

Bienvenido a tu Certificación - Jueves 12/04/2018 Turno Noche 19 hs.

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. Sede de cursada/libre *

Marca solo un óvalo.

- ☐ SIN CURSADA - LIBRE
- ☐ Otros: _____

6. Sede de Certificación *

Marca solo un óvalo.

- ☐ Cámara Empresaria de Tandil
- ☐ CENTRO DE INNOVACION E INVESTIGACION PARA EL DESA EDUCATIVO, PROD Y TEC
Dr. NESTOR KIRCHNER
- ☐ Universidad Nacional de San Juan
- ☐ EEST N° 5
- ☐ Ministerio de Producción
- ☐ PUNTO DIGITAL LA PLATA
- ☐ PUNTO DIGITAL LANUS
- ☐ Terrazas de Portezuelo
- ☐ Universidad Abierta Interamericana
- ☐ Universidad CAECE
- ☐ Universidad de Belgrano
- ☐ Universidad Nacional de Córdoba
- ☐ Universidad Nacional de Entre Rios
- ☐ Universidad Nacional de la Rioja
- ☐ Universidad Nacional del Nordeste
- ☐ Universidad Nacional Misiones
- ☐ Universidad Tecnológica Nacional - REGIONAL MENDOZA
- ☐ UTN - Facultad Regional Rosario
- ☐ UTN - Regional Resistencia (Departamento de Ingeniería en Sistemas de Información)
- ☐ UNIVERSIDAD NACIONAL DEL LITORAL
- ☐ CFP N° 23
- ☐ IFTS2
- ☐ UNIVERSIDAD FASTA

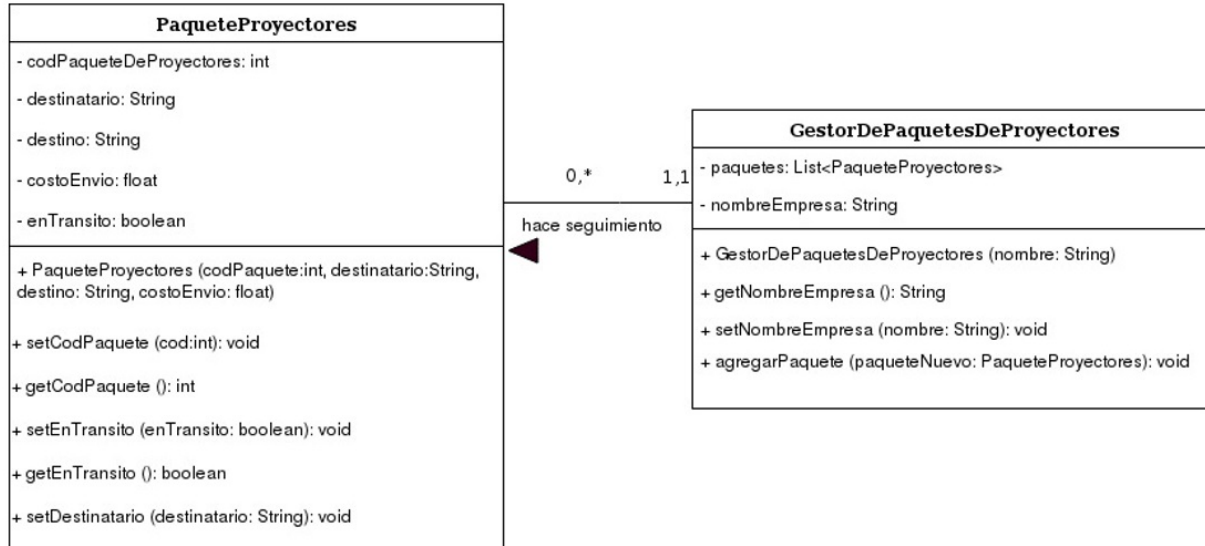
Contexto - Envío de Proyectoras

El programa 111Mil necesita enviar a sus alumnos proyectores para poder desarrollar las actividades del programa. Como existen sedes en todo el país es necesario tener un seguimiento de los paquetes con proyectores que se envían para saber si efectivamente llegaron a las sedes correspondientes. Para tener este control se le ha pedido a los alumnos del 111Mil que comiencen a implementar un sistema que realice el seguimiento de los paquetes.

Ejercicio 1 - Implementar desde el diagrama de clases

Dado que es necesario comenzar a modelar el sistema de seguimiento de los paquetes de proyectores se ha creado un diagrama de clases de UML para indicar las necesidades del sistema. Adicionalmente, se ha comenzado con la implementación pero nos piden que completemos el código para:

- Declarar las variables `codPaqueteDeProyectores`, `costoEnvio`, `enTransito` de la clase `PaqueteProyectores`.
- Implemente los métodos `getEnTransito`, `setEnTransito`, `getCodPaquete` y `setCodPaquete` de la clase `PaqueteProyectores`.
- Implemente el método `agregarPaquete` de la clase `GestorDePaquetesDeProyectores`. El mismo deberá agregar un nuevo paquete en la lista paquetes.



```
public class PaqueteProyectores {
```

```
    private String destinatario;
    private String destino;
```

```
    //a. DECLARAR LAS VARIABLES codPaqueteDeProyectores, costoEnvio, enTransito
```

```
    public PaqueteProyectores(int codPaqueteDeProyectores, String destinatario, String destino, float
costoEnvio) {
```

```
        this.codPaqueteDeProyectores = codPaqueteDeProyectores;
        this.destinatario = destinatario;
        this.destino = destino;
        this.costoEnvio = costoEnvio;
        this.enTransito = true;
```

```
    }
```

```
    public void setDestinatario (String destinatario) {
        this.destinatario = destinatario;
    }
```

```
    public String getDestinatario () {
        return this.destinatario;
    }
```

```
    public void setDestino (String destino) {
        this.destino = destino;
    }
```

```
    public String getDestino () {
        return this.destino;
    }
```

```

    }

    public void setCostoEnvio (float costoEnvio) {
        this.costoEnvio = costoEnvio;
    }

    public float getCostoEnvio () {
        return this.costoEnvio;
    }

    //b. IMPLEMENTAR LOS METODOS getEnTransito, setEnTransito, getCodPaquete y
    setCodPaquete
    }

    public class GestorDePaquetesDeProyectores {
        private List<PaqueteProyectores> paquetes;
        private String nombreEmpresa;

        public GestorDePaquetesDeProyectores (String nombreEmpresa) {
            this.paquetes = new ArrayList<>();
            this.nombreEmpresa = nombreEmpresa;
        }

        public void setNombreEmpresa(String nombreEmpresa) {
            this.nombreEmpresa = nombreEmpresa;
        }

        public String getNombreEmpresa(){
            return this.nombreEmpresa;
        }

        public void setNombreEmpresa(String nombreEmpresa) {
            this.nombreEmpresa = nombreEmpresa;
        }

        public String getNombreEmpresa(){
            return this.nombreEmpresa;
        }

        //c. IMPLEMENTAR EL METODO  agregarPaquete
        de acuerdo a la especificación del Diagrama de Clases

    }

```

7. a) Declarar las variables **codPaqueteDeProyectores**, **costoEnvio**, **enTransito** de la clase **PaqueteProyectores**.

8. b) Implemente los métodos `getEnTransito`, `setEnTransito`, `getCodPaquete` y `setCodPaquete` de la clase `PaqueteProyectores`.

9. c) Implemente el método `agregarPaquete` de la clase `GestorDePaquetesDeProyectores`. El mismo deberá agregar un nuevo paquete en la lista `paquetes`.

Ejercicio 2 - Implementar un método a partir de un enunciado

Programar en Java el código del método `buscarPaqueteDeProyectores` en la clase `GestorDePaqueteDeProyectores`. El mismo recibirá por parámetro una variable llamada `codPaq` de tipo `int` y retornará el paquete correspondiente a ese código. En caso de no encontrarlo deberá retornar `null`.

10.

Ejercicio 3 - Interpretación de código

11. Un desarrollador implementó el siguiente método en la clase `GestorDePaqueteDeProyectores` pero no usó nombres representativos. Indique cuál de las siguientes opciones explica lo que hace el código:

```
public int xxxx() {  
    int cant = 0;  
    for (PaqueteProyectores paq : paquetes) {  
        if (paq.getEnTransito()) {  
            cant = cant + 1;  
        }  
    }  
    return cant;  
}
```

Marca solo un óvalo.

- ☐ A. Calcula el total de paquetes de proyectores que se están administrando
- ☐ B. Calcula la cantidad de paquetes de proyectores en tránsito
- ☐ C. Calcula el promedio del precio de los paquetes de proyectores en la empresa
- ☐ D. Ninguna de las anteriores.

Ejercicio 4 - Interpretación de Diagrama de Entidades y Relaciones (DER).

A partir del DER, responda Verdadero (V) o Falso (F) a las siguientes afirmaciones (asignar a cada afirmación un valor V o F. Todas las afirmaciones tienen que tener un valor asignado. Puede haber varios V y varios F):



12. Marca solo un óvalo por fila.

	VERDADERO	FALSO
A- Entre las tablas PaqueteProyectores y GestorDePaquetesDeProyectores existe una relación 1 a 1	<input type="radio"/>	<input type="radio"/>
B- Entre las tablas GestorDePaquetesDeProyectores y PaqueteProyectores existe una relación 1 a N	<input type="radio"/>	<input type="radio"/>
C- Entre las tablas PaqueteProyectores y GestorDePaquetesDeProyectores existe una relación N a N	<input type="radio"/>	<input type="radio"/>
D- La clave primaria de la tabla PaqueteProyectores es GestorDePaqueteDeProyectores_idGestorDePaqueteDeProyectores	<input type="radio"/>	<input type="radio"/>
E- La clave primaria de la tabla GestorDePaqueteDeProyectores es idGestorDePaquetes	<input type="radio"/>	<input type="radio"/>
F- La tabla GestorDePaqueteDeProyectores no tiene clave foránea	<input type="radio"/>	<input type="radio"/>
G- La tabla PaqueteProyectores no tiene clave foránea	<input type="radio"/>	<input type="radio"/>

Ejercicio 5 - Consultas SQL.

13. Dado el diagrama de entidad-relación presentado en el ejercicio anterior, escriba una consulta SQL que liste todos los paquetes de proyectores cuya ciudad destinataria sea "Tucumán" y tenga un costo de envío mayor a 300\$ y el nombre de la empresa correspondiente.

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!