



Técnicas generales de diseño de software

I. Diseño procedimental

1. Diseño funcional descendente

1.1. Técnica del refinamiento progresivo

Aplicación, a la fase de diseño, del concepto de refinamientos sucesivos [Wirth]. Consiste en plantear la aplicación como única operación global, e irla descomponiendo en operaciones más sencillas, detalladas y específicas. En cada nivel de refinamiento, las operaciones identificadas se asignan a módulos separados.

La construcción de programas mediante refinamientos sucesivos proviene de la metodología de programación estructurada [Dijkstra]. Esta metodología de programación consiste en la utilización sólo de estructuras de control claras y sencillas, con esquemas de ejecución con un único punto de inicio y un único punto final.

1.2. Técnica de programación estructurada de Jackson

Aplicación, a la fase de diseño, de la metodología de programación estructurada [Dijkstra]. Esta metodología de programación consiste en la utilización sólo de estructuras de control claras y sencillas, con esquemas de ejecución con un único punto de inicio y un único punto final. Dichas estructuras son la **secuencia**, la **selección** y la **iteración**.

La técnica de Jackson consiste en construir las estructuras del programa de forma similar a las estructuras de los datos de entrada-salida que se manejan.

1.3. Técnica de diseño estructurado [Yourdon]

Técnica de diseño complementaria del 'análisis estructurado' (utilización de DFD). Consiste en:

1. Utilización de DFD para la descripción del funcionamiento de la aplicación.
2. Con los primeros niveles de la descomposición en DFD, se construye un único diagrama en el que se incluyen los procesos implicados y se prescinde de los elementos de almacenamiento. En este último diagrama, se realiza un análisis para identificar o bien un único flujo global (flujo de transformación) o bien uno o varios puntos en los que el flujo global se bifurca (flujo de transacción).



3. Según el patrón identificado en el análisis anterior, se construye el diagrama de diseño mediante la notación de los diagramas de estructura. Para ello, a partir de los DFD iniciales, se asignan procesos o grupos de procesos a los módulos de diseño y se establece una jerarquía o estructura de control en función del mencionado patrón identificado en el punto anterior (transformación o transacción).

2. Diseño con abstracciones

2.1. Descomposición modular con abstracciones

Consiste en dedicar módulos separados a la realización de cada tipo abstracto de datos y cada función importante. Se emplea la notación de los diagramas de bloques jerarquizados; en los que se representan las relaciones de uso y la jerarquía se establece de arriba hacia abajo ('los de arriba usan a los de abajo').

Se puede aplicar de forma descendente (ampliación del refinamiento progresivo con las abstracciones) o ascendente (ampliación de primitivas hacia abstracciones de nivel superior).

2.2. Método de Abbott

Metodología para identificar los elementos abstractos que formarán parte del diseño. Las abstracciones obtenidas se complementan y se organizan en un diagrama de diseño. El método se suele aplicar a las descripciones del análisis (formales o informales).

3. Diseño orientado a objetos

En esencia es similar al diseño con abstracciones pero, además de la relaciones de uso y agregación, hay que considerar la herencia y el polimorfismo. En la descomposición modular del sistema, cada módulo contiene la descripción de una clase o de varias clases de objetos relacionadas entre sí. Se suele utilizar un diagrama de estructura para describir la arquitectura del sistema y las relaciones de uso y diagramas ampliados de las clases y objetos en las que se representan las distintas relaciones entre los datos (herencia, polimorfismo, etc.).



II. Diseño de datos

1. Bases de datos relacionales

1.1. Formas normales

1.2. Diseño de entidades

1.3. Relaciones de asociación

1.4. Relaciones de agregación

1.5. Relaciones de herencia

1.6. Índices

2. Bases de datos de objetos