PROGRAMACIÓN III

Trabajo Práctico 6 - Colecciones

OBJETIVO GENERAL

- Practicar el uso de listas ordenadas en Java, permitir duplicados y mostrar los datos de los objetos de manera legible mediante la sobreescritura de toString().
- Comprender cómo asegurar la unicidad de los elementos en una colección Set, utilizando equals() y hashCode().
- Practicar el uso de Map para almacenar pares clave–valor, permitiendo acceso rápido a los objetos mediante una clave única

MARCO TEÓRICO

Concepto	Aplicación en el proyecto
List	Colección ordenada que permite duplicados.
ArrayList	Implementación dinámica de List, permite acceso por índice.
toString()	Método sobrescrito para representar objetos como cadenas legibles.
Set	Colección que no permite duplicados.
HashSet	Implementación basada en hash, rápida para búsqueda.
equals() y hashCode()	Definen la unicidad de los objetos en un Set.
Map	Colección de pares clave–valor, con claves únicas.
HashMap	Implementación basada en hash para acceso rápido.

entrySet()	Permite recorrer todos los pares clave–valor.

Caso Práctico 1

Crear un sistema que:

1. Crear clase Alumno con atributos nombre y nota.
2. Sobrescribir toString().
3. Almacenar varios alumnos en ArrayList.
4. Recorrer la lista con for-each e imprimir cada alumno.

CONCLUSIONES ESPERADAS

Se recorre e imprime la lista de alumnos respetando el orden de inserción. La sobrescritura de toString() facilita una salida legible. Los duplicados son permitidos.

Caso Práctico 2

Crear un sistema que:

1. Crear clase Producto con código y descripción.
2. Sobrescribir equals() y hashCode() basados en código.
3. Insertar varios productos en un HashSet, incluyendo duplicados.
4. Recorrer el set e imprimir los productos.

CONCLUSIONES ESPERADAS

El HashSet no permite elementos duplicados basados en codigo. Se garantiza la unicidad y se puede recorrer la colección mostrando productos de manera clara.

Caso Práctico 3

Crear un sistema que:

1. Crear clase Curso con atributos nombre y docente.
2. Usar HashMap<String, Curso> donde la clave es el código del curso.
3. Agregar varios cursos.
4. Recuperar cursos por clave.
5. Recorrer el map con entrySet() e imprimir las entradas.

CONCLUSIONES ESPERADAS

Permite acceder a los cursos de manera eficiente usando la clave. Se pueden recorrer todos los pares clave–valor y mostrar la información completa de forma clara.