



ACTIVIDAD

EL ENTORNO JAVA PARA LA PROGRAMACIÓN

Aprendizaje esperado

Utilizar la sintaxis básica del lenguaje java para la construcción de programas que resuelven un problema de baja complejidad.

Actividad

Contexto:

Se te ha solicitado desarrollar un sistema básico de gestión de notas para un grupo de estudiantes. El sistema debe permitir ingresar las notas de los estudiantes, calcular el promedio de sus calificaciones, determinar si aprobaron o no, y mostrar los resultados de manera ordenada. Para ello, deberás trabajar con arreglos, estructuras de control, y realizar cálculos utilizando las clases básicas de Java.

Este sistema debe ser capaz de gestionar hasta 30 estudiantes y debe incluir funcionalidades para agregar nuevas calificaciones, visualizar las calificaciones de los estudiantes, calcular el promedio y mostrar los estudiantes aprobados o reprobados según el promedio.

Objetivo:

El objetivo de esta actividad es aplicar el conocimiento sobre variables en Java, tipos de datos primitivos, operadores, expresiones y sentencias condicionales, sentencias repetitivas, arreglos, colecciones, y operaciones básicas con la clase String para crear un sistema funcional que resuelva un problema de gestión de datos numéricos (notas de estudiantes).

Instrucciones:

1. Definición de Clases y Estructura de Datos:

- Crea una clase Estudiante que tendrá los siguientes atributos:
 - Nombre (String): Nombre del estudiante.
 - Notas (arreglo de tipo double[]): Arreglo para almacenar las calificaciones de las materias del estudiante.
 - Promedio (double): Promedio de las notas del estudiante.

- Crea una clase GestionNotas que contendrá los métodos y operaciones del sistema. Esta clase debe permitir agregar estudiantes, calcular promedios, verificar si aprobaron, y mostrar el listado de estudiantes.

2. Operaciones Básicas:

- Agregar Estudiantes: Implementa un método para agregar un nuevo estudiante con su nombre y las calificaciones de las materias. Este método debe almacenar al estudiante en un arreglo de objetos Estudiante[].
- Calcular Promedio: Implementa un método que calcule el promedio de las notas del estudiante.
- Aprobar/Reprobar: Implementa un método que verifique si el estudiante aprobó o reprobó en base a su promedio. El promedio debe ser mayor o igual a 4.0 para aprobar.
- Mostrar Estudiantes: Implementa un método que recorra el arreglo de estudiantes y muestre el nombre, las notas, el promedio y el estado (aprobado/reprobado) de cada uno.
- Ordenar Estudiantes: Implementa un método que ordene a los estudiantes de acuerdo con su promedio (de mayor a menor).

3. Manejo de Entrada y Salida:

- Crea un menú en consola con las siguientes opciones:
 - Agregar estudiante.
 - Calcular y mostrar promedio.
 - Mostrar todos los estudiantes.
 - Mostrar estudiantes aprobados.
 - Salir.

4. Requerimientos de implementación:

A. Variables y Tipos de Datos:

- Variables primitivas: Usar tipos como int, double, String.
- Expresiones y Operadores: Utilizar operadores de asignación (=), operadores aritméticos (+, -, *, /), operadores lógicos (&&, ||), operadores de comparación (==, !=, >, <), y operadores unarios (++ , --).

B. Sentencias Condicionales:

- Usar sentencias if-else para:
 - Verificar si el promedio es mayor o igual a 4.0 para determinar si el estudiante aprobó.
 - Validar la entrada de datos, como el nombre del estudiante y las notas.

C. Sentencias Repetitivas:

- Utilizar bucles for o while para recorrer los arreglos de estudiantes y mostrar sus notas.
- Usar un bucle while para permitir que el usuario seleccione opciones del menú hasta que decida salir.

D. Arreglos y Colecciones:

- Usar un arreglo de tipo Estudiante[] para almacenar los estudiantes.
- El número de estudiantes no debe exceder de 30, por lo que el tamaño del arreglo será de 30.

E. Operaciones Básicas con la Clase String:

- Usar métodos de la clase String para realizar operaciones sobre los nombres de los estudiantes, como equalsIgnoreCase(), substring(), toUpperCase(), entre otros.

F. Operaciones Básicas con la Clase Math:

- Utilizar la clase Math para redondear el promedio del estudiante utilizando Math.round(), si es necesario.

Desarrollo del Programa

1. **Clase Estudiante:** Define los atributos y métodos para manejar los datos de los estudiantes.
 - La clase debe incluir los atributos mencionados y un constructor para inicializarlos.
 - Crear métodos para calcular el promedio de notas y determinar si el estudiante aprobó o reprobó.
2. **Clase GestionNotas:** Gestiona el sistema de notas de los estudiantes.
 - Crear un arreglo de objetos Estudiante[] para almacenar los estudiantes.
 - Crear métodos para agregar estudiantes, calcular promedios, verificar si aprobaron, y ordenar los estudiantes según su promedio.
 - Implementar validaciones para asegurarse de que las notas sean válidas (por ejemplo, que no sean menores que 0 ni mayores que 7).
3. **Clase Principal (Main):** Configura y ejecuta el sistema.
 - Crea un objeto de la clase GestionNotas.
 - Mostrar un menú con las opciones disponibles.
 - Ejecutar los métodos de la clase GestionNotas según la opción seleccionada por el usuario.

Entregables:

1. **Código en Java:** El código debe estar completamente funcional, organizado y comentado. El estudiante debe entregar el archivo .java correspondiente.
2. **Informe:** Un informe breve sobre la implementación, mencionando los conceptos trabajados (variables, operadores, arreglos, colecciones, etc.), cómo se resolvió el problema y cualquier dificultad encontrada.