Java Standard Edición Intermedia

Reactivos

Pregunta 1

Selecciona la API de Java que te permitirá conectarte a bases de datos.

Opciones:

a. JDBC.
b. MySQL.
c. SQL.
d. JavaScript.

Pregunta 2

¿Cuál de las siguientes opciones es un motor de bases de datos relacional?

Opciones:

a. MySQLite.
b. SQLite.
c. SQL.
d. MySQL.

Pregunta 3

Elige el código que crea una base de datos.

Opciones:

a. create PRUEBADB;
   use PRUEBADB;
b. create database PRUEBADB;
   use PRUEBADB;
c. create database PRUEBADB;
d. create database;
   use PRUEBADB;

Pregunta 4

Selecciona la opción que muestra el código que crea un usuario.

Opciones:

a. CREATE USER miUsuario IDENTIFIED BY
   GRANT usage on *.* to miUsuario@localhost identified by ' mipassword ';
   GRANT all privileges on PRUEBADB.* to miUsuario@localhost;

b. CREATE USER miUsuario IDENTIFIED BY 'mipassword';
   GRANT usage on *.* to miUsuario@localhost identified by ' mipassword ';

c. CREATE USER miUsuario IDENTIFIED BY 'mipassword';
   GRANT usage on *.* to miUsuario@localhost identified by ' mipassword ';
   GRANT all privileges on PRUEBADB.* to miUsuario@localhost;

d. CREATE USER miUsuario IDENTIFIED BY 'mipassword';
   GRANT usage on to miUsuario@localhost identified by ' mipassword ';
   GRANT all privileges on PRUEBADB.* to miUsuario@localhost;

Pregunta 5

Selecciona la clase que te permitirá registrar los errores con la base de datos.

Opciones:

a. ResultSet.
b. ResultSet executeQuery.
c. boolean Execute.
d. SQLException.

Pregunta 6

Es el proceso de ejecutar múltiples hilos (*threads*) simultáneamente.

Opciones:

a. Hilos múltiples.
b. *Multithread*.
c. Thread *pool*.
d. *Multithreading* o multihilos.

Pregunta 7

Es la función que ejecuta la acción en el hilo.

Opciones:

a. *Start*()
b. *Join*()
c. *Run*()
d. *Begin*()

Pregunta 8

¿Cuál es la palabra clave que se utiliza para extender un thread?

Opciones:

a. *Expands*.
b. *Extends*.
c. *Inherts*.
d. *Instance*.

Pregunta 9

Es el grupo de hilos de trabajo que están en espera de ser usados.

Opciones:

a. Java Thread pool.
b. Thread.
c. JDBC.
d. Pool.

Pregunta 10

¿Qué ocurre cuando dos o más tareas se ejecutan en el mismo bloque de tiempo, pero su proceso no avanza en el mismo momento?

Opciones:

a. Ocurrencia.
b. Concurrencia.
c. Segmentación.
d. Secuencia.

Pregunta 11

Selecciona el modelo en donde la concurrencia sucede cuando dos o más tareas se ejecutan en el mismo bloque de tiempo, pero su proceso no avanza en el mismo momento.

Opciones:

a. Programación orientada a objetos.
b. Programación lineal.
c. Programación funcional.
d. Programación web.

Pregunta 12

La programación orientada a objetos y la programación procedural son ___________, es decir, se enfocan en cómo hacer las cosas.

Opciones:

a. Imperativas.
b. Declarativas.
c. Declarativas.
d. Orientadas.

Pregunta 13

Una variable es un contendor en memoria, al que se asigna un valor (x = 3), esta puede cambiar a través del tiempo, reasignando valores (x = 2, después x = 8, etc.).

El concepto anterior se refiere a:

Opciones:

a. *Function* pura.
b. Funciones de primera clase.
c. *Function interface*.
d. Inmutabilidad.

Pregunta 14

```
import java.util.function.Function;
public class Tema3 {
 protected static class MiCalculo {
 public static Integer doble(Integer x) {
 return x * 2;
 }
 }
 public static void main(String[] args) {
 Function<Integer, Integer> miDoble = MiCalculo::doble;
 Integer resultado = miDoble.apply(5);
 System.out.println(resultado);
 }
}
```

El código anterior es un ejemplo de:

Opciones:

a. Function pura.
b. Function interface.
c. Function impura.
d. Function simple.

Pregunta 15

¿Cuáles son las funciones que usan dos argumentos?

Opciones:

a. Funciones Bi.
b. Funciones duples.
c. Funciones argumento.
d. Funciones dobles.

Pregunta 16

La ___________ es lo que permite a Java tener tareas trabajando simultáneamente dentro de subprocesos.

Opciones:

a. Programación orientada a objetos.
b. Programación funcional.
c. Programación síncrona.
d. Programación asíncrona.

Pregunta 17

Los procesos ________ se ejecutan de manera independiente.

Opciones:

a. Secuenciales.
b. Separados.
c. Paralelos.
d. En bloque.

Pregunta 18

Es el elemento que detiene o bloquea la ejecución del hilo principal que fue llamado hasta que el resultado de este sea recibido por parte del servidor.

Opciones:

a. *Non Blocking* API.
b. *Blocking* API.
c. API.
d. *Future* interfaz.

Pregunta 19

¿Cuál elemento representa el llamado de una función asíncrona?

Opciones:

a. *Interface* Future<V>
b. Interface Future<F>
c. Interface Future<M>
d. Interface Future<S>

Pregunta 20

Elige la opción que representa el siguiente código:

```
String name = null;
// ...
CompletableFuture<String> completableFuture
= CompletableFuture.supplyAsync(() -> {
if (name == null) {
throw new Runtime Exception("Computation error!");
}
return "Hello, " + name;
}).handle((s, t) -> s != null ? s : "Hello, Stranger!");
assertEquals("Hello, Stranger!", completableFuture.get());
```

Otro Ejemplo:

```
CompletableFuture<String> completableFuture = new CompletableFuture<>();
// ...
completableFuture.completeExceptionally(
new RuntimeException("Tarea fallo!"));
// ...
completableFuture.get(); // ExecutionException
```

Opciones:

a. Ejecución de múltiples futuros.
b. Expresión lambda.
c. Función futura.
d. Manejo de errores.

Pregunta 21

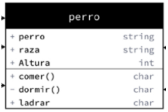
Selecciona cuáles son las primeras librerías en el desarrollo reactivo, las cuales se localizan en *frameworks* como Spring MVC, Spring Cloud y Netflix OSS.

Opciones:

a. Project Reactor y JavaR.
b. JavaR y RxJava.
c. RxJava y Project Reactor.
d. JavaRx y Project Reactor.

Pregunta 22

¿Cuál es el núcleo de RxJava?



Opciones:

a. Observables.
b. *Subscriber*.
c. Mono.
d. Flux.

Pregunta 23

¿Qué opción es la que se usa para construir aplicaciones reactivas y *non-blocking* usando Spring Framework?

Opciones:

a. JavaR.
b. Project Reactor.
c. JavaRx.
d. RxJava.

Pregunta 24

Elige qué representa el siguiente código:
Mono<String> mono = Mono.just("Hola");

Opciones:

a. Se subscribe al objeto Mono.
b. Se usa un objeto Mono.
c. Se define un objeto Mono.
d. Se crea un objeto Mono.

Pregunta 25

___________ es un *publisher* estándar que representa de 0 a N valores asíncronos.

Opciones:

a. Mono.
b. Publisher especial.
c. Project Reactor.
d. Flux.

Pregunta 26

Las clases _________ no se comprometen con un tipo específico de dato, sino hasta el momento de crear su instancia.

Opciones:

a. Generales.
b. Genéricas.
c. Locales.
d. Globales.

Pregunta 27

El término _________ se refiere a cualquier clase, interfaz o enumerador, pudiendo estar ya incluido en las librerías de Java o hecho a la medida (customizado).

Opciones:

a. *List*.
b. *Set*.
c. Tipo.
d. *Map*.

Pregunta 28

Selecciona el ejemplo que muestra una clase no genérica.

Opciones:

a. public class ArrayList.....{}
b. class ArrayList.....{}
c. public class ArrayList<E>.....{}
d. public ArrayList.....{}

Pregunta 29

Elige el código que represente una clase genérica.

Opciones:

a. public class ArrayList.....{}
b. class ArrayList.....{}
c. public ArrayList.....{}
d. public class ArrayList<E>.....{}

Pregunta 30

La letra “E” se utiliza para generar un:

Opciones:

a. Objeto *map*.
b. Parámetro en una clase genérica.
c. Número.
d. Parámetro en una clase no genérica.

Pregunta 31

Los _______ son estilos de arquitectura de software que se fundamentan en la creación de una serie de pequeños servicios.

Opciones:

a. Microservicios.
b. Macroservicios.
c. Miniservicios.
d. Servicios minimizados.

Pregunta 32

¿Cuál de las siguientes opciones es un protocolo para acceder a objetos simples?

Opciones:

a. REST.
b. API.
c. API RESTFul.
d. SOAP.

Pregunta 33

Selecciona el protocolo que significa *Representational State Transfer*.

Opciones:

a. API.
b. API RESTFul.
c. REST.
d. SOAP.

Pregunta 34

Elige cuál es el patrón de diseño que significa que, para las operaciones CRUD (*Create*, *Read*, *Update*, and *Delete*), utilizará métodos específicos HTTP, llamados verbos HTTP, para tal fin.

Opciones:

a. API.
b. API RESTFul.
c. REST.
d. SOAP.

Pregunta 35

Tecnología a la que se delega la configuración de dependencias y el despliegue de las aplicaciones elaboradas.

Opciones:

a. Spring Boot.
b. JDBC.
c. Spring shot.
d. Spring.

Pregunta 36

Los patrones de diseño _________ facilitan mecanismos para crear objetos con más flexibilidad y reutilización de código.

Opciones:

a. Estructurales.
b. De comportamiento.
c. SOLID.
d. Creacionales.

Pregunta 37

Los patrones de diseño __________ detallan cómo armar objetos y clases en estructuras mayores, con flexibilidad y eficiencia.

Opciones:

a. SOLID.
b. Estructurales.
c. Creacionales.
d. De comportamiento.

Pregunta 38

Los patrones de diseño _______ se enfocan en la comunicación y asignación de responsabilidades entre los objetos de programación.

Opciones:

a. Estructurales.
b. SOLID.
c. De comportamiento.
d. Creacionales.

Pregunta 39

¿Qué significa SOLID?

Opciones:

a. Single responsibility, Open/closed, Liskov substitution, Interface segregation, and Dependency inversion.
b. Single responsibility, Open, Liskov substitution, Interface segregation, and Dependency.
c. Single responsibility, Open/closed, Substitution, Interface segregation, and Dependency investment.
d. Single responsibility, Open/closed, Liskov substitution, Segregation, and Dependency interface.

Pregunta 40

¿Cuántos pilares de programación orientada a objetos existen?

Opciones:

a. Tres.
b. Cuatro.
c. Dos.
d. Cinco.