

Programación modular

1. Introducción

- Programación estructurada: todo el código dentro de un mismo bloque
- Definición de módulos programados de forma independiente y compilados por separado
- Se usa programación modular y estructurada al mismo tiempo
- Prog. Modular descompone problema en subproblemas (refinamiento) llamados módulos
- Módulos se comunican entre ellos mediante paso de parámetros

2. Programación modular

- Objetivos módulo:
 - Cada uno realiza una tarea, un identificador diferente. Se pasan parámetros
 - Módulos tienen un único punto de entrada y de salida
 - Unos módulos pueden llamar a otros

Ventajas

- Menor coste de desarrollo. Trabajo simultáneo en paralelo
- Reutilización de código. Facilidad de probar y mantener
- No es necesario conocer el funcionamiento interior del módulo
- Módulos se pueden agrupar en librerías o bibliotecas para poder ser usados en otros

3. Diseño descendente TOP-DOWN

El disseny descendent TOP-DOWN és una **metodologia de disseny de programari** que implica la descomposició d'un problema complex en subproblemes més simples i específics en successius nivells, fins que cada subproblema pugui ser resolt amb una solució clara i concisa. Això permet la reducció de la mida dels subproblemes i, per tant, la minimització de la duplicitat de codi.

Cada mòdul en el disseny descendent TOP-DOWN ha de tenir una funció el més específica possible i ha de ser reutilitzable en altres projectes, si és possible. El disseny descendent TOP-DOWN també implica l'agrupació de mòduls comuns segons la seva funcionalitat o temàtica, el que facilita la gestió i la resolució de problemes.

4. Clasificación de módulos

Según el módulo que lo invoca

- Interno: están en el mismo fichero que el módulo que lo invoca
- Externo: está en un fichero diferente

Según retorno de valores

- Funciones: deben devolver un valor
- Definición, Declaración, Llamada
 - Procedimientos:
 - No devuelven ningún valor. Parámetros de entrada, salida, o E/S.

- Permiten devolver más de un valor
- Definición, declaración, llamada

Según cuando ha sido desarrollado

- De programa: se ha desarrollado en el programa actual
- De librería: ha sido desarrollado previamente y está contenido en ficheros de librerías

Según el número de módulos que realizan la llamada

- Subprograma: es invocado por un solo módulo
- Rutina o subrutina: es invocado por diversos módulos

5. Estructura modular

- Se puede representar mediante diagrama basado en jerarquía, descomposición e independencia de módulos
 - Estructura secuencial
 - Estructura selectiva
 - Estructura repetitiva

6. Diseño de funciones

Parámetros

- Entrada, salida, entrada y salida
- Paso por copia, por valor, por referencia

Ámbito de identificadores

- Identificador puede ser variable, función, procedimiento, constante, etv.
- Identificadores globales y locales
- Recomendaciones

- Independencia funcional

7. Recursividad

- En el cuerpo de un subprograma hay una llamada al propio subprograma.
- Condiciones: necesita una condición de parada. Cada iteración debe utilizar cada vez una versión del problema más reducida.
- En algoritmos o en datos
- Simple y múltiple

- Directa e indirecta
- Ventajas: solución simple y elegante a problemas complejos. No es necesario definir secuencia de pasos
- Inconvenientes: sobrecarga puede producir desbordamiento de la pila. Ineficiencia de algunos algoritmos

8. Librerías

Les llibreries són conjunts de codi predefinit que inclouen procediments i funcions compilades, documentades i provades, les quals poden ser invocades des d'un altre programa.

Tipos de librerías

Hi ha diferents tipus de llibreries, com ara les de codi font, les compilades i les d'enllaç dinàmic, que poden ser compartides per diverses aplicacions.

L'enllaç dinàmic és un mecanisme utilitzat en el desenvolupament de programari que permet que diverses aplicacions comparteixin una mateixa llibreria en temps d'execució, en lloc de tenir una còpia individual de la llibreria per a cada aplicació.

Això vol dir que quan una aplicació necessita utilitzar una determinada funció o procediment que es troba en una llibreria compartida, en lloc d'incloure tot el codi de la llibreria al seu propi codi font, simplement fa referència a la llibreria. En el moment de l'execució, el sistema operatiu carrega la llibreria i totes les aplicacions que la necessiten la poden utilitzar.

Un dels avantatges de l'enllaç dinàmic és que redueix la mida dels executables de les aplicacions, ja que no cal incloure tot el codi de la llibreria a cadascuna. A més, també s'aprofita millor la memòria del sistema, ja que diverses aplicacions poden utilitzar la mateixa llibreria alhora. D'altra banda, si es produeix una actualització a la llibreria compartida, totes les aplicacions que la utilitzen es beneficien automàticament de la millora sense necessitat d'actualitzar individualment cadascuna.

Características

Ejemplos (Java, C)

Ejemplos de librerías:

- C: `stdio.h`, `stdlib.h`
- Java: `java.io`, `java.net`