

Bienvenido a tu Certificación

Ya estás un paso más cerca de ser un Analista del Conocimiento - Dimensión Programador!

El examen consta de 5 ejercicios basados en los conocimientos exigidos por el 111 Mil y por la industria, para convertirte en programador junior. Tené en cuenta y leé con mucha atención las siguientes pautas para la correcta realización y aprobación del examen:

- 1- El examen tiene una duración máxima de 2 (dos) horas. Deberás enviar los resultados antes de cumplir ese tiempo.
- 2- Es necesario responder TODOS los ejercicios para poder aprobar el examen.
- 3- El resultado es APROBADO o DESAPROBADO, no tiene puntaje.
- 4- Los veedores estarán presentes para verificar que el examen se tome en las condiciones adecuadas.
- 5- Está prohibido utilizar el chat, el correo electrónico (fuera del uso normal para abrir este formulario), o cualquier página web que no sea este Google Form.
- 6- Tené presente que el teclado no reconoce la tecla "TAB" para escribir código Java, por lo que deberás usar 3 (tres) veces la tecla "ESPACIO" para poner sangrías (indentar/tabular) al alinear el código.
- 7- En caso de no encontrar el tipeo de una tecla, podrás usar el mapa de caracteres, que podés encontrarlo en: "Tecla Windows + R" y en el cuadro de texto escribir "charmap", y apretar "Enter". También podés encontrarlo en "Inicio--> Accesorios ---> Herramientas del Sistema ---> Mapa de caracteres".
- 8- Cuando en el punto siguiente el formulario te pida la dirección de correo electrónico, tenés que declarar el que usaste para inscribirte a la certificación.

En los próximos días recibirás un correo electrónico con los resultados del examen.

Te deseamos mucha suerte y a trabajar en los ejercicios!

***Obligatorio**

1. Dirección de correo electrónico *

2. Apellidos *

3. Nombres *

4. DNI (con puntos) *

5. Fecha de nacimiento *

Ejemplo: 15 de diciembre de 2012

6. Sede de cursada/libre *

Marca solo un óvalo.

☐ SIN CURSADA - LIBRE

☐ Otros: _____

7. Sede de Certificación *

Marca solo un óvalo.

☐ E.E.T. N°3 - Necochea

☐ Club de Emprendedores de Pilar

☐ PUNTO DIGITAL LA PLATA

☐ Ministerio de Producción - CABA

☐ CFP N° 23 - ALMAGRO

☐ Universidad de San Juan

☐ Universidad Nacional de San Luis

☐ Universidad Nacional de Córdoba

☐ UTN - Facultad Regional Rosario

☐ Santa Fe - Universidad Nacional del Litoral

☐ Club de Emprendedores San Justo - Santa Fe

☐ Misiones - CFP N° 16 Posadas

☐ UTN - Facultad Regional Resistencia (Departamento de Ingeniería en Sistemas de Información)

☐ UTN - San Miguel de Tucumán

☐ Universidad Nacional de Tucumán

☐ Universidad Nacional de Catamarca - IDI

☐ SALTA - E.E.T. N° 3100

☐ ORAN - E.E.T. N° 3104 "Lanza Colombres"

☐ GRAL. MOSCONI - E.E.T. N° 3113 "Francisco Tobar"

☐ MENDOZA - Campus Tecnología de la Información y la Comunicación (TIC) del ITU

Contexto - Control de Stock

En el comando central de 111Mil se concentran todos los elementos necesarios para el dictado y promoción de los cursos, incluyendo las computadoras con las que se realizan las actividades prácticas. Debido a la cantidad de computadoras, el encargado del almacenamiento solicita el desarrollo de un sistema de control de stock.

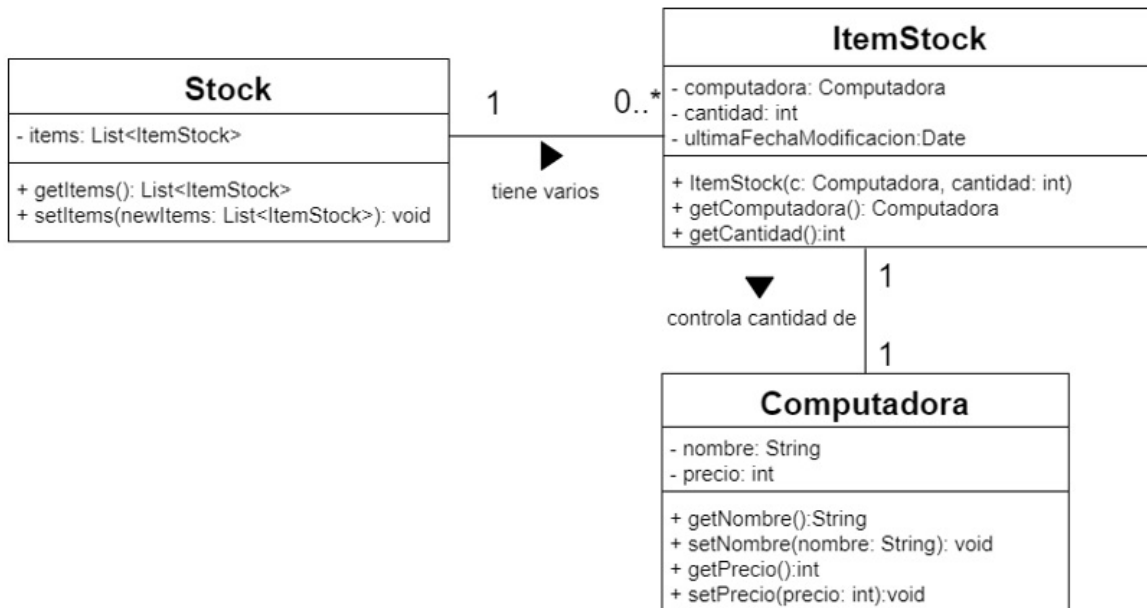
De acuerdo a lo comentado por el encargado, las computadoras almacenadas tienen un nombre, por ejemplo: laptop, all-in-one, pc en miniatura, etc; y un precio asociado. Para cada Computadora, se debe registrar la cantidad existente en el almacén.

Al encargado le interesa, por sobre todas las cosas, conocer aquellos ítems que se encuentran por debajo de cierto límite (para realizar reposiciones), y conocer qué computadora es la más cara (para tomar medidas de seguridad). El equipo de desarrollo, del cual Ud. es parte, acepta el desafío y se embarca en la creación del sistema de control de stock, comenzando por el diagrama de clases.

Ejercicio 1 - Implementar desde el diagrama de clases

Parte del equipo de desarrollo ya comenzó con la implementación del sistema de control de stock y se nos pide completar el código creado hasta el momento. A partir del diagrama de clases y el código dado, complete:

- La declaración de las variables: computadora y cantidad en la clase ItemStock, nombre en la clase Computadora.
- El método getItems de la clase Stock, el cual retorna la lista de ítems.
- El método setItems de la clase Stock, el cual asigna a la variable items la lista pasada como parámetros.



```

public class ItemStock {
    // A: Agregar variables computadora y cantidad

    private Date ultimaFechaModificacion;

    public ItemStock(Computadora c, int cantidad){
        this.computadora = c;
        this.cantidad = cantidad;
    }

    public int getCantidad(){ return this.cantidad; }
    public Computadora getComputadora(){ return this.computadora; }
}

public class Computadora {
    // A: Agregar la variable nombre

    private int precio;

    public void setNombre(String nombre){ this.nombre = nombre; }
    public String getNombre(){ return this.nombre; }

    public void setPrecio(int precio){ this.precio = precio; }
    public int getPrecio(){ return this.precio; }
}

public class Stock {
    private List<ItemStock> items = new ArrayList<ItemStock>();

    // B: Agregar método getItems, retornando una copia de items.

    // C: Agregar método setItems, copiando los elementos en newItems
}

```

8. A. La declaración de las variables: computadora y cantidad en la clase ItemStock, nombre en la clase Computadora.

9. B. El método `getItems` de la clase `Stock`, el cual retorna la lista de ítems.

10. C. El método `setItems` de la clase `Stock`, el cual asigna a la variable `items` la lista pasada como parámetros.

Ejercicio 2 - Implementar un método a partir de un enunciado

Para satisfacer uno de los requerimientos del encargado, se pide crear en la clase `Stock` el método `consultarItemsFaltantes`, con la siguiente signatura:

```
public List<ItemStock> consultarItemsFaltantes(int cantidadMáxima) {  
}
```

11.

El mismo debe retornar una **nueva lista de ítems**, pero sólo incluyendo aquellos **ItemStock** con **cantidad menor al parámetro "cantidadMáxima"**.

Afortunadamente, el equipo nos ha dejado la siguiente lista de sugerencias para implementar el método:

- Puede crear una lista auxiliar utilizando el siguiente código:
`List<ItemStock> resultado = new ArrayList<ItemStock>()`
- Debe agregar cada ItemStock (de la lista "items") al resultado solo si cumplen la condición dada.
- Recuerde, ¡no retorne del método sin antes haber recorrido la totalidad de la lista!

Ejercicio 3 - Identificar error en código.

12. Un colega nos acerca una porción de código que no hace lo que debería hacer y nos solicita ayuda. El código en cuestión es un pequeño algoritmo de la clase Stock que busca la Computadora más costosa, de manera que el encargado del almacén pueda guardarla en un lugar seguro.

```
public Computadora obtenerComputadoraMasCostosa() {  
    Computadora masCostosa = null;  
    for (ItemStock item : this.items) {  
        if (masCostosa == null  
            || item.getComputadora().getPrecio() <= masCostosa.getPrecio())  
            masCostosa = item.getComputadora();  
    }  
    return masCostosa;  
}
```

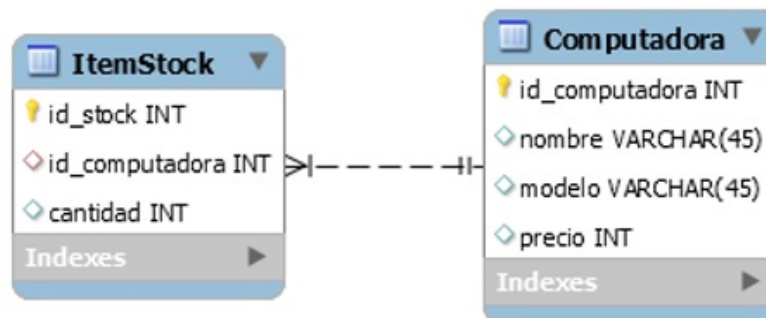
Por suerte, nuestro colega ya identificó posibles errores y solo debemos indicar cual de ellos (**solo uno**) es el que genera el problema.

Marca solo un óvalo.

- ☐ 1. La variable "masCostosa" debería estar inicializada con una Computadora con precio 0.
- ☐ 2. Debería utilizar un while en lugar de un for.
- ☐ 3. La comparación entre el precio de cada ítem y el de la variable "masCostosa" debería ser mayor estricto (>) o mayor o igual (>=).
- ☐ 4. El retorno de la función debería ser "this.items".

Ejercicio 4 - Interpretación de DER.

A partir del DER del sistema, califique como Verdadero (V) o Falso (F) las siguientes afirmaciones (se tiene que poner Verdadero (V) o Falso (F) en cada afirmación. Puede haber más de un Verdadero (V) y más de un Falso (F)):



13. Marca solo un óvalo por fila.

	VERDADERO	FALSO
A- En ItemStock, la tabla tiene una clave compuesta	<input type="radio"/>	<input type="radio"/>
B- Cada ItemStock tiene, como máximo, una sola computadora asociada	<input type="radio"/>	<input type="radio"/>
C- El valor 200 es un valor válido para la columna "cantidad" de ItemStock	<input type="radio"/>	<input type="radio"/>
D- La relación entre ItemStock y Computadora es N a 1	<input type="radio"/>	<input type="radio"/>
E- La clave principal de Computadora es el id_computadora	<input type="radio"/>	<input type="radio"/>
F- id_computadora es una clave foránea de ItemStock	<input type="radio"/>	<input type="radio"/>
G- La relación entre ItemStock y Computadora es 1 a N	<input type="radio"/>	<input type="radio"/>
H- "159 pesos" es un valor válido para el precio de Computadora	<input type="radio"/>	<input type="radio"/>

Ejercicio 5 - Consultas SQL.

El encargado del depósito desea averiguar la cantidad de unidades disponibles de aquellas computadoras cuyo precio que supera los 50000 pesos. Los datos que necesita el encargado son:

- Nombre de la computadora
- Cantidad en el stock
- Precio de la computadora

Basándose en el DER del ejercicio 4, realice una consulta SQL de acuerdo a lo solicitado por el encargado.

14.

Te recordamos revisar tu examen antes de presionar el botón "ENVIAR", ya que una vez que lo hagas, tus respuestas serán enviadas y no podrás modificarlas.

Éxitos!