

Tarea Final de la Unidad PROG08

Realizar un ejercicio en JAVA sobre el desarrollo de clases con Arrays.

Entrega: Deberás crear una carpeta comprimida en formato 'zip' con el nombre "Apellido_Nombre_Tarea_Final_PROG08". Dentro de esta carpeta, incluye el proyecto o el paquete donde están las clases del ejercicio.

Finalmente, sube el archivo comprimido en la tarea correspondiente del aula virtual.

Enunciado del Problema

Título: Gestión de Estudiantes y Notas

Descripción:

Desarrolla un programa en Java que permita a un profesor gestionar las **notas de exámenes y tareas** de un número **ilimitado** de estudiantes (hasta donde la memoria lo permita), calcular sus medias individuales y la media de la clase, y mostrar resúmenes detallados.

Se trabajará con **colecciones de datos en Java** en lugar de arrays.

Puedes seguir con lo que entregaste o la solución propuesta de la tarea de la unidad PROG06.

El programa debe incluir las siguientes funcionalidades y estar basado en las siguientes especificaciones:

Especificaciones

Clase Estudiante

1. Atributos:

- dni: Identificador único del estudiante.
- nombre: Nombre del estudiante.
- notasExamen: **ArrayList<Double>** para almacenar las notas de los exámenes.
- notasTarea: **ArrayList<Double>** para almacenar las notas de las tareas.

2. Constructores:

- Constructor parametrizado que reciba el dni y el nombre. Inicializa las listas notasExamen y notasTarea.

3. Métodos:

- añadirNotaExamen(double nota): Añade una nota a la lista de exámenes.
- añadirNotaTarea(double nota): Añade una nota a la lista de tareas.
- calcularMediaExámenes(): Calcula y devuelve la media de las notas de examen.
- calcularMediaTareas(): Calcula y devuelve la media de las notas de tarea.
- calcularNotaMedia(): Calcula y devuelve la nota final ponderada (75% exámenes, 25% tareas).
- ordenarNotasExamen(boolean ascendente):
Ordena las notas de examen **de menor a mayor** (true) o **de mayor a menor** (false).
- ordenarNotasTarea(boolean ascendente):
Ordena las notas de tarea **de menor a mayor** (true) o **de mayor a menor** (false).
- mostrarNotas(), muestras las notas de un estudiante, tanto de exámenes como de tareas.
- @Override public String toString(): Devuelve una cadena con toda la información del estudiante, incluyendo las medias de exámenes, tareas y la nota final.

Programa Principal

1. Estructura de almacenamiento:

Se usará un **HashMap<String, Estudiante>**, donde la clave será el **DNI** del estudiante.

2. Menú interactivo: El programa debe mostrar un menú con las siguientes opciones:

a) Crear estudiante:

- Solicita el DNI y el nombre del estudiante.

- Crea un nuevo objeto Estudiante y lo almacena en el HashMap.
- Si el estudiante ya existe, muestra un error.

b) Introducir nota a un estudiante:

- Solicita el DNI del estudiante.
- Si el estudiante no existe, muestra un mensaje de error.
- Si el estudiante existe, pregunta si se desea introducir una nota de **examen** o de **tarea**.
- Solicita la nota (debe ser un número double entre 0 y 10) y la **almacena en la lista correspondiente**.

c) Ordenar notas de un estudiante

- Solicita el **DNI** del estudiante.
- Si **no existe**, muestra un mensaje de error.
- Si **existe**, permite ordenar sus **notas de examen** o **notas de tarea**.
- Solicita si el usuario quiere ordenar las notas **ascendente** o **descendente**.

d) Mostrar notas de un estudiante:

- Solicita el DNI del estudiante.
- Si **no existe**, muestra un mensaje de error.
- Si **existe**, **muestra sus notas**.

e) Mostrar resumen de estudiantes:

- Muestra un listado de todos los estudiantes, con su DNI, nombre, medias de exámenes, medias de tareas y nota final.

f) Calcular media de la clase:

- Calcula y muestra la media global de todas las notas finales de los estudiantes registrados.

g) Salir:

- Finaliza el programa.

Rúbrica de Evaluación

Criterio	Descripción	Puntos
Clase Estudiante	Implementación correcta de atributos y constructores.	1
Métodos de la clase Estudiante	Implementación correcta de los métodos añadirNotaExamen, añadirNotaTarea, calcularMediaExámenes, calcularMediaTareas, calcularNotaMedia, ordenarNotasExamen, mostrarNotas, ordenarNotasTarea y toString.	2
Estructura de almacenamiento	Se usa un HashMap<String, Estudiante> en lugar de un array.	1
Menú interactivo	Se implementa correctamente el menú con las opciones indicadas.	1
Ordenación de notas	Se implementa la funcionalidad de ordenar las notas en orden ascendente y descendente.	1
Mostrar resumen de estudiantes	Se implementa correctamente la funcionalidad para mostrar el resumen de estudiantes.	1
Mostrar notas estudiante	Se implementa correctamente la funcionalidad para mostrar las notas de un estudiante.	1
Calcular media de la clase	Implementa correctamente el cálculo de la media de la clase.	1
Estructura general del programa	Uso correcto de colecciones (ArrayList, HashMap) , clases y métodos.	1
Total	10 puntos	