Profundiza más

Recurso de Profundización

Clase 15: Gestión de Calificaciones de Estudiantes con Funciones, Arreglos, Archivos y Manejo de Excepciones

Autor: Damián Nicolalde Rodríguez

Objetivo del Documento:

Desarrollar un programa que lea datos de estudiantes desde un archivo de texto, procese sus calificaciones utilizando funciones, calcule su promedio, clasifique el rendimiento, y actualice el archivo de manera robusta usando estructuras de control, arreglos, manejo de errores y principios de programación modular.

Descripción Detallada del Problema:

Tenemos un archivo estudiantes.txt que contiene datos separados por punto y coma. Cada línea representa un estudiante con su nombre y sus calificaciones. El objetivo es:

1. Leer este archivo línea a línea.
2. Procesar las calificaciones de cada estudiante.
3. Calcular su promedio.
4. Clasificar su rendimiento: Excelente (>=9), Aprobado (>=6), Reprobado (<6).
5. Imprimir un informe y escribir los resultados en un nuevo archivo resultados.txt.

```python
def leer_estudiantes(ruta_archivo):
    """
    Lee los datos de estudiantes desde un archivo.
    Cada línea tiene el formato: nombre;nota1,nota2,...
    """
    try:
        with open(ruta_archivo, "r", encoding="utf-8") as f:
            lineas = f.readlines()
            return [linea.strip() for linea in lineas]
    except FileNotFoundError:
        print("Error: El archivo no fue encontrado.")
        return []
    except Exception as e:
        print("Ocurrió un error inesperado:", e)
```

Profundiza más

```
        return []

def procesar_calificaciones(linea):
    """
    Procesa una línea y devuelve nombre, lista de notas, promedio y
estado.
    """
    try:
        nombre, notas_str = linea.split(";")
        notas = list(map(int, notas_str.split(",")))
        promedio = sum(notas) / len(notas)

        if promedio >= 9:
            estado = "Excelente"
        elif promedio >= 6:
            estado = "Aprobado"
        else:
            estado = "Reprobado"
        return nombre, notas, promedio, estado
    except ValueError:
        print(f"Error procesando línea: {linea}")
        return None

def guardar_resultados(estudiantes, ruta_salida):
    """
    Guarda los resultados de los estudiantes en un nuevo archivo.
    Además, los imprime en consola.
    """
    try:
        with open(ruta_salida, "w", encoding="utf-8") as f:
            for estudiante in estudiantes:
                if estudiante:
                    nombre, notas, promedio, estado = estudiante
                    notas_str = ",".join(map(str, notas))
                    linea_resultado = f"{nombre};{notas_str};Prom:
{promedio:.2f};{estado}"
                    f.write(linea_resultado + "\n")
                    print(linea_resultado)  # <--- imprime en consola
        print(f"\nResultados guardados en {ruta_salida}")
    except IOError as e:
        print("Error al guardar los resultados:", e)
```

Profundiza más

```python
def main():
    ruta_entrada = "estudiantes.txt"
    ruta_salida = "resultados.txt"

    datos = leer_estudiantes(ruta_entrada)
    resultados = [procesar_calificaciones(linea) for linea in datos]
    guardar_resultados(resultados, ruta_salida)

# Ejecutar el programa
main()
```

Explicación Detallada:

1. **Lectura de archivos:** Se usa open(..., "r") y readlines() para leer el contenido del archivo, aplicando strip() para limpiar los saltos de línea.
2. **Procesamiento modular:** Cada paso está encapsulado en funciones específicas (leer_estudiantes, procesar_calificaciones, guardar_resultados) para cumplir con los principios de programación estructurada y reutilización.
3. **Estructuras de control:** Se usan condicionales (if, elif, else) y bucles (for) para iterar y clasificar los promedios.
4. **Arreglos:** Las calificaciones están representadas como listas (notas), manipuladas usando map() y sum().
5. **Excepciones:** Se capturan errores comunes como FileNotFoundError y ValueError, asegurando robustez.
6. **Salida formateada:** El archivo resultados.txt contiene los resultados organizados y legibles.

Salida Esperada en **resultados.txt**:

```
Juan;8,9,7,6,8;Prom: 7.60;Aprobado
Ana;10,9,9,10,9;Prom: 9.40;Excelente
Luis;5,6,4,5,5;Prom: 5.00;Reprobado
Marta;9,8,9,7,9;Prom: 8.40;Aprobado
Carlos;6,6,6,6,6;Prom: 6.00;Aprobado
Elena;4,5,5,4,3;Prom: 4.20;Reprobado
Pablo;10,10,9,10,9;Prom: 9.60;Excelente
Lucía;7,8,6,7,8;Prom: 7.20;Aprobado
Diego;3,2,4,3,5;Prom: 3.40;Reprobado
Valeria;9,9,9,10,8;Prom: 9.00;Excelente
```