7.2 Diseño a nivel de rutina

Código completo

Cohesión:

La idea de cohesión fue introducida en un documento de Wayne Stevens, Glenford Myers y Larry Constantine (1974).

Otros conceptos más modernos, incluyendo la abstracción y la encapsulación, tienden a generar más información a nivel de clase.

Para las rutinas, la cohesión se refiere a qué tan estrechamente están relacionadas las operaciones de una rutina.

Entender los conceptos es más importante que recordar términos específicos. Utilice los conceptos como ayuda para pensar en cómo hacer que las rutinas sean lo más coherentes posible.

La cohesión funcional es el tipo de cohesión más fuerte y mejor, y se produce cuando una rutina realiza una sola operación. Ejemplos de rutinas altamente cohesivas incluyen sin (), GetCustomerName (),

La evaluación de su cohesión asume que las rutinas hacen lo que sus nombres dicen que hacen: si hacen algo más, son menos cohesivas y mal nombradas.

Varios otros tipos de cohesión normalmente se consideran menos que ideales:

La cohesión secuencial existe cuando una rutina contiene operaciones que deben realizarse en un orden específico, que comparten datos de un paso a otro y que no constituyen una función completa cuando se realizan juntas.

¿Cómo haría que la rutina sea funcionalmente cohesiva? Crearías rutinas separadas.

La cohesión comunicacional se produce cuando las operaciones de una rutina utilizan los mismos datos y no están relacionadas de ninguna otra manera.

Si una rutina imprime un informe de resumen y luego reinicializa los datos de resumen que se le han pasado, la rutina tiene cohesión comunicacional: las dos operaciones están relacionadas solo por el hecho de que utilizan los mismos datos.

Dividir las operaciones en rutinas individuales. El primero imprime el informe. El segundo reinicializa los datos, cerca del código que crea o modifica los datos. Llame a ambas rutinas de la rutina de nivel superior que originalmente llamó la rutina comunicativamente cohesiva.

La cohesión temporal ocurre cuando las operaciones se combinan en una rutina porque todas se realizan al mismo tiempo.

Los ejemplos típicos serían Startup (), CompleteNewEmployee () y Shutdown (). Algunos programadores consideran inaceptable la cohesión temporal.

Quedará claro que el objetivo de la rutina es orquestar actividades en lugar de realizarlas directamente.

Los tipos restantes de cohesión son generalmente inaceptables: dan como resultado un código mal organizado, difícil de depurar y difícil de modificar.

La cohesión procedimental se produce cuando las operaciones de una rutina se realizan en un orden específico.

Un ejemplo es una rutina que obtiene el nombre de un empleado, luego una dirección y luego un número de teléfono. El orden de estas operaciones es importante solo porque coincide con el orden en el que se solicitan los datos al usuario en la pantalla de entrada.

Para lograr una mejor cohesión, coloque las operaciones separadas en sus propias rutinas.

La cohesión lógica ocurre cuando varias operaciones se integran en la misma rutina y una de las operaciones es seleccionada por una bandera de control que se pasa. Se llama cohesión lógica porque el flujo de control o "lógica" de la rutina es lo único que une las operaciones. juntos, están todos juntos en una gran declaración if o case.

Las operaciones no están relacionadas, un mejor nombre podría ser "cohesión ilógica".

Un ejemplo sería una rutina InputAll () que ingresa los nombres de los clientes, la información de la tarjeta de tiempo de los empleados o los datos de inventario, dependiendo de una marca que se pase a la rutina.

Si las operaciones usan parte del mismo código o comparten datos, el código debe moverse a una rutina de nivel inferior y las rutinas deben empaquetarse en una clase.

Usualmente está bien, sin embargo, crear una rutina lógicamente cohesiva si su código consiste únicamente en una serie de declaraciones if o case y llamadas a otras rutinas. En tal caso, si la única función de la rutina es enviar comandos y no realiza el procesamiento por sí misma, ese es generalmente un buen diseño. El término técnico para este tipo de rutina es "controlador de eventos". Un controlador de eventos se utiliza a menudo en entornos interactivos como el Apple Macintosh, Microsoft Windows y otros entornos GUI.

Referencia cruzada

Aunque la rutina podría tener una mejor cohesión, un problema de diseño de nivel superior es si el sistema debe utilizar una declaración de caso en lugar de polimorfismo. Para obtener más información sobre este tema, consulte "Reemplazar condicionales con polimorfismo (especialmente declaraciones de casos repetidas)" en Refactorizaciones específicas.

La cohesión coincidente ocurre cuando las operaciones en una rutina no tienen una relación discernible entre sí. Otros buenos nombres son "sin cohesión" o "cohesión caótica".

Resumen:

Ninguno de estos términos es mágico o sagrado.

Aprende las ideas en lugar de la terminología.

Casi siempre es posible escribir rutinas con cohesión funcional, así que concentre su atención en la cohesión funcional para obtener el máximo beneficio.