

Los tiempos dispuestos en cada apartado equivalen a un estimado de duración de cada test basado en su nivel de dificultad, no obstante contarás con de 3 días para completar la prueba.

TEST BÁSICO DE CONOCIMIENTOS EN PROGRAMACIÓN ORIENTADA A OBJETOS (JAVA)

20 Minutos



El siguiente test es una herramienta de ayuda para comprobar tu familiaridad con algunos conceptos de la POO y el lenguaje Java.

1- ¿Cuál es la descripción que crees que define mejor el concepto 'clase' en la programación orientada a objetos?

- a) Es un concepto similar al de **array**
- b) Es un tipo particular de variable
- c) Es un modelo o plantilla a partir del cual creamos objetos
- d) Es una categoría de datos ordenada secuencialmente

2- ¿Qué elementos cree que definen un objeto?

- a) Sus cardinalidad y su tipo
- b) Sus atributos y métodos
- c) La forma en que establece comunicación e intercambia mensajes
- d) Su interfaz y los eventos asociados

3- ¿Cuál de las siguientes sentencias tiene que ver con la herencia?

- a) **Public class** Component **extends** Producto
- b) **Public class** Component **inherit** Producto
- c) **Public class** Component **implements** Producto
- d) **Public class** Component **belong to** Producto

4- ¿Qué es el bytecode en Java?

- a) El formato de intercambio de datos
- b) El formato que obtenemos tras compilar un fuente .java
- c) Un tipo de variable
- d) Un depurador de código

5- ¿Qué código asociarías a una Interfaz en Java?

- a) **public class** Componente **interface** Product
- b) Componente cp = **new** Componente (interfaz)
- c) **public class** Componente **implements** Printable
- d) Componente cp = **new** Componente.interfaz

6- ¿Qué significa la sobrecarga (overload) de un método?

- a) Editarlo para modificar su comportamiento
- b) Cambiarle el nombre dejándolo con la misma funcionalidad
- c) Crear un método con el mismo nombre pero diferentes argumentos
- d) Añadir funcionalidades a un método

TEST BÁSICO EN RESOLUCIÓN DE PROBLEMAS Y ALGORITMOS

2 Horas



El siguiente test es una herramienta de ayuda para comprobar tus habilidades lógicas y desarrollo de algoritmos.

7- En el espacio marcado con __ hace falta una pieza de código necesaria para indicarle al algoritmo cuando la variable int0 es divisible por 5. Marca la opción correcta.

boolean isDivisibleBy5 = __

- a) **Math.isDivisible**(int0, 5);
- b) int0 % 5 != 5
- c) int0 / 5 ? **true** : **false**
- d) int0 % 5 ==0

8- ¿Cuántas veces el código imprime "Hola Mundo!" ?.

```
class Main {  
    public static void main(String args){  
        for(int i=0; i<10; i++){  
            i+=1;  
            System.out.println("Hello World!");  
        }  
    }  
}
```

- a) 10 veces
- b) 5 veces

- c) 9 veces
- d) Un número infinito de veces.

9- Ordenamiento de vectores:

- a) Se debe crear un vector de 100 posiciones y cargarlo dinámicamente con valores entre 1 y 100000.
- b) Se debe ordenar el vector de mayor a menor.
- c) Imprimir el resultado en pantalla.

10- Operaciones Aritméticas: Se debe crear una función que permita elevar un número A, a otro número B solo usando sumas.

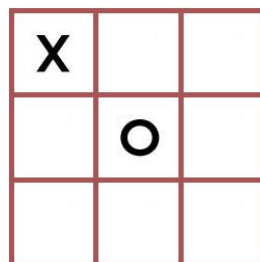
TEST DE PROGRAMACIÓN

1 Día



El siguiente test es una herramienta de ayuda para comprobar tus habilidades lógicas y desarrollo de algoritmos. Para este reto, se le solicita a el candidato desarrollar una aplicación web utilizando las tecnologías citadas en el apartado requerimientos.

N en Línea



Reglas de N en Línea

Hay dos jugadores; uno juega con el símbolo X y el otro juega con el símbolo O. Ambos se mueven alternativamente y el jugador X empieza. Pueden colocar sus símbolos solamente en cuadrados vacíos. Inicialmente, el tablero consiste de n casillas vacías, alineadas en un tablero n^2 . El objetivo del juego es colocar n símbolos propios en una columna, fila o diagonal. Si el tablero es relleno pero ninguno de los jugadores consiguió este objetivo, el partido acaba en empate.



La aplicación debe permitir crear una partida de N en línea, para crear la partida se le debe pedir al usuario ingresar el nombre de los jugadores 1 y 2, adicionalmente el usuario debe poder ingresar el tamaño de la matriz de juego n. Al terminar la partida la aplicación deberá guardar el estado final del juego indicando que jugador se quedó con la victoria. La aplicación, debe mostrar en un histórico de las partidas que se han jugado organizado por fecha de último juego.

REQUERIMIENTOS:

1. Uso de Java
2. Uso de Eclipse u otro IDE que permita compilar JAVA. <https://www.eclipse.org/>
3. Uso del framework ZK <https://www.zkoss.org/> , <https://www.zkoss.org/zkdemo>
4. Uso de Ant (incluido dentro de Eclipse) para la compilación de aplicaciones y la creación del archivo WAR que se desplegará en el contenedor de aplicaciones.
<https://www.vogella.com/tutorials/ApacheAnt/article.html>,
<https://rohitprabhakar.com/2010/02/03/how-to-run-ant-build-from-eclipse/>
5. Uso de Tomcat como servidor de aplicaciones <https://tomcat.apache.org/download-90.cgi>
6. Uso de H2 como motor de base de datos. <http://www.h2database.com/html/main.html>

ENTREGABLES:

- Correo: andres.rahn@starttodoit.com
- Asunto: Solución Prueba Desarrollador de Software 2021

En el cuerpo del mensaje incluir:

- Nombre, Correo electrónico de contacto y número telefónico.
- a) El Url al Repositorio Git de la aplicación Web y los ejercicios, incluido el formulario que se está remitiendo.
- b) La aplicación debe contener su base de datos, y todas las instrucciones para ponerla en funcionamiento o enviar la Url de la aplicación Web para acceder.
- c) Credenciales del usuario para probar la aplicación.
- d) Cada punto de algoritmia básica debe tener sus parámetros de ejecución y las debidas impresiones en pantalla para probar las aplicaciones.

¡MUCHOS ÉXITOS! . No se revisará ningún código cargado una vez terminado el plazo.