

Algorítmica y Programación por Objetos 2

Ejercicio de Nivel 8

CupiPalooza

Descripción global

Se desea construir una aplicación que permita administrar las bandas y los escenarios que constituyen el festival de música CupiPalooza. Este festival está compuesto por escenarios, y en cada escenario hay una lista de bandas que se presentarán en él.

Se desea que la aplicación permita agregar, eliminar y ordenar las bandas, y crear o eliminar escenarios.

Objetivos del ejercicio

El objetivo de este ejercicio es que el estudiante comprenda y adquiera práctica en:

- El desarrollo de aplicaciones siguiendo un proceso incremental.
- La creación y captura de distintos tipos de excepciones para informar al usuario de manera conveniente cualquier problema detectado.
- La construcción de métodos para hacer persistir la información del modelo del mundo por medio del mecanismo de serialización.
- La construcción de métodos para importar y exportar información desde y hacia archivos secuenciales de texto, utilizando las clases que provee Java para manejo de archivos y manipulación de cadenas de caracteres.
- La utilización del elemento gráfico JComboBox de Java como mecanismo para manejar listas desplegables en la interfaz de usuario.

Los siguientes pasos conforman el plan sugerido para desarrollar el ejercicio. La idea es ir desarrollando y probando incrementalmente los métodos de las clases.

Este ejercicio debe ser realizado de manera **INDIVIDUAL**.

Preparación

Esta sección presenta una lista de chequeo de todas las tareas necesarias para la preparación del ejercicio. Por favor, revise que cada tarea haya sido completada **antes** de pasar a la siguiente sección de esta guía de trabajo.

Nota: En el siguiente enlace se encuentran las instrucciones de instalación de Java y Eclipse: <https://cupitaller.uniandes.edu.co/instaladores/>.

1. Para conocer el funcionamiento esperado de la aplicación, descargue y/o ejecute el archivo demo del ejercicio que se encuentra en el siguiente enlace:
<https://cupi2.virtual.uniandes.edu.co/ejercicios-del-semester-apo2/ejercicio-n8>.
2. Descargue el esqueleto del ejercicio que se encuentra en el siguiente enlace:
<https://cupi2.virtual.uniandes.edu.co/ejercicios-del-semester-apo2/ejercicio-n8>.
3. Descomprima este archivo e importe el proyecto llamado **n8_cupiPalooza** a Eclipse. Limpie el ejercicio; para ello en Eclipse vaya a: **Project > Clean > Ok**. En el siguiente enlace hay un video con un ejemplo de cómo limpiar un proyecto en eclipse:
<https://youtu.be/mbcpY46wXS0>.
4. Lea el enunciado del problema disponible en:
n8_cupiPalooza/docs/specs/Descripcion.pdf.
5. Estudie el documento de requerimientos funcionales disponible en:
n8_cupiPalooza/docs/specs/RequerimientosFuncionales.pdf.
6. Estudie el documento de requerimientos no funcionales disponible en:
n8_cupiPalooza/docs/specs/RequerimientosNoFuncionales.pdf.
7. Estudie el documento de consideraciones adicionales de diseño disponible en:
n8_cupiPalooza/docs/specs/ConsideracionesAdicionalesDeDisenho.pdf.
8. Estudie el modelo del mundo diseñado para este ejercicio. Este modelo se encuentra en: **n8_cupiPalooza/docs/specs/ModeloConceptual.png**. Identifique las clases, relaciones entre clases, constantes, atributos y métodos.
9. Asegúrese de tener activado el uso de aserciones para la ejecución del programa. Puede ver el tutorial en: <http://cupitaller.virtual.uniandes.edu.co/videos-guia/>.

Desarrollo

Parte 1: Clase Banda

1. Copie la clase Banda desarrollada en el nivel 7.
2. Modifique la declaración de la clase para que sea serializable.

Parte 2: Creación de nuevas excepciones

1. Revise el modelo del mundo e identifique las clases que permiten manejar las excepciones de la aplicación. Estudie su relación y dependencia con el resto del modelo.

2. Implemente y documente las clases propias de excepción de la aplicación. Estas son: **PersistenciaException**, **FormatoArchivoException**, **ElementoExistenteException** y **CupoMaximoException**. Para su implementación, puede guiarse por su definición en el modelo conceptual y en el documento de consideraciones adicionales de diseño. Las excepciones **DEBEN** estar de acuerdo al modelo conceptual.

Parte 3: Clase Escenario

En el archivo Escenario.txt encontrará los contratos y firmas de algunos de los métodos que debe completar de la clase Escenario. Recuerde reemplazar los contratos y las firmas de sus métodos con los proporcionados en el archivo.

1. Copie la clase Escenario desarrollada en el nivel 7.
2. Modifique la declaración de la clase para que sea serializable.
3. Modifique el método agregarBanda para que cumpla con la documentación que se encuentra en el archivo Escenario.txt.
4. Modifique el método eliminarBanda para que cumpla con la documentación que se encuentra en el archivo Escenario.txt.

Parte 4: Clase Festival

En el archivo Festival.txt encontrará los contratos y firmas de algunos de los métodos que debe completar de la clase Festival. Recuerde reemplazar los contratos y las firmas de sus métodos con los proporcionados en el archivo.

1. Copie la clase Festival desarrollada en el nivel 7.
2. Modifique la declaración de la clase para que sea serializable.
3. Modifique el método constructor para que cumpla con la documentación que se encuentra en el archivo Festival.txt.
4. Modifique el método agregarBanda para que cumpla con la documentación que se encuentra en el archivo Festival.txt
5. Modifique el método eliminarBanda para que cumpla con la documentación que se encuentra en el archivo Festival.txt
6. Modifique el método crearEscenario para que cumpla con la documentación que se encuentra en el archivo Festival.txt

7. Cree e implemente el método guardar para que cumpla con la documentación que se encuentra en el archivo Festival.txt.
8. Cree e implemente el método cargar para que cumpla con la documentación que se encuentra en el archivo Festival.txt.
9. Cree e implemente el método importarArchivoTexto para que cumpla con la documentación que se encuentra en el archivo Festival.txt. Tenga en cuenta el formato del archivo descrito en el documento de descripción del ejercicio.
10. Cree e implemente el método generarReporte para que cumpla con la documentación que se encuentra en el archivo Festival.txt. Tenga en cuenta el formato del reporte descrito en el documento de descripción del ejercicio.

Parte 5: Modificación de la interfaz

En PanelOpciones completar los TODOs que comienzan por: **//TODO**

Validación

Para comprobar el funcionamiento de su ejercicio usted puede:

1. Ejecutar el programa e interactuar con todas las opciones disponibles en la interfaz. Los resultados obtenidos deben ser iguales a aquellos mostrados en el video demo.
2. Ejecutar las pruebas automáticas disponibles en el ejercicio. En el siguiente video <https://youtu.be/rVd4AD8XMJk> se explica cómo efectuar esas pruebas. Estas pruebas deben presentar resultados en verde (0 errores y 0 fallas) cuando el ejercicio ha sido completado correctamente.

Tenga en cuenta que esas pruebas no son exhaustivas y que su correcto funcionamiento no garantiza que no haya ningún error en su programa.

Entrega

Este ejercicio debe ser realizado de manera **INDIVIDUAL**.

1. Indente el código fuente de todas las clases del mundo. En el siguiente enlace <https://youtu.be/cwQ9QiauaSc> encuentra un video que explica cómo indentar el código fuente de su ejercicio.
2. Limpie el proyecto para que la entrega no contenga archivos ejecutables ni temporales (<https://youtu.be/mbcpY46wXS0>).

3. Construya el archivo entregable con su ejercicio desarrollado y validado completamente. En el siguiente video <https://youtu.be/xuSDFfEZW78> se explica detalladamente el proceso para producir el comprimido del ejercicio y enviarlo a SicuaPlus. Renombre el archivo a entregar con su login de la siguiente forma:

n<nivel del ejercicio>_<login estudiante>.zip

Por ejemplo: **n8_tsuarez.zip**

La no indentación del código fuente o el nombramiento incorrecto del ejercicio en su entrega es una acción penalizada en la plantilla de calificación del mismo.

4. Entregue el archivo del ejercicio vía SicuaPlus, de acuerdo con las normas, fecha y hora de entrega.