

Análisis al código (previo intervención o arreglo de este mismo)

El código lo que hace es añadir estudiantes y con estos sus promedios (los guarda en listados creados), creando listados de estos junto sus calificaciones. El código da la opción de agregar, mostrar estudiantes, calcular promedio de notas, muestra la nota más alta y opción a salirse del menú (programa).

Empieza pidiendo agregar a un estudiante, después de esto su calificación o nota, ya dependiendo de esto y si se añaden más hace promedio de notas y todo lo demás mencionado.

Análisis del Programa.

El código permitía modularizar funciones como agregar estudiantes, listado de estudiantes, calcular promedios etc. También el hecho de solo usar unas variables globales facilita el uso y alcance en el programa, es mejor para la memoria durante el uso del programa, tal el caso con el uso de las variables globales que crean los listados para guardar la información de los nombres y las calificaciones; con las variables locales son de alcance limitado, necesario para realizar las funciones específicas como añadir estudiantes y tal cual nos ayudan en el guardado (o modulación). Variables locales para datos que solo se necesitan dentro de un método o bloque específico (creados para las funciones añadidas) y variables globales para datos que deben ser accesibles desde múltiples métodos o clases (listado de nombres y calificaciones).

- ¿Qué datos deben de ser accesibles para todo el programa? R// Los nombres y calificaciones guardadas o agregadas al programa en las variables globales.
- ¿Qué datos solo son necesarios dentro de una función específica? R// Van dentro de las variables locales, cosas como el promedio (sacado a calificaciones).

Se hizo lo siguiente:

- 1. Agregar estudiantes:** Se realiza al principio en la opción 1 del menú y se pueden convertir en función tal como AgregarEstudiante (), en nuestro caso addEstudiante (nombre de la función), donde solo llamamos al listado a guardar los datos ingresados de nombre y calificaciones.
 - 2. Lista de estudiantes:** Función creada para llamar al listado y colocar los nombres de los estudiantes ingresados.
 - 3. Promedio de Calificación:** Con los datos guardados se creó una función donde usando estos se calcule la media y lo guarde en la variable promedio para después mostrarlo.
 - 4. Calificación mas alta:** Función creada para buscar entre el listado el nombre del estudiante con la nota más alta, usando unos arrays (o parecido a vectores).
-
- ¿Qué datos necesitan ser compartidos entre múltiples funciones? R// estudiantes y calificación.
 - ¿Qué datos solo son relevantes dentro de una función? R// nombre, calificación (en addEstudiante), suma, promedio (en promCalificacion), maxCalificacion, estudianteMax (en caliEstudiante), opcion (en Main).

Responde las siguientes preguntas:

1. ¿Qué ventajas tiene dividir el código en funciones?

R// me ofrece una mejor organización al modular el código, me permite dividirlo y manejarlo mejor por partes, me permite reutilizar código y es más fácil corregir errores.

2. ¿Por qué es importante limitar el uso de variables globales?

R// Para tener menos riesgo en errores inesperados ya que se pueden modificar desde cualquier parte del programa, es poco más difícil al depurar y en mantenimiento dificulta un poco por sus interacciones en el programa.

3. ¿Cómo se puede mejorar la legibilidad del código?

R// Con nombres más descriptibles y comentarios que ayuda un montón al dar la idea de lo que se hace en el código, ayuda a entender a los demás y recordar a uno mismo, además es bueno mantener una estructura clara y funciones pequeñas.

Trabajo en pareja:

- Estalin Alejandro Godoy Campos 0905-24-5051
- Deivid Alberto Guerra Carpio 0905-24-23552