
Evaluación Formativa de la Práctica

Temas: Archivos.

Objetivos: Determinar si los alumnos han adquirido competencias para...

- Modelar y documentar la resolución de un problema mediante diagramación UML de clases, identificando las clases y las relaciones entre ellas.
- Emplear las estructuras de datos estudiadas, y sus operaciones básicas, en la resolución de problemas mediante diagramación de flujo.
- Desarrollar una solución de programación empleando la orientación a objetos mediante la creación de un proyecto en lenguaje Java utilizando un Entorno de Desarrollo Integrado y aplicando la reutilización de código.

Consignas.

- Esta evaluación formativa será desarrollada y presentada en forma grupal, y podrá ser solicitada su defensa en modalidad individual.
- Los diagramas UML de clases podrán ser desarrollados utilizando una aplicación o un plugin, así como en papel y lápiz.
- Los diagramas de flujo solicitados deberán estar completamente desarrollados, en lápiz y papel, de manera prolija, e incluyendo las descripciones necesarias para su mejor seguimiento (identific., var.utilizadas y func., etc.).
- La ejecución del proyecto deberá poder realizarse en forma completa y cumpliendo con todos los requisitos planteados en el problema en tratamiento.
- La codificación del proyecto en lenguaje Java deberá realizarse aplicando POO.
- La evaluación se calificará con Aprobado o Desaprobado, y en ambos casos podrá incluir indicaciones y correcciones que el grupo de estudio deberá aplicar sobre el trabajo presentado.

Criterios de evaluación y aprobación.

Se aplicará una valoración dicotómica de aprobado-desaprobado.

Para aprobar la presentación debe cumplir con los siguientes ítems:

- El trabajo solicitado debe estar desarrollado completamente.
- La codificación en lenguaje Java debe realizarse siguiendo el paradigma de la POO.
- El desarrollo de los diagramas de flujo debe realizarse cumpliendo las indicaciones relativas a la diagramación estructurada y modular.
- El programa deberá estar correctamente documentando mediante diagramación UML expresando las relaciones entre clases identificadas y sus propiedades y métodos.
- La presentación del código solicitado deberá realizarse en tiempo y forma mediante la plataforma CUV.FCEYT: <http://cuv.unse.edu.ar>.

Fechas de presentación:

- 05/09/2024: Diseño orientado a objetos (DOO), utilizando diagramación UML de clases y diagramación de flujo estructurado. Entrega personal, en papel y lápiz.
- 12/09/2024: Desarrollo, verificación y ejecución aplicando el lenguaje Java y la POO. Entrega mediante CUV.FCEYT.

Evaluación Formativa de la Práctica

Recursos bibliográficos

Bibliografía (*)	Páginas	Acceso web / Ubicación física	Tema
[Jimenez de Parga, 2021]	Cap. 6	https://elibro.net/es/lc/unsebiblio/titulos/222720/	Diagramación UML de clases
[Oviedo Regino, 2015]	Cap. 7 pags. 234-245	https://elibro.net/es/lc/unsebiblio/titulos/70431/	Pasos a seguir para resolver un problema mediante POO
	Cap. 7 pags. 271-278		Polimorfismo
[Blasco, 2019]	Cap. 8	https://elibro.net/es/lc/unsebiblio/titulos/222723/	Interfaces
	Cap. 5 pags. 225-241		Excepciones
	Cap. 7		Flujos de E/S
	Cap. 9		Polimorfismo
[Deitel, 2008]	Cap. 14	Centro de Documentación	Archivos
	Cap.10		Polimorfismo
	Cap. 13		Excepciones
[López Goytía, 2015]	Cap. 5, pags. 192-198	https://elibro.net/es/lc/unsebiblio/titulos/39461/	Polimorfismo
	Cap. 5, pags. 175-191		Excepciones
[Joyanes, 2008]	Cap.4 pags.112-120	Centro de Documentación	Polimorfismo
[Vegas Gertrudix, 2020]	Cap. 1 pags. 43-47	https://elibro.net/es/lc/unsebiblio/titulos/222715/	Interfaces
	Cap. 1 pags. 47-54		Excepciones
	Cap. 1 pags. 70-75		Excepciones

(*) La descripción completa de la bibliografía se encuentra en el documento de la Planificación de la asignatura.

Evaluación Formativa de la Práctica

Descripción General del Problema:

La Biblioteca Sarmiento necesita un sistema para gestionar las publicaciones que maneja. La biblioteca cuenta con dos tipos de publicaciones:

- libros, y
- revistas.

El sistema debe permitir registrar, actualizar, consultar y listar la información de estas publicaciones, guardando todos los datos en archivos directos.

Se debe aplicar:

- Paradigma de Programación Orientada a Objetos (POO).
- Herencia entre clases y polimorfismo.
- Manejo de excepciones.
- Flujos de entrada/salida: Archivos directos.

Requisitos Funcionales:

1. Información involucrada en el problema:

Cualquiera de las publicaciones cuenta con un título, un autor, el año de publicación y el código ISBN que la identifica de manera única.

Los libros cuentan con una determinada cantidad de páginas y la editorial que los publica. Por otra parte, las revistas cuentan con un número de edición y un mes de publicación.

La muestra de la información de cada uno de los tipos de publicaciones implica considerar la información que las caracteriza en cada caso.

2. Gestión de Archivos Directos:

- Los datos de las publicaciones deben ser almacenados en archivos directos usando las clases RandomAccessFile y File, y reutilizando el proyecto proporcionado a tal efecto.
- Se debe implementar un método grabar() en cada clase, que permita guardar la información de la publicación en un archivo directo.
- Se debe implementar un método leer() en cada clase, que permita leer y reconstruir un objeto desde el archivo.
- Cada clase deberá contar con todos los métodos necesarios para responder a la estructura del proyecto proporcionado.
- El funcionamiento de las clases Archivo y Registro no deberá ser afectado como resultado de la reutilización del proyecto.

3. Operaciones que debe completar el sistema:

- Alta de publicaciones: Permitir al usuario ingresar datos de nuevas publicaciones, ya sean libros o revistas, y guardarlos en el archivo correspondiente.

Gestión de Biblioteca Sarmiento

- 1- Alta de publicaciones
- 2- Bajas de publicaciones
- 3- Consultar publicaciones
- 4- Actualización publicaciones
- 5- Listar publicaciones
- 6- Salir

Ingrese su opción:

Evaluación Formativa de la Práctica

- Consultar Publicaciones: Permitir al usuario buscar publicaciones por código ISBN y mostrar su información.
- Actualizar publicaciones: Permitir al usuario modificar los datos de publicaciones existentes y actualizar la información en el archivo.
- Eliminar publicaciones: Permitir al usuario eliminar registros de publicaciones del archivo.
- Listar Publicaciones: Permitir que el usuario obtenga en pantalla el detalle de todas las publicaciones, que podrán ser libros o revistas (no ambos). Los datos deben mostrarse encolumnados.

4. Interfaz de Usuario:

- Implementar una interfaz de usuario simple basada en consola para interactuar con el sistema.
- La interfaz debe permitir al usuario realizar las operaciones mencionadas de manera sencilla e intuitiva.
- Todos los mensajes deben ser claros para la operación por parte del usuario.
- Todas las entradas de datos deben ser adecuadamente validadas con su correspondiente información (mensaje) al usuario.

Requisitos No Funcionales:

1. Desempeño: El sistema debe ser capaz de manejar hasta 100 publicaciones por archivo de manera eficiente.
2. Integridad: Los archivos deben mantener la integridad de los datos entre ejecuciones del programa.
3. Seguridad: El sistema debe verificar que no se puedan ingresar publicaciones con el mismo código ISBN dos veces.
4. Estructura de archivos: Cada registro en los archivos directos debe incluir todos los atributos de la clase correspondiente, y deben estar organizados de manera que se permita una lectura y escritura directa en cualquier punto del archivo.

Consideraciones:

- Deben aplicarse correctamente las relaciones entre clases, considerando la relación de herencia y por tanto, la sobrescritura de métodos.
- El sistema debe ser capaz de aplicar adecuadamente el polimorfismo.
- El manejo de excepciones debe realizarse observando en particular aquellas que surjan de la lectura y escritura de archivos.
- El sistema debe ser capaz de gestionar dos tipos de publicaciones con características diferentes, almacenarlos en archivos y proporcionar funcionalidades básicas de CRUD (Crear, Leer, Actualizar, Eliminar).