Profundiza más

Recurso de Profundización

Clase 11: Estructuras de Control en Conjunto y Arreglos (Parte 1)

Autor: Damián Nicolalde Rodríguez

Objetivo del Documento:

El objetivo de este recurso es proporcionar un ejercicio práctico que permita a los estudiantes aplicar todos los conocimientos adquiridos en las clases anteriores, desde el uso de estructuras condicionales (if, else, elif) hasta el manejo de bucles anidados, validación de entradas, y manipulación avanzada de arreglos unidimensionales. El ejercicio se centrará en la creación de un sistema interactivo que permita gestionar una lista de estudiantes, almacenar calificaciones y generar estadísticas de rendimiento, todo ello utilizando arreglos unidimensionales y control de flujo.

Tema: Sistema de Gestión de Calificaciones de Estudiantes

Descripción del Ejercicio:

En este ejercicio, vamos a crear un programa que simule un sistema de gestión de calificaciones de estudiantes utilizando arreglos unidimensionales (listas). El programa permitirá al usuario ingresar los nombres de los estudiantes y sus calificaciones, almacenarlos en una lista de diccionarios y luego calcular estadísticas como el promedio de las calificaciones, la cantidad de aprobados y reprobados, así como los estudiantes con las mejores y peores calificaciones.

Este ejercicio cubre una amplia gama de conceptos, como:

- Arreglos unidimensionales (listas).
- Condicionales para validar la entrada.
- Bucle for para iterar sobre listas.
- Uso de break y continue para control de flujo.
- Uso de listas de diccionarios para almacenar datos complejos.

Requerimientos del Programa:

Profundiza más

1. **Ingreso de Estudiantes y Calificaciones:** El usuario ingresará el nombre y las calificaciones de cada estudiante. El sistema validará que las calificaciones estén en un rango válido (de 0 a 10).
2. **Cálculo de Promedio y Clasificación:** El sistema calculará el promedio de las calificaciones de cada estudiante y los clasificará en "Aprobado" (promedio >= 6) o "Reprobado" (promedio < 6).
3. **Estadísticas:** El sistema mostrará:
    - El promedio general de todos los estudiantes.
    - La cantidad de estudiantes aprobados y reprobados.
    - El estudiante con la mejor calificación y el de la peor calificación.
4. **Opción de Salir:** El usuario podrá salir del programa en cualquier momento.
5. **Control de Errores:** El programa validará que las calificaciones ingresadas estén dentro del rango esperado y que los datos sean consistentes.

Código:

```
# Sistema de gestión de calificaciones de estudiantes

# Listas para almacenar los datos
estudiantes = []  # Lista para almacenar nombres de estudiantes
calificaciones = []  # Lista para almacenar calificaciones de estudiantes

# Bucle principal para la entrada de datos
continuar = True  # Variable de control para el bucle

while continuar:
    # Ingreso de información del estudiante
    nombre = input("Ingrese el nombre del estudiante: ")

    # Validación de nombre no vacío
    if nombre.strip() == "":
        print("El nombre no puede estar vacío. Intente nuevamente.")
        continue  # Si el nombre está vacío, se omite esta iteración

    # Ingreso de las calificaciones
    while True:
        try:
            calif = input(f"Ingrese las calificaciones de {nombre}
separadas por coma: ").split(',')
            # Convertir las calificaciones a flotantes
            calif = [float(x) for x in calif]
```

Profundiza más

```
            # Verificar que las calificaciones estén en el rango de 0 a
10
            if any(x < 0 or x > 10 for x in calif):
                print("Las calificaciones deben estar entre 0 y 10.
Intente nuevamente.")
                continue
            break
        except ValueError:
            print("Error: Debe ingresar solo números separados por
coma.")

    # Guardar el nombre y las calificaciones
    estudiantes.append(nombre)
    calificaciones.append(calif)

    # Preguntar si se desea ingresar más estudiantes
    continuar_respuesta = input("¿Desea ingresar otro estudiante? (s/n):
").lower()
    if continuar_respuesta == "n":
        continuar = False  # Si la respuesta es no, salir del bucle

# Cálculo de estadísticas
if len(estudiantes) > 0:
    total_aprobados = 0
    total_reprobados = 0
    mejores_calif = float('-inf')
    peores_calif = float('inf')
    mejor_estudiante = ""
    peor_estudiante = ""
    suma_promedios = 0

    for i in range(len(estudiantes)):
        promedio = sum(calificaciones[i]) / len(calificaciones[i])
        suma_promedios += promedio

        # Clasificación del estudiante
        if promedio >= 6:
            total_aprobados += 1
        else:
            total_reprobados += 1
```

Profundiza más

```
        # Mejor y peor calificación
        if promedio > mejores_calif:
            mejores_calif = promedio
            mejor_estudiante = estudiantes[i]

        if promedio < peores_calif:
            peores_calif = promedio
            peor_estudiante = estudiantes[i]

    # Promedio general
    promedio_general = suma_promedios / len(estudiantes)

    # Mostrar estadísticas
    print(f"\nPromedio general de todos los estudiantes:
{promedio_general:.2f}")
    print(f"Estudiantes aprobados: {total_aprobados}")
    print(f"Estudiantes reprobados: {total_reprobados}")
    print(f"Estudiante con la mejor calificación: {mejor_estudiante} con
{mejores_calif:.2f}")
    print(f"Estudiante con la peor calificación: {peor_estudiante} con
{peores_calif:.2f}")
else:
    print("No se ingresaron estudiantes.")
```

Explicación del Código:

1. **Ingreso de Datos:**
   - El usuario ingresa el nombre y las calificaciones de los estudiantes. Se realizan validaciones para asegurarse de que el nombre no esté vacío y que las calificaciones estén dentro del rango de 0 a 10. En caso de errores, el programa vuelve a solicitar los datos.
2. **Cálculo del Promedio:**
   - El programa calcula el promedio de las calificaciones de cada estudiante utilizando la función sum() y luego muestra las estadísticas correspondientes.
3. **Estadísticas Generales:**
   - El programa muestra el promedio general de las calificaciones, la cantidad de estudiantes aprobados y reprobados, y las calificaciones del estudiante con mejor y peor rendimiento.

4. **Uso de continue:**
    - Se utiliza continue para omitir iteraciones si el nombre está vacío o si las calificaciones no son válidas.
5. **Control de Flujo con if y else:**
    - Se usa if para realizar las verificaciones de las calificaciones y de la condición de si el estudiante debe ser aprobado o reprobado.

Este ejercicio integra múltiples conceptos como el uso de listas para almacenar datos, el uso de bucles while y for, y el manejo de condicionales para gestionar entradas de usuario. Además, refuerza la validación de datos y el cálculo de estadísticas, lo que mejora las habilidades de los estudiantes en la programación estructurada y en la manipulación de datos.