Module

Ran Liao

May 22, 2019

Definition

A lexically contiguous sequence of program statements, bounded by boundary elements, with an aggregate identifier.

Cohesion - the higher the better

Cohesion represents the degree of interaction within a module.

1. **Coincidental Cohesion**

   A module has coincidental cohesion if it performs multiple, completely unrelated actions.

2. **Logical Cohesion**

   A module has logical cohesion when it performs a series of related actions, one of which is selected by the calling module.

3. **Temporal Cohesion**

   A module has temporal cohesion when it performs a series of actions related in time.

4. **Procedural Cohesion**

   A module has procedural cohesion if it performs a series of actions related by the procedure to be followed by the product.

5. **Communicational Cohesion**

   A module has communicational cohesion if it performs a series of actions related by the procedure to be followed by the product, but in addition all the actions operate on the same data.

6. **Functional Cohesion**

   A module with functional cohesion performs exactly one action.

7. **Informational Cohesion**

   A module has informational cohesion if it performs a number of actions, each with its own entry point, with independent code for each action, all performed on the same data structure.

Coupling - the lower the better

Coupling represents the degree of interaction between two modules.

1. **Content Coupling**

   Two modules are content coupled if one directly references contents of the other.

2. **Common Coupling**

   Two modules are common coupled if they have write access to global data.

3. **Control Coupling**

   Two modules are control coupled if one passes an element of control to the other.

4. **Stamp Coupling**

   Two modules are stamp coupled if a data structure is passed as a parameter, but the called module operates on some but not all of the individual components of the data structure.

5. **Data Coupling**

   Two modules are data coupled if all parameters are homogeneous data items (simple parameters, or data structures all of whose elements are used by called module).

Information Hiding - the higher the better

Ensure that implementation details are not visible outside the module in which they are declared.