

# MEDRAR SOLUCIONES DE SOFTWARE EMPRESARIAL

**CARGO A DESEMPEÑAR:** Desarrollador JAVA **FECHA DE INICIACIÓN:** Abril 3 de 2020.

**CIUDAD DE CONTRATACIÓN:** Armenia, Quindío. **TIEMPO PARA EJECUCIÓN:** (12) Horas.

# PRUEBA TÉCNICA: DESARROLLO DE SOFTWARE JAVA

1. **Preguntas conceptuales OOP**
   1. ¿Qué significa “alta cohesión bajo acoplamiento” en OOP?

Significa que cada clase debe estar diseñada con un proposito y no debe conocer o depender de la implementación de las clases que utiliza.

* 1. ¿En qué caso favorecería el uso de la herencia sobre la composición en OOP?

Cuando la respuesta a es un sea afirmativa y la respuesta a tiene un sea negativa

* 1. ¿Por qué en OOP la encapsulación es importante?

Por que nos permite controlar el estado de un objeto exclusivamente desde su interior.

* 1. ¿Qué implicaciones trae en el diseño de una aplicación que respete el principio “cerrado a la modificación abierto a la extensión”?

Se obtiene una aplicación en que los cambios no afectaran la funcionalidad anterior de la aplicación y que facilitará la implementación de los cambios. En general se tiene una aplicaciónmas fácil de mantener.

* 1. ¿Qué significa la palabra clave static, y dónde puede ser usada?

La palabra static es una palabra reservada del lenjuage java, se puede utilizar a nivel de propiedad de la clase, y a nivel de método de la clase. En general indican propiedades o métodos que pueden ser utilizados sin una instancia de la clase.

* 1. ¿Qué es polimorfismo? Descríbalo con un ejemplo

Polimorfismo es una característica que permite cambiar el comportamieto de la aplicación de forma dinámica en tiempo de ejecución. Por ejemplo el área de una figura geométrica se calcula de diferentes formas dependiendo de la figura, aplicando polimorfismo es posible que se utilice cada una de las diferentes formas de calcular el área sin conocer de antemano la figura.

Esto se logra utilizando una interfaz común en todas las figuras y usando la interfaz para calcular el área en lugar de las figuras directamente.

* 1. ¿Qué es el garbage collector?

Es una aplicación dentro de la máquina virtual de java que se encarga de detectar y destruir si es necesario los objetos que ya no son alcanzables desde ningún programa en ejecución

* 1. ¿Qué hace la palabra clave synchronized?

Obtiene el Lock de un objeto, método o parte de código de forma que solo pueda ser accesado por un hilo de forma simultanea

* 1. ¿Cuándo y por qué son los getters y setters importantes?

Siempre son importantes por que nos brindan la posibilidad de tener enapsulamiento.

* 1. ¿Cuáles son las diferencias entre interfaces, clases abstractas, clases e instancias?

Antes de java 8 las interfaces se diferenciaban de las clases abstractas en que las interfaces no podian implementar métodos, a partir de java 8 esta diferencia a desaparecido.

En las interfaces las propiedades son por defecto public static final, en la clase abstracta podemos tener cualquier modificador para las propiedades.

Las interfaces pueden tener el modificador de acceso default las clases abstractas no.

* 1. ¿Qué es sobrecarga de métodos?

Es una característica del lenguaje que permite usar el mismo nombre en dos métodos diferentes siempre y cuando tengan diferente numero deparámetros y/o diferente tipo de parámetros.

* 1. ¿Cómo se maneja o controla una excepción?

Utilizando un bloque try, catch, finally.

# Preguntas de algoritmos y estructuras de datos

1. ¿Qué diferencias existen entre las estructuras de datos: lista enlazada (linked list), un arreglo y un tabla de hash (hastable/hashmap)?

|  |  |  |
| --- | --- | --- |
| Lista enlazada | arreglo | hashmap |
| Collección de objetos donde un objeto tiene referencia al siguiente de la lista, pueden estar colocados en ubicaiones no sucesivas de memoria | Colección de objetos o primitivos que están colocados secuencialmente en cuanto a posiciones de menoria | Colección de objetos organizados como un diccionario de clave valor |
| Solo soporta objetos | Soporta objetos y primitivos | Solo soporta objetos |
| Tiempo de acceso variable depende del número de elementos | Tiempo de acceso constante | Tiempo de acceso variable depende de la calidad de la implementación del hash de la clave |

1. ¿Bajo qué criterio una función recursiva es más costosa que una función iterativa?

Bajo el criterio de la eficiencia.

1. ¿Con qué estructura de datos modelaría el sistema de transporte Transmilenio para realizar cálculos sobre las rutas y tiempos totales de cada ruta y encontrar la ruta más rápida de estación a estación?

Con grafos

1. ¿Con qué tipo de algoritmo se enfrentaría al problema de distribución de paquetes por peso en camiones de una empresa de mensajería para que se realizara de manera óptima?

Algoritmos de selección optima

# Ingeniería de software

1. Liste las etapas básicas de la construcción de software, independientes a la metodología de desarrollo y defina qué es calidad de software.

Requerimiento, diseño, codificación, implementación, verificación, mantenimiento.

1. Describa los escenarios de prueba que le definiría al proceso de hervir un huevo

existe huevo, existe olla, hay agua, existe fuente de calor, fuente de calor encendida, olla llenada con agua, huevo en olla con agua, olla sobre la fuente de calor, tiempo de cocción cumplido, fuente de calor apagada.

1. ¿Cuáles la **diferencia** entre concurrencia, disponibilidad y consistencia?

Concurrencia: cosas que ocurren o deberian ocurrir al mismo tiempo

Disponibilidad: Relación del tiempo en que el sistema es funcional con respecto al tiempo esperado.

Consistencia: La concordancia entre lo que se necesita y lo que se entrega

1. Mencione algunos patrones de diseño:

Singleton, decorador, abstract factory, facade, observer.

1. Explique la inyección de dependencia y su relación con la inversión de control.

Inyección de dependencia es un proceso que permite manejar la instancia de objetos que se necesitan para el correcto funcionamiento de la aplicación, y se relaciona con la inversión de control con el hecho de que permite utilizar la programación a traves de interfaces para codificar un sistema sin tener que conocer los detalles de la implementación de subsistemas de los cuales depende.

1. ¿Cuáles la diferencia entre un bloqueo optimista y un bloqueo pesimista?

Bloqueo optimista supone que generalmente no se presentan intentos de modificación concurrente por lo que la base de datos por ejemplo no bloquea los registros que se han consultado, mientras que el bloqueo pesismista supone que toda consulta lleva a una modificación por lo que bloquea todo registro consultado.

# Problemas prácticos

1. Resuelva de manera recursiva los siguientes ejercicios:
   * Encontrar el número mayor en un arreglo de 12 posiciones llenado de manera aleatoria
   * Sumar los dígitos de un número de 6 caracteres dado de forma aleatoria
   * Encontrar el factorial de un número entre 1 y 10 dado de manera aleatoria

NOTA: para estos 3 ejercicios se debe imprimir en consola los datos de entrada y salida.

1. Se pide realizar un desarrollo para un cine (solo de una sala) la cual tiene un conjunto de asientos (8 filas por 9 columnas).

Del cine nos interesa conocer la película que se está reproduciendo, cuantas sillas están ocupadas, cuantas disponibles y el precio de la entrada en el cine.

De las películas nos interesa saber el título, duración, edad mínima y director.

Del espectador, nos interesa saber su nombre, edad y el dinero que tiene.

Los asientos son etiquetados por una letra (columna) y un número (fila), la fila 1 empieza al final de la matriz como se muestra en la tabla. También deberemos saber si está ocupado o no el asiento.

8 A 8 B 8 C 8 D 8 E 8 F 8 G 8 H 8 I

7 A 7 B 7 C 7 D 7 E 7 F 7 G 7 H 7 I

6 A 6 B 6 C 6 D 6 E 6 F 6 G 6 H 6 I

5 A 5 B 5 C 5 D 5 E 5 F 5 G 5 H 5 I

4 A 4 B 4 C 4 D 4 E 4 F 4 G 4 H 4 I

3 A 3 B 3 C 3 D 3 E 3 F 3 G 3 H 3 I

2 A 2 B 2 C 2 D 2 E 2 F 2 G 2 H 2 I

1 A 1 B 1 C 1 D 1 E 1 F 1 G 1 H 1 I

Se debe realizar una pequeña simulación, en el que se generaran muchos espectadores (más que la disponibilidad de la sala) y se sentaran aleatoriamente (verificar si la silla está ocupada o no).

Solo se podrá sentar si tienen el suficiente dinero, hay espacio libre y tiene edad para ver la película.

Los datos del espectador y la película pueden ser totalmente aleatorios.

NOTA: Se deben imprimir de manera ordenada los datos de todos los espectadores y películas generadas antes de pasar al proceso de selección de película (aleatoria de las generadas) y a sentar y validar los espectadores, esto al final debe ser impreso adicional a la información solicitada inicialmente.

Para las pruebas prácticas se debe entrar los proyectos y/o clases java para su ejecución y pruebas.

Para enviar los entregables de la prueba se debe tener o crear un repositorio en git público en el cual se deben subir el documento resuelto y los proyectos java para ser revisados y ejecutados. Una vez tengan este proyecto en el repositorio, se debe enviar la URL del repositorio al correo [talentos@medrarsof.com,](mailto:talentos@medrarsof.com) para poder clonarlo y hacer su revisión.

* La tecnología para realizar el aplicativo podrá ser elegida por el aspirante.

Página **4** de **4**