Taller de repaso - Aplicación de contenidos
POO, UML y Java

A continuación se plantea un set de ejercicios con el fin de repasar y aplicar los conceptos revisados hasta ahora sobre POO, UML y Java.

Dependiendo del caso, deberá Ud. y el grupo asignado, desarrollar y resolver cada uno de los 3 desafíos propuestos. Pero no sólo la respuesta final es la relevante, sino saber que hizo/aportó/opinó c/u de los estudiantes en las respuestas.

Pueden dividirse la tarea, donde c/u hace un desafío o todos los estudiantes colaboran en todos los desafíos.

Los resultados deberán ser entregados en un único reporte en formato PDF hasta el viernes 21/08/20 a las 18hrs en el link que se indica en campus virtual.

El reporte debe contener:
- Portada (indicando al menos la tarea, integrantes y curso)
- Desarrollo (proceso de resolución, que hizo c/u y las respuestas a cada desafío)
- Comentarios finales/conclusiones

Desafíos

1.) Analizando un código fuente implementado en Java.

Considere un nuevo e interesante servicio de música en formato streaming llamado Spotifly…*para no arriesgar demandas ;-)*
Spotifly es un servicio que da acceso a miles de canciones provenientes de todo el mundo y administrar nuestras propias listas de reproducción. Este servicio entrega las opciones de agregar canciones al repositorio y crear listas de reproducción con un conjunto de canciones presentes en el repositorio.
El desarrollador que inició el proyecto se ha ido en busca de nuevos desafíos laborales, pero no dejó la documentación ni el diseño del proyecto. Sólo tenemos un poco del código fuente.

A partir del siguiente código fuente, se le pide:
- construir un diagrama de clases UML

```
public class Spotifly {
    private int totalCanciones;
    private ArrayList<Cancion> canciones = new ArrayList<Cancion>();
    private ArrayList<ListaReproduccion> listasReproduccion = new ArrayList<ListaReproduccion>();
```

```java
public class ListaReproduccion {
    private String nombre;
    private int duracionLista;
    private int cantidadCanciones;
    private ArrayList<Cancion> canciones = new ArrayList<Cancion>();
```

```java
public class Cancion {
    private String artista;
    private int año;
    private String duracion;
    private String titulo;
    private Genero genero;
```

Genero puede ser una clase como tal o ser un ***enum***

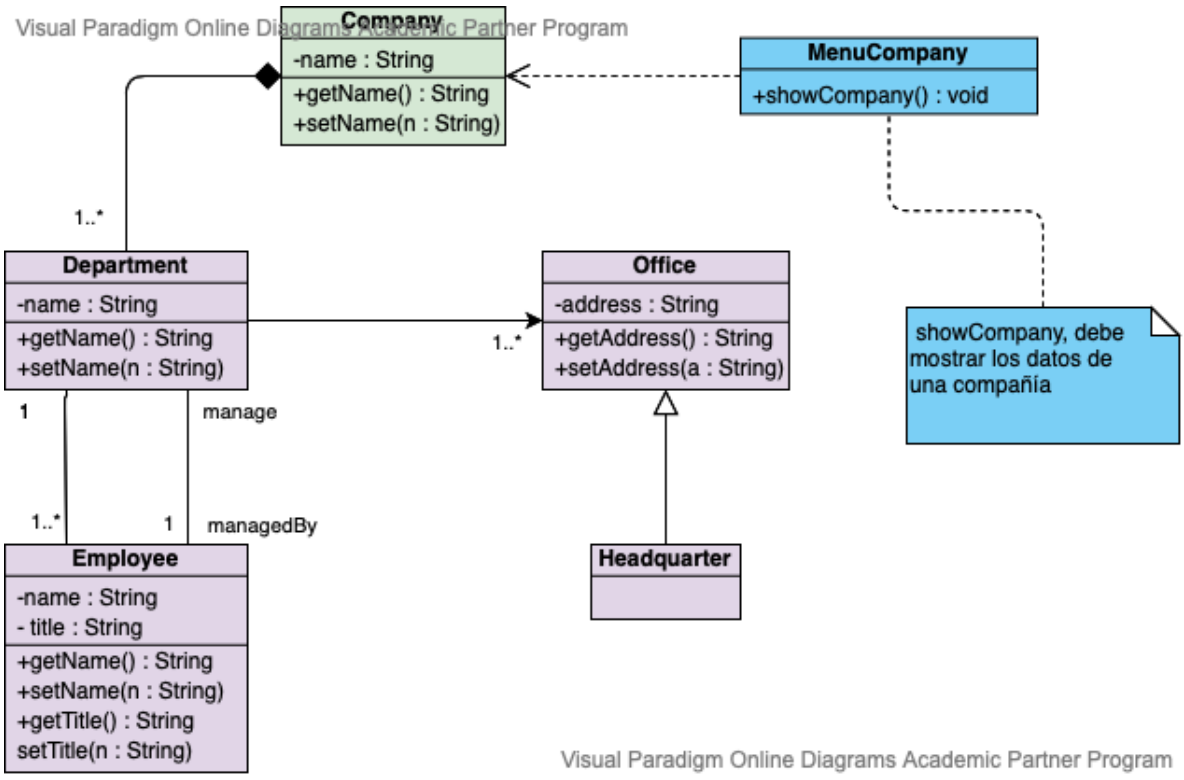
```java
public enum Genero {
    POP,
    HIP_HOP,
    ROCK;
}
```

Además considere que un *usuario* puede usar las funcionalidades que le otorga Spotifly. Las acciones que un usuario puede hacer son: escuchar una canción, escuchar una lista, buscar cancion (por título) y buscar lista (por nombre) dentro del repositorio, cada usuario dispondrá de un nickName como identificador.

2.) Suponga que está postulando a un nuevo cargo de Ingeniero de Software en la prestigiosa empresa internacional YVM (ninguna relación con la otra...IBx)
Como prueba de sus capacidades en el modelado de software, le proponen modelar en UML con diagramas de clases los siguientes enunciados.

a. Una empresa de transporte de carga cuenta con 12 caminones por el momento, pero está pronta a comprar nuevos y dar de baja algunos que ya están obsoletos. Entre los 12 camiones, hay camiones del tipo camión ¾ y camiones del tipo carga pesada.
b. Una persona estima al menos a otra persona, que puede ser considerada su amigo/a. Cada persona es dueña al menos un teléfono móvil, pero el teléfono no sabe quien es su dueño.
c. Un equipo deportivo tiene al menos un jugador y cada jugador pertenece a un sólo equipo. Cada equipo es capitaneado por un jugador. Hay 3 tipos de equipos: fútbol, polo y rayuela.

3.) Entendiendo el código fuente
El analista de vuestro equipo de desarrollo ha terminado de modelar los aspectos fundamentales de la empresa del cliente. Le presenta al resto del equipo de desarrollo el siguiente modelo de clases UML, tras lo cual les pide implementar en lenguaje Java "el esqueleto" de las clases y relaciones que se puedan inferir del modelo.



OBS: la respuesta, además de la explicación, la puede entregar “pegando” los segmentos de su código fuente o indicando el link a su repositorio Github.