

PROGRAMACIÓN 2

Subprogramas

Rafael Rivas Estrada

SUBPROGRAMAS

Agenda

- Conceptos.
- Recomendaciones sobre el:
 - Diseño.
 - Codificación.
 - Revisión.
 - Justificación.
- Cohesión y acoplamiento.

Conceptos

- ¿Qué es un subprograma?
- Tipos:
 - Procedimientos
 - Funciones
- Tipos de Variables:
 - Variables globales.
 - Variables locales.

Conceptos

- Tipos de parámetros
 - Parámetros formales.
 - Parámetros actuales.
- Comunicación por medio de :
 - Pase por valor (Lenguajes C y C++).
 - Pase por referencia. (Lenguaje C++)

Recomendaciones de diseño

- Definir el problema que el subprograma debe resolver.
- Definir la entradas.
- Definir las salidas incluyendo las variables globales aceptadas.
- Manejo de errores en el subprograma.
- ¿Cómo probar el subprograma?

Recomendaciones en la codificación

- Escribir la declaración
- Colocar los comentarios
- Colocar el código debajo del comentario.
- Completar el código.

Revisión del código

- Revisar el subprograma en forma manual.
- Compilar el subprograma de forma independiente.
- Usar el depurador para revisar el subprograma.

Razones para crear un subprograma

- Reducir la complejidad.
- Evitar duplicar código.
- Limitar efectos de cambios.
- Ocultar secuencias de códigos.
- Aumentar el desempeño
- Ocultar estructuras de datos.

Razones para crear un subprograma

- Ocultar operaciones con apuntadores.
- Promover la reutilización de código.
- Aislard operaciones complejas.
- Crear un punto de control.
- Aislard el uso de subprogramas no estándezar.
- Simplificar operaciones lógicas complejas.
- ¿Aumentar la modularidad?

(Ver <http://www.stevemcconnell.com/cchqsub.htm>)

Cohesión y Acoplamiento

- ¿Qué es la cohesión en la programación ?
- ¿Qué es el acoplamiento en subprogramas ?

Cohesión

Cohesión fuerte:

- Cohesión funcional.
- Cohesión secuencial.
- Cohesión comunicacional.
- Cohesión temporal.

Cohesión

Cohesión débil:

- Cohesión procedural
- Cohesión lógica.
- Cohesión coincidencial.

Acoplamiento

- Tamaño de la conexión.
- Intimidad.
- Visibilidad.
- Flexibilidad.

Referencias

- Steve McConnell. “Code Complete: A Practical Handbook of Software Construction”. Redmond, Wa.: Microsoft Press, 880 pages, 1993.
- Cohesión:
http://en.wikipedia.org/wiki/Cohesion_%28computer_science%29
- Acoplamiento:
http://en.wikipedia.org/wiki/Coupling_%28computer_science%29