bar")

Respuesta correcta

foo.equals("bar")

What is the result?*

0/1

```
interface Rideable {
    String getGait();
}

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);
    }
}
```

16 mph, lope

24 mph, lope.

27 mph, lope.

Compilation fails

Respuesta correcta

Compilation fails

What is the result?*

0/1

```
class X {  
    String str = "default";  
    X(String s) { str = s; }  
    void print() { System.out.println(str); }  
    public static void main(String[] args) { new X("hello").print(); }  
}
```

Hello

Default

Compilation fails.

The program prints nothing.

An exception is thrown at run time.

Respuesta correcta

Hello

What is the result?*

1/1

```
public static void main(String[] args) {  
    int[] array = {1, 2, 3, 4, 5};  
    System.arraycopy(array, 2, array, 1, 2);  
    System.out.print(array[1]);  
    System.out.print(array[4]);  
}
```

14

15

24

25

34

35

What is the result?*

0/1

```
int a = 10;  
int b = 37;  
int z = 0;  
int w = 0;  
if (a == b) { z = 3; } else if (a > b) { z = 6; }  
w = 10 * z;
```

0
30

60

Respuesta correcta

0

What is the result?*

0/1

```
public class DoWhile {  
    public static void main(String[] args) {  
        int ii = 2;  
        do {  
            System.out.println(ii);  
        } while (--ii);  
    }  
}
```

2 1 2
1 0
null

An infinite loop.
Compilation fails

Respuesta correcta

Compilation fails

What changes will make this code compile?*

1/1

```
class X {  
    X() { }  
    private void one() { }  
}  
public class Y extends X {  
    Y() {}  
    private void two() {  
        one();  
    }  
    public static void main(String[] args) {  
        new Y().two();  
    }  
}
```

Adding the public modifier to the declaration of class X.

Adding the protected modifier to the X() constructor.

Changing the private modifier on the declaration of the one() method to protected.

Removing the Y() constructor.

What changes will make this code compile?*

1/1

```

class X {
    X() { }
    private void one() { }
}
public class Y extends X {
    Y() { }
    private void two() {
        one();
    }
    public static void main(String[] args) {
        new Y().two();
    }
}

```

Adding the public modifier to the declaration of class X. (

Adding the protected modifier to the X() constructor.

Changing the private modifier on the declaration of the one() method to protected.

Removing the Y() constructor

Removing the private modifier from the two() method.

Which two declarations will compile?*

0/1

```

14.    public static void main(String[] args) {
15.        Int a, b, c = 0;
16.        int a, b, c;
17.        int g, int h, int i = 0;
18.        int d, e, f;
19.        Int k, l, m, = 0;
20.    }

```

Line 15.

Line 16.

Line 17.

Line 18.

Line 19.

Line 20.

Respuesta correcta

Line 16.

Line 18.

Which three methods, inserted individually at line, will correctly complete class Two(Choose three)?*

0/1

```
10. class One {  
11.     void foo() { }  
12. }  
13. class Two extends One {  
14.     //insert method here  
15. }
```

```
int foo() { /* more code here */ }
```

```
void foo() { /* more code here */ }
```

```
public void foo() { /* more code here */ }
```

```
private void foo() { /* more code here */ }
```

```
protected void foo() { /* more code here */ }
```

Respuesta correcta

```
int foo() { /* more code here */ }
```

```
public void foo() { /* more code here */ }
```

```
protected void foo() { /* more code here */ }
```

What is the result?*

0/1

```

try {
    // assume "conn" is a valid Connection object
    // assume a valid Statement object is created
    // assume rollback invocations will be valid
    // use SQL to add 10 to a checking account
    Savepoint s1 = conn.setSavePoint();
    // use SQL to add 100 to the same checking account
    Savepoint s2 = conn.setSavePoint();
    // use SQL to add 1000 to the same checking account
    // insert valid rollback method invocation here
} catch (Exception e) { }

```

If conn.rollback(s1) is inserted, account will be incremented by 10.

If conn.rollback(s1) is inserted, account will be incremented by 1010.

If conn.rollback(s2) is inserted, account will be incremented by 100

If conn.rollback(s2) is inserted, account will be incremented by 110.

If conn.rollback(s2) is inserted, account will be incremented by 1110

Respuesta correcta

If conn.rollback(s1) is inserted, account will be incremented by 10.

If conn.rollback(s2) is inserted, account will be incremented by 110.

What is the result?*

0/1

```

class Alpha { String getType() { return "alpha"; } }
class Beta extends Alpha { String getType() { return "beta"; } }
public class Gamma extends Beta { String getType() { return "gamma"; }
    public static void main(String[] args) {
        Gamma g1 = new Alpha();
        Gamma g2 = new Beta();
        System.out.println(g1.getType() + " " + g2.getType());
    }
}

```

Alpha beta

Beta beta.

Gamma gamma.

Compilation fails.

Respuesta correcta

Compilation fails.

Whats is true about the class Wow?*

0/1

```
public abstract class Wow {  
    private int wow;  
    public Wow(int wow) { this.wow = wow; }  
    public void wow() {}  
    private void wowza() {}  
}
```

It compiles without error.

It does not compile because an abstract class cannot have private methods

It does not compile because an abstract class cannot have instance variables.

It does not compile because an abstract class must have at least one abstract method.

It does not compile because an abstract class must have a constructor with no arguments.

Respuesta correcta

It compiles without error.

Whats is the result?*

0/1

```
public static void main(String[] args) {  
    System.out.println("Result:" + 3 + 5);  
    System.out.println("Result:" + (3 + 5));  
}
```

Result: 8 Result: 8

Result: 35 Result: 8

Result: 8 Result: 35

Result: 35 Result: 35

Respuesta correcta

Result: 35 Result: 8

How many times is 2 printed?*

0/1

Zero.

Once.

Twice.

Thrice.

It is not printed because compilation fails.

Respuesta correcta

Once.

What is printed out when the program is excuted?*

1/1

```
public class MainMethod {  
    void main() {  
        System.out.println("one");  
    }  
    static void main(String args) {  
        System.out.println("two");  
    }  
    public static final void main(String[] args) {  
        System.out.println("three");  
    }  
    void mina(Object[] args) {  
        System.out.println("four");  
    }  
}
```

one

two
three

four
There is no output.

What is the result?*

0/1

```
import java.util.*;
public class MyScan {
    public static void main(String[] args) {
        String in = "1 a 10 . 100 1000";
        Scanner s = new Scanner(in);
        int accum = 0;
        for (int x = 0; x < 4; x++) {
            accum += s.nextInt();
        }
        System.out.println(accum);
    }
}
```

11
111

1111
An exception is thrown at runtime.

Respuesta correcta

An exception is thrown at runtime.

What is the result?*

0/1

```

public class X {
    public static void main(String[] args) {
        String theString = "Hello World";
        System.out.println(theString.charAt(11));
    }
}

```

There is no output.

d is output.

A StringIndexOutOfBoundsException is thrown at runtime.

An ArrayIndexOutOfBoundsException is thrown at runtime.

A NullPointerException is thrown at runtime.

A StringArrayIndexOutOfBoundsException is thrown at runtime.

Respuesta correcta

A StringIndexOutOfBoundsException is thrown at runtime.

What will make this code compile and run?*

0/1

```

01. public class Simple {
02.
03.     public float price;
04.     public static void main(String[] args) {
05.
06.         Simple price = new Simple();
07.         price = 4;
08.     }
09. }

```

Change line 3 to the following: public int price;

Change line 7 to the following: int price = new Simple();

Change line 7 to the following: float price = new Simple ();

Change line 7 to the following: price = 4f;

Change line 7 to the following: price.price = 4;

Respuesta correcta

Change line 7 to the following: price.price = 4;

What value of x, y, z will produce the following

result?

1234,1234,1234 -----, 1234, -----*

0/1

```
public static void main(String[] args) {  
    // insert code here  
    int j = 0, k = 0;  
    for (int i = 0; i < x; i++) {  
        do {  
            k = 0;  
            while (k < z) {  
                k++;  
                System.out.print(k + " ");  
            }  
            System.out.println(" ");  
            j++;  
        } while (j < y);  
        System.out.println("---");  
    }  
}
```

int x = 4, y = 3, z = 2;

int x = 3, y = 2, z = 3;

int x = 2, y = 3, z = 3;

int x = 4, y = 2, z = 3;

int x = 2, y = 3, z = 4;

Respuesta correcta

int x = 2, y = 3, z = 4;

In the Java collections framework a Set is:*

0/1

A collection that cannot contain duplicate elements.

An ordered collection that can contain duplicate elements

An object that maps value key sets and cannot contain values Duplicates

Respuesta correcta

A collection that cannot contain duplicate elements.

What is the result?*

0/1

```
public class SampleClass {  
    public static void main(String[] args) {  
        AnotherSampleClass asc = new AnotherSampleClass();  
        SampleClass sc = new SampleClass();  
        sc = asc;  
        System.out.println("sc: " + sc.getClass());  
        System.out.println("asc: " + asc.getClass());  
    }  
}  
class AnotherSampleClass extends SampleClass { }
```

sc: class.Object asc: class.AnotherSampleClass

sc: class.SampleClass asc: class.AnotherSampleClass

sc: class.AnotherSampleClass asc: class.SampleClass

sc: class.AnotherSampleClass asc: class.AnotherSampleClass

Respuesta correcta

sc: class.AnotherSampleClass asc: class.AnotherSampleClass

What is the result?*

0/1

```

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(); }
}

```

Compilation fails.

Atom granite.

Granite granite.

Atom granite granite.

An exception is thrown at runtime.

Atom granite atom granite.

Respuesta correcta

Atom granite atom granite.

What is the result?

0/1

```

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);
        for (String s : sa) { System.out.println(nf.parse(s)); }
    }
}

```

111.234 222.567

An exception is thrown at runtime

111.234 222.568
111.234 222.5678

Correct answer

111.234 222.5678

What is the result?

0/1

```
public static void main(String[] args) {  
    int[] array = { 1, 2, 3, 4, 5 };  
    System.arraycopy(array, 2, array, 1, 2);  
    System.out.print(array[1]);  
    System.out.print(array[4]);  
}
```

14

15

34

24

35

25

Correct answer

35

Which declaration initializes a boolean variable?

0/1

- boolean j = (1 < 5);
- boolean m = null;

- boolean h = 1;
- boolean k = 0;

Correct answer

- boolean j = (1 < 5);

Which two will compile, and can be run successfully using the following command? java Fred1 Hello walls.

0/1

```
abstract class Fred1 {public static void main(String[] args){System.out.println(args[2]);}}
```

```
class Fred 1{ public static void main(String args) { System.out.println(args[1]); } }
```

```
class Fred1 { public static void main(String[] args) { System.out.println(args[1]); } }
```

```
class Fred1 { public static void main(String[] args) { System.out.println(args); } }
```

Correct answer

```
class Fred1 { public static void main(String[] args) { System.out.println(args); } }
```

How many times is 2 printed?

0/1

```
public static void main(String[] args) {  
    String[] table = {"aa", "bb", "cc"};  
    int ii = 0;  
    for (String ss : table) {  
        while (ii < table.length) {  
            System.out.println(ii); ii++;  
            break;  
        }  
    }  
}
```

Thrice

It is not printed because compilation fails

Twice

Zero

Once

Correct answer

Once

Which five methods, inserted independently at line 5, will compile? (Choose five).

0/1


```

1 public class Blip{
2     protected int blipvert(int x){ return 0 ;}
3 }
4 class Vert extends Blip{
5     //insert code here
6 }

```

Protected long blipvert(long x) { return 0; }

Public int blipvert(int x) { return 0; }

Private int blipvert(long x) { return 0; }

Protected long blipvert(int x, int y) { return 0; }

Protected long blipvert(int x) { return 0; }

Protected int blipvert(long x) { return 0; }

Private int blipvert(int x) { return 0; }

Correct answer

Protected long blipvert(long x) { return 0; }

Public int blipvert(int x) { return 0; }

Private int blipvert(long x) { return 0; }

Protected long blipvert(int x, int y) { return 0; }

Protected int blipvert(long x) { return 0; }

Which two are valid instantiations and initializations of a multidimensional array?

0/1

array2D[0][0] = 1; array2D[0][1] = 2; array2D[1][0] = 3; array2D[1][1] = 4;

array3D[0][0] = array; array3D[0][1] = array; array3D[1][0] = array; array3D[0][1] = array;

int[] array2D = {0, 1};

int[][] array3D = {{0, 1}, {2, 3}, {4, 5}}; int[] array = {0, 1}; int[][] array3D = new int[2][2][2];

int[] array2D = {{0, 1, 2, 4}{5, 6}}; int[] array2D = new int[2][2];

Correct answers

array2D[0][0] = 1; array2D[0][1] = 2; array2D[1][0] = 3; array2D[1][1] = 4;

int[][] array3D = {{0, 1}, {2, 3}, {4, 5}}; int[] array = {0, 1}; int[][] array3D = new int[2][2][2];

What is the result?

0/1

```
public static void main(String[] args) {  
    int[][] array2D = {{0,1,2}, {3,4,5,6}};  
    System.out.print(array2D[0].length + "");  
    System.out.print(array2D[1].getClass().isArray() + "");  
    System.out.println(array2D[0][1]);  
}
```

3false3
3true1
3false1

2false1
2false3
2true3
2true1

Correct answer

3true1

In Java the difference between throws and throw is:

1/1

Throws indicates the type of exception that the method does not handle and throw an exception.

Throws throws an exception and throw indicates the type of exception that the method.
Throws is used in methods and throw in constructors.

The standard API for accessing databases in Java is:

0/1

JPA/Hibernate

JDBC
ODBC

Correct answer

JDBC

What changes will make this code compile?

0/1

```

class X {
    X() {}
    private void one() {}
}
public class Y extends X{
    Y() {}
    private void two() {
        one();
    }
    public static void main(String[] args) {
        new Y().two();
    }
}

```

Adding the public modifier to the declaration of class X.

Adding the protected modifier to the X() constructor.

Changing the private modifier on the declaration of the one() method to protected.

Removing the private modifier from the two() method.

Removing the Y() constructor.

Correct answer

Changing the private modifier on the declaration of the one() method to protected.

Which statement is true?

0/1

```

class ClassA{
    public int numberOfInstances;
    protected ClassA(int numberOfInstances){
        this.numberOfInstances = numberOfInstances;
    }
}
public class ExtendedA extends ClassA{
    private ExtendedA(int numberOfInstances){
        super(numberOfInstances);
    }
    public static void main(String[] args){
        ExtendedA ext = new ExtendedA(420);
        System.out.print(ext.numberOfInstances);
    }
}

```

Constructors CANNOT use the protected modifier
Constructors CANNOT use the private modifier

An exception is thrown at runtime.
All constructors must be declared public.
420 is the output.

Correct answer
420 is the output.

Which three are valid? (Choose three).

0/1

```

class ClassA{}
class ClassB extends ClassA{}
class ClassC extends ClassA{}

```

And:

```

ClassA p0 = new ClassA();
ClassB p1 = new ClassB();
ClassC p2 = new ClassC();
ClassA p3 = new ClassB();
ClassA p4 = new ClassC();

```

p1 = p2;
p2 = (ClassC)p4;

```
p2 = p4;  
p1 = (ClassB)p3;  
p0 = p1;
```

```
p2 = (ClassC)p1;
```

Correct answer

```
p2 = (ClassC)p4;  
p1 = (ClassB)p3;  
p0 = p1;
```

What is the result?

0/1

```
class X {  
    static void m(int i) {  
        i+=7;  
    }  
    public static void main(String[] args) {  
        int i = 12;  
        m(i);  
        System.out.println(i);  
    }  
}
```

Compilation fails.

12

7

19

An exception is thrown at run time.

Correct answer

12

What is the result?

0/1

```
import java.util.*;
public class App{
    public static void main(String[]args){
        List p = new ArrayList();
        p.add(7);
        p.add(1);
        p.add(5);
        p.add(1);
        p.remove(1);
        System.out.println(p);
    }
}
```

[7, 1]
 [7, 1, 5, 1]
 [7, 5, 1]
 [7, 5]

Correct answer

[7, 5, 1]

What is the result?

0/1

```
public class SampleClass{
    public static void main(String[]args{
        AnotherSampleClass asc = new AnotherSampleClass();
        SampleClass sc = new SampleClass();
        sc = asc;
        System.out.println("sc:" + sc.getClass());
        System.out.println("asc:" + asc.getClass());
    }
}
class AnotherSampleClass extends SampleClass{}
```

sc: class.Object asc: class.AnotherSampleClass

sc: class.AnotherSampleClass asc: class.SampleClass

sc: class.AnotherSampleClass asc: class.AnotherSampleClass

sc: class.SampleClass asc: class.AnotherSampleClass

Correct answer

sc: class.AnotherSampleClass asc: class.AnotherSampleClass

What is the result?

0/1

```
public class Dowhile{  
    public static void main(String[] args){  
        int ii = 2;  
        do{  
            System.out.println(ii);  
        }while (--ii);  
    }  
}
```

Compilation fails.

2 1 2

1 0

null.

An infinite loop.

Correct answer

Compilation fails.

What is the best way to test that the values of h1 and h2 are the same?

0/1

```
public static void main(String[] args){  
    String h1 = "Bob";  
    String h2 = new String("Bob");  
}
```

if (h1.equals(h2)).

if (h1 == h2).

if (h1.toString() == h2.toString()).

if (h1.same(h2)).

Correct answer

if (h1.equals(h2)).

The pom.xml file is the configuration file For:

0/1

Maven.

Ant.

Gradle

Correct answer

Maven.

The BUILDER pattern is used to:

0/1

Having several constructor methods in a class.

Simplify the creation of complex objects.

Implement the constructor of a class.

Correct answer

Simplify the creation of complex objects.

What is the result?

0/1

```
public class X{  
    public static void main(String[] args){  
        String theString = "Hello World";  
        System.out.println(theString.charAt(11));  
    }  
}
```

A StringIndexOutOfBoundsException is thrown at runtime.

There is no output.

A StringArrayIndexOutOfBoundsException is thrown at runtime.

d is output.

A NullPointerException is thrown at runtime.

An ArrayIndexOutOfBoundsException is thrown at runtime.

Correct answer

A StringIndexOutOfBoundsException is thrown at runtime.

What is the result?

0/1

```
1 public class Boxer1{
2     Integer i;
3     int x;
4     public Boxer1(int y){
5         x = i+y;
6         System.out.println(x);
7     }
8     public static void main(String[] args){
9         new Boxer1(new Integer(4));
10    }
11 }
```

A NullPointerException occurs at runtime.
The value "4" is printed at the command line.
An IllegalStateException occurs at runtime.
Compilation fails because of an error in line 9.

Compilation fails because of an error in line 5.
A NumberFormatException occurs at runtime.

Correct answer

A NullPointerException occurs at runtime.

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

0/1

```

13 public class Speak{
14     public static void main (String[]args){
15         Speak speakIT =new Tell();
16         Tell tellIt = new Tell();
17         speakIT.tellItLikeItIs();
18         (Truth) speakIT'.tellItLikeItIs();
19         ((Truth) speakIT'.tellItLikeItIs();
20         tellIt.tellItLikeItIs();
21         (Truth) tellItLikeItIs();
22         ((Truth) tellItLikeItIs();
23     }
24 }

class Tell extends Speak implements Truth{
    @Override
    public void tellItLikeItIs(){
        System.out.println("Right on!");
    }
}

interface Truth{
    public void tellItLikeItIs();
}

```

Line 22.

Line 17.

Line 20.

Line 21

Line 19.

Line 18.

Correct answer

Line 22.

Line 20.

Line 19.

0/1

We have the following Java class:

```
package gal.dicoruna.example;  
public class C {  
    protected String v;  
    ...  
}
```

From the class, the package, subclasses.

Any site but only read if it is outside the package.

From the class, the package, subclasses and all sites.

Correct answer

From the class, the package, subclasses.

What is printed out when the program is executed?

*

1/1

```
public class MainMethod{  
    void main(){  
        System.out.println("one");  
    }  
    static void main(String args){  
        System.out.println("two");  
    }  
    public static final void main(String[] args){  
        System.out.println("three");  
    }  
    void mina(Object[] args){  
        System.out.println("four");  
    }  
}
```

three

four

one
two
There is no output

Which two statements are true?

*

0/1

An abstract class can be extended by a concrete class.

An abstract class can be extended by an interface.

An abstract class can implement an interface.

An interface CANNOT be extended by another interface.

An interface can be extended by an abstract class.

An abstract class CANNOT be extended by an abstract class.

Correct answer

An abstract class can implement an interface.

An abstract class can be extended by a concrete class.

Which declaration initializes a boolean variable?

*

0/1

- boolean h = 1;
- boolean j = (1 < 5);
- boolean k = 0;

- boolean m = null;

Correct answer

- boolean j = (1 < 5);

What is the result?

*

0/1

```

public class DoWhile {
    public static void main(String[] args) {
        int ii = 2;
        do{
            System.out.println(ii);
        }while(--ii);
    }
}

```

Compilation fails.

1 0

2 1 2

null

An infinite loop

Correct answer

Compilation fails.

Which two will compile, and can be run successfully using the following command?

java Fred1 hello walls

*

0/1

class Fred1 { public static void main(String[] args) { System.out.println(args[1]); } }

abstract class Fred1 { public static void main(String[] args) { System.out.println(args[2]); } }

class Fred1 { public static void main(String[] args) { System.out.println(args); } }

class Fred 1 { public static void main(String args) { System.out.println(args[1]); } }

Correct answer

class Fred1 { public static void main(String[] args) { System.out.println(args); } }

What is the result?

*

0/1

```
class Foo{
    public void addFive(){ a += 5; System.out.println("f");}
}
class Bar extends Foo{
    public int a = 8;
    public void addFive(){ this.a += 5; System.out.println("b");}
}
Invoked with:
    Foo f = new Bar();
    f.addFive();
    System.out.println(f.a);
```

b 3

Compilation fails

f 8

b 8

f 13

f 3

An exception is thrown at runtime.

b 13

Correct answer

Compilation fails

What is the result?

*

0/1

```
public class ScopeTest{
    int z;
    public static void main(String[] args){
        ScopeTest myScope = new ScopeTest();
        int z = 6;
        System.out.println(z);
        myScope.doStuff();
        System.out.print(z);
        System.out.print(myScope.z);
    }
    void doStuff(){
        int z = 5;
        doStuff2();
        System.out.print(myScope.z);
    }
    void doStuff2(){
        z = 4;
    }
}
```

6554

6564

6565

Compilation fails

6566

Correct answer

Compilation fails

What is the result?

*

0/1

```

public class Barn{
    public static void main(String[] args){
        new Barn().go("hi", 1);
        new Barn().go("hi", "world", 2);
    }
    public void go(String... y, int x){
        System.out.print(y[y.length - 1] + " ");
    }
}

```

world world.

hi hi

hi world

An exception is thrown at runtime.

Compilation fails

Correct answer

Compilation fails

What is the result?

*

1/1

```

public class Test{
    public static void main(String[] args){
        int b = 4;
        b--;
        System.out.print(--b);
        System.out.println(b);
    }
}

```

12

33

22

32

What is the result?

*

1/1

```
public class MyStuff{
    String name;
    MyStuff (String n){ name = n;}
    public static void main (String[] args){
        MyStuff m1 = new MyStuff ("guitar");
        MyStuff m2 = new MyStuff ("tv");
        System.out.println (m2.equals(m1));
    }
    public boolean equals (Object o){
        MyStuff m = (MyStuff) o;
        if(m.name != null){ return true; }
        return false;
    }
}
```

The output is true and MyStuff does NOT fulll the Object.equals() contract

The output is false and MyStuff does NOT fulll the Object.equals() contract

The output is true and MyStuff fullls the Object.equals() contract

The output is false and MyStuff fullls the Object.equals() contract

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

*

1/1

```
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 is thrown with x != 6 as additional output.

Truthy.java does NOT compile.

Truthy.java compiles and an AssertionError is thrown with no additional output

Which code fragment is illegal?

*

1/1

```
class Base3 { abstract int var1 = 89; }
```

```
class Base2 { abstract class Abs3 extends Base2 { } }
```

```
Abstract class Abs2 { void doit() { } }
```

```
Class Base1 { abstract class Abs1 { } }
```

What is the result?

*

1/1

```
1 public class Boxer1{
2     Integer i;
3     int x;
4     public Boxer1(int y){
5         x = i+y;
6         System.out.println(x);
7     }
8     public static void main(String[] args){
9         new Boxer1(new Integer(4));
10    }
11 }
```

Compilation fails because of an error in line 5

An IllegalStateException occurs at runtime.

Compilation fails because of an error in line 9

A NullPointerException occurs at runtime

A NumberFormatException occurs at runtime

The value "4" is printed at the command line.

What is the result? Given:

*

0/1

0 tom 0 jerry 1 tom 1 jerry 2 tom 2 jerry

0 tom 0 jerry 1 tom 2 tom 2 jerry

0 tom 0 jerry 2 tom 2 jerry

0 tom 0 jerry 1 tom 1 jerry

Correct answer

0 tom 0 jerry 1 tom 2 tom 2 jerry

What is the result when this program is executed?

*

1/1

```
public class Student{
    public String name = "";
    public int age = 0;
    public String major = "Undeclared";
    public boolean fulltime = true;
    public void display(){
        System.out.println("Name:" + name + "Major:" + major);
    }
    public boolean isFullTime(){
        return fulltime;
    }
}

And:
class TestStudent {
    public static void main(String[] args){
        Student bob = new Student();
        Student jian = new Student();
        bob.name = "Bob";
        bob.age = 19;
        jian = bob;
        jian.name = "Jian";
        System.out.println("Bob's Name:" + bob.name);
    }
}
```

Nothing prints

Bob's name.

Bob's Name: Bob.

Bob's Name: Jian

The standard API for accessing databases in Java is:

*

0/1

ODBC.

JPA/Hibernate

JDBC

Correct answer

JDBC

What is the result?

*

0/1

```
public class SampleClass{
    public static void main(String[] args){
        AnotherSampleClass asc = new AnotherSampleClass();
        SampleClass sc = new SampleClass();
        sc = asc;
        System.out.println("sc:" + sc.getClass());
        System.out.println("asc:" + asc.getClass());
    }
}
class AnotherSampleClass extends SampleClass{}
```

sc: class.AnotherSampleClass asc: class.AnotherSampleClass

sc: class.Object asc: class.AnotherSampleClass

sc: class.AnotherSampleClass asc: class.SampleClass

sc: class.SampleClass asc: class.AnotherSampleClass

Correct answer

sc: class.AnotherSampleClass asc: class.AnotherSampleClass

Which class has a default constructor?

*

0/1

```
class X{}  
class Y{  
    Y(){}  
}  
class Z{  
    Z(int i){}}
```

- Z only.
- X only.
- X, Y and Z
- X and Y
- X and Z
- Y only
- Y and Z

Correct answer

X and Y

What is the result?

*

1/1

```

interface Rideable{
    String getGaint();
}

public class Camel implements Rideable{
    int weight = 2;
    String getGaint(){
        return "mph, lope";
    }
    void go(int speed){
        ++speed;
        Weight++;
        int walkrate = speed * weight;
        System.out.print(walkrate + getGaint());
    }
    public static void main (String[] args){
        new Camel().go(8);
    }
}

```

24 mph, lope
 Opción 3
 27 mph, lope
 Compilation fails

16 mph, lope

What is the result?

*

0/1

```

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);
        for (String s : sa){ System.out.println(nf.parse(s));}
    }
}

```

111.234 222.568
 An exception is thrown at runtime.

111.234 222.567
111.234 222.5678

Correct answer

111.234 222.5678

Which two may precede the word "class" in a class declaration?

*

0/1

Public
Static
Synchronized
Local

Volatile

Correct answers

Public
Static

What values of x, y, z will produce the following
result? 1234 1234 1234 --- 1234 ---

*

0/1

```

public static void main(String[] args){
    //insert code here
    int j = 0, k = 0;
    for (int i = 0; i<x; i++){
        do{
            k = 0;
            while (k<z){
                k++;
                System.out.print(k + " ");
            }
            System.out.println(" ");
            j++;
        }while (j<y);
        System.out.println("----");
    }
}

```

- int x = 2, y = 3, z = 4;
 - int x = 4, y = 2, z = 3;
 - int x = 2, y = 3, z = 3;

int x = 3, y = 2, z = 3;
 - int x = 4, y = 3, z = 2;

Correct answer

- int x = 2, y = 3, z = 4;

What is the result?

*

0/1

```

public class Test {
    public static void main(String[] args){
        int[][] array = {{0},{0,1},{0,2,4},{0,3,6,9},{0,4,8,12,16}};
        System.out.println(array[4][1]);
        System.out.println(array[1][4]);
    }
}

```


An IllegalArgumentException is thrown at run time

4 An ArrayIndexOutOfBoundsException is thrown at run time.

Null 4.

4 Null

Correct answer

4 An ArrayIndexOutOfBoundsException is thrown at run time.

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

*

1/1

```
13 public class Speak{
14     public static void main(String[] args){
15         Speak speakIT = new Tell();
16         Tell tellIt = new Tell();
17         speakIT.tellItLikeItIs();
18         (Truth)speakIT.tellItLikeItIs();
19         ((Truth)speakIT).tellItLikeItIs();
20         tellIt.tellItLikeItIs();
21         (Truth)tellIt.tellItLikeItIs();
22         ((Truth)tellIt).tellItLikeItIs();
23     }
24 }
25
26 class Tell extends Speak implements Truth {
27     @Override
28     public void tellItLikeItIs(){
29         System.out.println("Right on!");
30     }
31 }
32
33 interface Truth{
34     public void tellItLikeItIs();
35 }
```

Line 19

Line 17

Line 22

Line 20
Line 21
Line 18

What is true about the class Wow?

*

0/1

```
public abstract class Wow{  
    private int wow;  
    public Wow(int wow) {this.wow =wow;}  
    public void wow() {}  
    private void wowza() {}  
}
```

It does not compile because an abstract class must have at least one abstractmethod.
It does not compile because an abstract class must have a constructor with noarguments

It does not compile because an abstract class cannot have instance variables
It does not compile because an abstract class cannot have private methods
It compiles without error

Correct answer

It compiles without error

What two keywords may precede the word 'class' in a class declaration?*

0/1

Local

Synchronized

Volatile

Public

Static

Correct answer

Public
Static

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

*

0/1

```
public class Truthy {
    public static void main(String[] args) {
        int x=7;
        assert (x==6)?"x==6":"x!=6";
    }
}
```

Truthy.java does NOT compile.

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

Truthy.java compiles and an AssertionError is thrown with no additional output

Truthy.java compiles and an AssertionError is thrown with x != 6 as additional output

Correct answer

Truthy.java does NOT compile.

What values of x, y, z will produce the following result? 1234 1234 1234 --- 1234 ---

*

0/1

```
1  public static void main(String[] args) {
2      int j = 0, k = 0;
3      for (int i = 0; i < x; i++) {
4          do {
5              k = 0;
6              while (k < z) {
7                  System.out.print(k + "");
8              }
9              System.out.println("");
10             j++;
11         } while (j < y);
12         System.out.println("-");
13     }
14 }
```

- int x = 2, y = 3, z = 3;

- int x = 4, y = 3, z = 2;
- int x = 4, y = 2, z = 3;
- int x = 3, y = 2, z = 3;

- int x = 2, y = 3, z = 4;

Correct answer

- int x = 4, y = 2, z = 3;

What is the result?

*

0/1

```
public class Main {  
    public static void main(String[] args) {  
        int a = 0;  
        a++;  
        System.out.println(a++);  
        System.out.println(a);  
    }  
}
```

0 1

1 2

2 2

1 1

Correct answer

1 2

In Java the difference between throws and throw is:

*

0/1

Throws throws an exception and throw indicates the type of exception that the method.
Throws is used in methods and throw in constructors.

Throws indicates the type of exception that the method does not handle and throw an exception.

Correct answer

Throws indicates the type of exception that the method does not handle and throw an exception.

What is the result?

*

0/1

```
1  class Person{
2      String name = "No name";
3      public Person (String nm) {name nm; }
4      class Employee extends Person {
5          String empID = "0000";
6          public Employee (String id) {empID = "id";}
7      }
8  }
9  public class EmployeeTest {
10     public static void main(String[] args) {
11         Employee e = new Employee ("4321");
12         System.out.println(e.empID);
13     }
```

4321

Compilation fails because of an error in line 9

An exception is thrown at runtime.

0000

Correct answer

Compilation fails because of an error in line 9

What is the result value of w?

*

0/1

```

public class Main {
    public static void main(String[] args) {
        int a = 10;
        int b = 37;
        int z = 0;
        int w = 0;

        if (a == b) {
            z = 3;
        } else if (a > b) {
            z = 6;
        }

        w = 10 * z;

        System.out.println("El valor de w es: " + w);
    }
}

```

60

30

0

Correct answer

0

What is the result?

*

0/1

```

1  public class SampleClass {
2      public static void main(String[] args) {
3          AnotherSampleClass asc = new AnotherSampleClass()
4          SampleClass sc = new SampleClass();
5          sc = asc;
6          System.out.println("sc: " + sc.getClass());
7          System.out.println("asc: " + asc.getClass());
8      }
9  }
10
11  class AnotherSampleClass extends SampleClass {}

```

sc: class.AnotherSampleClass asc: class.SampleClass
sc: class.Object asc: class.AnotherSampleClass

sc: class.SampleClass asc: class.AnotherSampleClass
sc: class.AnotherSampleClass asc: class.AnotherSampleClass

Correct answer

sc: class.AnotherSampleClass asc: class.AnotherSampleClass

What is the result?

*

1/1

```
public class Dowhile {  
    public static void main(String[] args) {  
        int ii = 2;  
        do {  
            System.out.println(ii);  
        } while (ii > 0);  
    }  
}
```

1 0

2 1 2

An infinite loop

Compilation fails

What value should replace kk in the comment to cause jj = 5 to be output?

*

1/1

```

public class MyFive {
    public static void main(String[] args) {
        short kk = 11;
        short ii;
        short jj = 0;
        for (ii = kk; ii > 6; ii -= 1) {
            jj++;
        }
        System.out.println("jj=" + jj);
    }
}

```

-1
1
5
8
11

What is a static block of code in Java?

*

0/1

A block of code within a class that runs whenever the class is load on the JVM.

A block of code inside a class that runs when that class first loaded in JVM

A block of code within a class that always runs before the builder.

Correct answer

A block of code inside a class that runs when that class first loaded in JVM

What is the result?

*

0/1


```

class Feline {
    public String type = "f";

    public Feline() {
        System.out.print("feline ");
    }
}

public class Cougar extends Feline {
    public Cougar() {
        System.out.print("cougar ");
    }

    void go() {
        type = "c";
        System.out.print(this.type + super.type);
    }

    public static void main(String[] args) {
        new Cougar().go();
    }
}

```

Compilation fails
 Feline cougar c c
 Cougar c f

Feline cougar c f

Correct answer

Feline cougar c c

What is the result?

*

0/1

```
import java.util.Comparator;
import java.util.Arrays;

class MySort implements Comparator<Integer> {
    public int compare(Integer x, Integer y) {
        return y.compareTo(x);
    }
}

public class Main {
    public static void main(String[] args) {
        Integer[] primes = {2, 7, 5, 3};
        MySort ms = new MySort();
        Arrays.sort(primes, ms);
        for (Integer p2 : primes) {
            System.out.print(p2 + " ");
        }
    }
}
```

2 7 5 3

2 3 5 7

7 5 3 2

Compilation fails

Correct answer

7 5 3 2

The SINGLETON pattern allows:

*

1/1

Have a single instance of a class and this instance cannot be used by other classes.

Having a single instance of a class that can only be accessed by the first method that calls it.

Having a single instance of a class, while allowing all classes have access to that instance.

What will make this code compile and run?

*

0/1

```
public class Simple {  
    public float price;  
    public static void main(String[] args) {  
        Simple price = new Simple();  
        price =4;  
    }  
}
```

Change line 7 to the following: float price = new Simple ();

Change line 7 to the following: price.price = 4;

Change line 7 to the following: price = 4f;

Change line 7 to the following: int price = new Simple();

Change line 3 to the following: public int price;

Correct answers

Change line 3 to the following: public int price;

Change line 7 to the following: price.price = 4;

What is the result if the integer value is 33?

*

1/1

```

public static void main(String[] args) {
    int value = 33;

    if (value >= 0) {
        if (value != 0) {
            System.out.println("the ");
        } else {
            System.out.println("quick ");
        }

        if (value < 10) {
            System.out.println("brown ");
        }

        if (value > 30) {
            System.out.println("fox");
        } else if (value < 50) {
            System.out.println("jumps ");
        } else if (value < 10) {
            System.out.println("over ");
        } else {
            System.out.println("the ");
        }

        if (value > 10) {
            System.out.println("lazy ");
        } else {
            System.out.println("dog ");
        }

        System.out.println("...");
    }
}

```

The fox jump lazy...

The fox lazy...

Quick fox over lazy...

Quick fox the...

In the Java collections framework a Set Is:

*

0/1

An ordered collection that can contain duplicate elements.

An object that maps value key sets and cannot contain values Duplicates.

A collection that cannot contain duplicate elements.

Correct answer

A collection that cannot contain duplicate elements.

What is the result?

*

0/1

```
class X {
    X() {
        System.out.print(1);
    }
    X(int x) {
        this();
        System.out.print(2);
    }
}

public class Y extends X {
    Y() {
        super(6);
        System.out.print(3);
    }
    Y(int y) {
        this();
        System.out.print(4);
    }

    public static void main(String[] a) {
        new Y(5);
    }
}
```

2134

4321

2143

13

1213

134

Correct answer

1213

Which two actions, used independently, will permit this class to compile?

*

0/1

```
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();
    }
}
```

Adding throws IOException to the main() methodIOException

Adding throws IOException to the dosomething() method signature and changing the catch

Adding throws IOException to the main() method signature and to the dosomething() method

Adding throws IOException to the main() method signature

Adding throws IOException to the doSoomething() method signature.

Correct answers

Adding throws IOException to the main() method signature and to the dosomething() method

Adding throws IOException to the main() methodIOException

Which two statements are true?

*

0/1

An abstract class can be extended by an interface

An abstract class can be extended by a concrete class.

An interface can be extended by an abstract class
An interface CANNOT be extended by another interface
An abstract class CANNOT be extended by an abstract class
An abstract class can implement an interface

Correct answer

An abstract class can implement an interface
An abstract class can be extended by a concrete class.

The pom.xml file is the configuration file For:

*

0/1

Maven
Gradle

Ant

Correct answer

Maven

What is the result?

*

0/1

```

public class ScopeTest{
    int z;
    public static void main(String[] args){
        ScopeTest myScope = new ScopeTest();
        int z=6;
        System.out.print(z);
        myScope.doStuff();
        System.out.print(z);
        System.out.print(myScope.z);
    }
    void doStuff() {
        int z=5;
        doStuff2();
        System.out.print(z);
    }
    void doStuff2() {
        z=4;
    }
}

```

6565
6504
6560
6550

Correct answer

6560

A method is declared to take three arguments. A program calls this method and passes only two arguments. What is the result?

*

0/1

Compilation fails

The third argument is given the value void.

The third argument is given the value zero.

An exception occurs when the method attempts to access the third argument.
The third argument is given the appropriate false value for its declared type.
The third argument is given the value null.

Correct answer

Compilation fails

Which code fragment is illegal?

★

1/1

```
class Base3 { abstract int var1 = 89; }
```

```
class Base2 { abstract class Abs3 extends Base2 { } }
```

```
Abstract class Abs2 { void doit() { } }
```

```
Class Base1 { abstract class Abs1 { } }
```