

CURSO DE PROGRAMACIÓN CON JAVA

LABORATORIO FINAL



Por el experto: Ing. Ubaldo Acosta



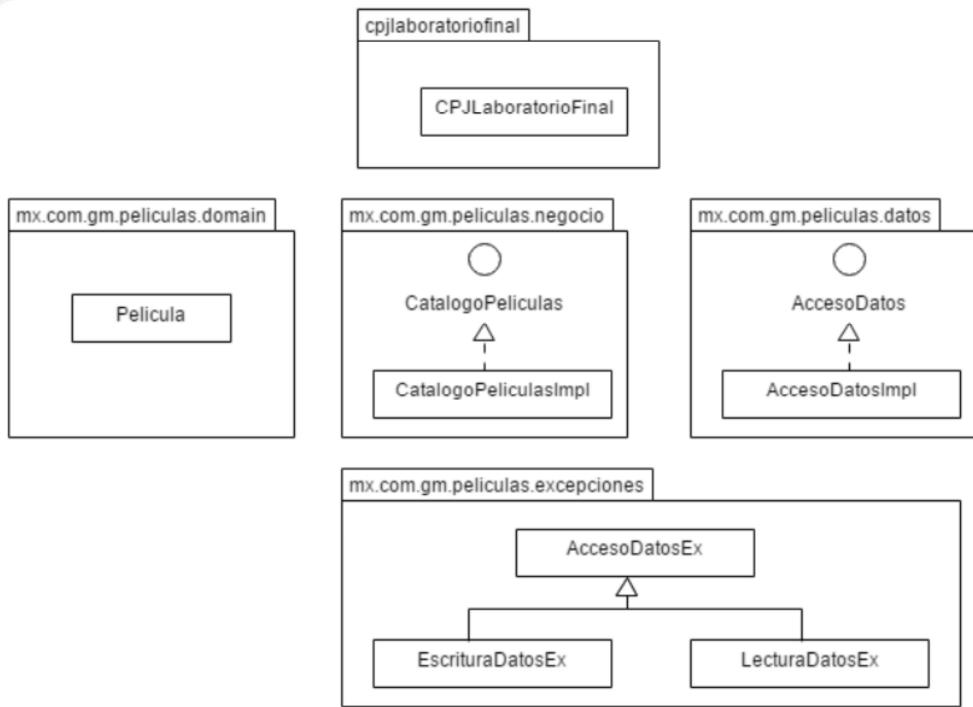
www.globalmentoring.com.mx

Hola, te saluda Ubaldo Acosta. Bienvenidos nuevamente a este curso de Programación con Java.

Esta es nuestra lección final, por lo que vamos a realizar un ejercicio donde pondremos en práctica lo visto en el curso de Programación con Java.

Comencemos de inmediato.

DIAGRAMA DE CLASES



En este laboratorio final vamos a crear un ejercicio que simule un catálogo de películas. Para este ejercicio, vamos a almacenar la información del catálogo de películas en un archivo de texto, en una carpeta definida por nosotros, por ejemplo en la carpeta: c:\catalogoPeliculas\películas.txt

Vamos a crear varios paquetes, cada uno con tareas específicas a desarrollar en el programa.

Paquetes:

1. Crear un paquete de excepciones, similar a la lección de excepciones vistas en el curso, según se muestra en el diagrama de clases.
2. Crear un paquete mx.com.gm.películas.domain que incluya una clase llamada Pelicula. Las clases almacenadas en este paquete, se conoce como las clases del dominio del problema.
3. Crear un paquete llamado mx.com.gm.películas.datos la cual incluye una interface y una clase que implementa dicha interface. El objetivo de estas clases es agregar la funcionalidad para interactuar con el archivo donde se va a guardar la información del catálogo de películas.
4. Crear el paquete llamado mx.com.gm.peliculas.negocio el cual incluye las clases para definir la funcionalidad de nuestra aplicación, también conocido como las reglas de negocio de la aplicación.
5. Finalmente crearemos la clase CPJLaboratorioFinal, que significa CursoProgramacionJavaLaboratorioFinal, el cual incluye un menú de opciones para escoger la opción a procesar en el programa.

A continuación veremos a más detalle cada uno de los paquetes de la aplicación Catalogo de Peliculas a desarrollar.

PELICULA.JAVA

mx.com.gm.peliculas.domain::Pelicula
-nombre: String
+Pelicula()
+Pelicula(nombre: String)
+getNombre(): String
+setNombre(nombre:String)
+toString(): String
Responsibilities
-- Representa los objetos película utilizados en la aplicación catálogo de películas

CURSO DE PROGRAMACIÓN CON JAVA

www.globalmentoring.com.mx

Esta clase nos servirá para representar el objeto Pelicula. En esta aplicación dejamos simple la definición de este objeto, sin embargo podría tener más atributos y hacer más compleja esta aplicación, pero el objetivo es que podamos desarrollar esta aplicación con la estructura planteada, y no agregar más complejidad de la necesaria para aplicar los conceptos estudiados a lo largo de este curso.

Por ello este objeto se debe codificar con lo indicado en el diagrama de clase mostrado.

ACCESODATOS.JAVA

mx.com.gm.peliculas.datos::AccesoDatos
+existe(nombreArchivo : String): boolean
+listar(nombre: String): List<Pelicula>
+escribir(pelicula: Pelicula, nombreArchivo: String, anexar: boolean): void
+buscar(nombreArchivo:String, buscar:String): String
+crear(nombreArchivo: String): void
+borrar(nombreArchivo: String): void
Responsibilities
-- Contiene las operaciones a ejecutar en el archivo de peliculas.txt

CURSO DE PROGRAMACIÓN CON JAVA

www.globalmentoring.com.mx

Esta interfaz define los métodos para interactuar con el archivo que administrará el catálogo de películas. La clase AccesoDatosImpl es la que finalmente se encargará de agregar el comportamiento de cada uno de los métodos descritos.

A partir de este proyecto empezaremos a separar los paquetes y clases según la funcionalidad de cada una de ellas. En el caso del paquete de acceso a datos, su función es definir la forma en que se interactúa con el archivo que es donde se almacenará la información.

Sin embargo, el uso de interfaces y su implementación empieza a tomar importancia, debido a que en el curso de Java con JDBC, en lugar de trabajar con la implementación para un guardar la información en un archivo, agregaremos la implementación pero escribiendo a una base de datos. Así que por ello es importante definir la interface que es la que contiene los métodos que se utilizarán por otras clases en la aplicación, pero la implementación podemos definir cual se va a utilizar, si por ejemplo una que guarde la información en un archivo o alguna otra implementación que guarde la información en base de datos.

Por último, cabe mencionar que la implementación de esta interfaz es completamente técnica respecto al uso del manejo de archivos, por lo que a diferencia del paquete de negocios que veremos a continuación, el paquete de datos, realiza las tareas más técnicas que tienen que ver con la manipulación directa del archivo, en cambio, el paquete de negocios se encarga más de definir las operaciones y funcionalidad desde el punto de vista de lo que necesita realizar la aplicación, pero de alguna manera no le interesa si la información se almacena o proviene de un archivo, ya que de eso se encargará precisamente el paquete de datos.

CATALOGOPELICULAS.JAVA

mx.com.gm.peliculas.negocio::CatalogoPeliculas
--

+agregarPelicula(nombrePelicula: String, nombreArchivo : String): void
+listarPeliculas(nombreArchivo: String): void
+buscarPelicula(nombreArchivo: String, buscar: String): void
+iniciarArchivo(nombreArchivo: String): void

Responsibilities

-- Contiene las operaciones necesarias de la aplicación CatalogoPeliculas

mx.com.gm.peliculas.negocio::CatalogoPeliculasImpl
--

datos: AccesoDatos

+CatalogoPeliculasImpl()
+agregarPelicula(nombrePelicula: String, nombreArchivo : String): void
+listarPeliculas(nombreArchivo: String): void
+buscarPelicula(nombreArchivo: String, buscar: String): void
+iniciarArchivo(nombreArchivo: String): void

Responsibilities

-- Contiene la implementación de las operaciones necesarias de la aplicación CatalogoPeliculas
--

CURSO DE
www.globalmentoring.com.mx

El paquete de `mx.com.gm.peliculas.negocio` contiene las clases que implementarán las reglas de negocio de nuestra aplicación. Y a su vez este es el paquete que será el intermediario entre la interfaz con el usuario (método main) y el guardado de la información (paquete de datos).

Similar al paquete de datos, este paquete hemos agregado también una interfaz y una implementación. De esta manera la interfaz de usuario (método main) hará uso de la interface, y será la implementación de `CatalogoPeliculasImpl.java` quien finalmente ejecutará las tareas que sean necesarias para esta aplicación.

A su vez la implementación del catálogo de películas será quien utilice la interface de `AccesoDatos.java`, por lo que al hacer uso de interfaces es que en cualquier momento podríamos cambiar la implementación y afectaríamos lo menos posible al código que ya hemos programado. A esta característica se le conoce como **bajo acoplamiento**, ya que un cambio en un componente o una clase afecta lo menos posible a otra clase, así que es sólo una de las muchas ventajas de programar utilizando interfaces y no utilizar directamente la implementación de la funcionalidad que necesitamos.

CLASE MAIN

```

cpjlaboratoriofinal::CPJLaboratorioFinal
scanner: java.util.Scanner
opcion: int
nombreArchivo: String
catalogoPeliculas: CatalogoPeliculas
+main(args: String[]): void
Responsibilities
-- Contiene el menu que permite al usuario
seleccionar la accion a ejecutar
sobre el catalogo de peliculas
  
```

CURSO DE PROGRAMACIÓN CON JAVA

www.globalmentoring.com.mx

Esta clase representa la interface con la que el usuario interactua. Esta clase solicita via consola la acción requerida por parte del usuario, y solicita opciones como:

- 1.- Iniciar catalogo películas
- 2.- Agregar película
- 3.- Listar Películas
- 4.- Buscar Película
- 0.- Salir

Y dependiendo de la opción elegida por el usuario es la acción que se ejecuta.

Esta clase se apoya de la clase Catalogopeliculas para realizar las operaciones que necesita esta aplicación.

A su vez la variable opción es la que se encargará de almacenar la opción seleccionada por parte del usuario.

Y el nombre del archivo indica la ruta donde se almacenará el archivo. Por ejemplo, la cadena Java que se puede utilizar si estamos en Windows es:

"c:\\catalogoPeliculas\\peliculas.txt"

Sin embargo si es otro sistema operativo, como MacOS, la ruta podría ser:

"/Volumes/gm/catalogoPeliculas/películas.txt"

Debemos recordar que la carpeta sobre la que vayamos a trabajar ya debe estar creada y en caso necesario debe tener permisos de escritura, esto es particular de los requerimientos de cada sistema operativo.

SALIDA DEL PROGRAMA

Elige opción:

- 1.- Iniciar catalogo peliculas
- 2.- Agregar pelicula
- 3.- Listar Peliculas
- 4.- Buscar Pelicula
- 0.- Salir

2

Introduce el nombre de una pelicula a agregar:

batman

Se ha escrito correctamente al archivo

Elige opción:

- 1.- Iniciar catalogo peliculas
- 2.- Agregar pelicula
- 3.- Listar Peliculas
- 4.- Buscar Pelicula
- 0.- Salir

3

Pelicula:batman

CURSO DE PROGRAMACIÓN CON JAVA

www.globalmentoring.com.mx

Según hemos comentado, las opciones que muestra el menú de opciones para el usuario son:

- 1.- Iniciar catalogo películas
- 2.- Agregar película
- 3.- Listar Películas
- 4.- Buscar Película
- 0.- Salir

Y como vemos, dependiendo de la opción elegida por el usuario es la acción que se ejecuta. Así que este es el menú que debemos programar para que el usuario pueda interactuar con la aplicación.

Así que tenemos todo listo para comenzar a realizar la aplicación de Catálogo de Películas, manos a la obra y te deseamos el mayor éxito en la realización de tu práctica final.

CURSO ONLINE

PROGRAMACIÓN CON JAVA

Por: Ing. Ubaldo Acosta



Experiencia y Conocimiento para tu vida

**CURSO DE PROGRAMACIÓN CON JAVA**www.globalmentoring.com.mx

En Global Mentoring promovemos la Pasión por la Tecnología Java. Te invitamos a visitar nuestro sitio Web donde encontrarás cursos Java Online desde Niveles Básicos, Intermedios y Avanzados, y así te conviertas en un experto programador Java.

A continuación te presentamos nuestro listado de cursos:

- | | |
|--------------------------|-------------------------------------|
| ✓ Lógica de Programación | ✓ Hibernate Framework |
| ✓ Fundamentos de Java | ✓ Spring Framework |
| ✓ Programación con Java | ✓ JavaServer Faces |
| ✓ Java con JDBC | ✓ Java EE (EJB, JPA y Web Services) |
| ✓ HTML, CSS y JavaScript | ✓ JBoss Administration |
| ✓ Servlets y JSP's | ✓ Android con Java |
| ✓ Struts Framework | ✓ HTML5 y CSS3 |

Datos de Contacto:Sitio Web: www.globalmentoring.com.mxEmail: informes@globalmentoring.com.mx