

Coupling

- Degree of interdependence between software modules
 - A module can be any unit of software
 - Function
 - Class
 - Libraries
 - Databases
 - etc.

- tight or high coupling

- more interdependency

- a change in one module forces a change in the other

- testing is more difficult due to dependencies

- loose or low coupling

- less interdependency

- a change in one module causes little or no change in another

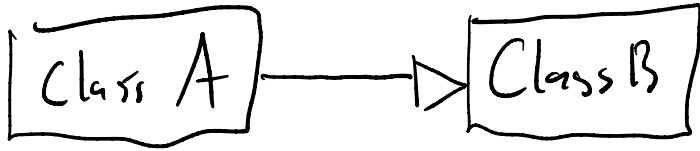
- easier to test modules as independent units

- types of coupling

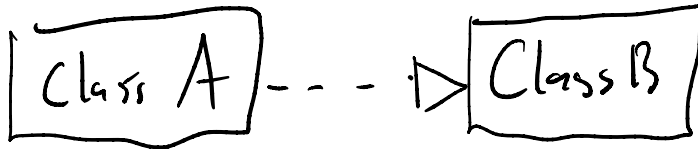
- Function or method name and parameters
- Message passing
- Data format (XML, JSON, custom format, etc.)
- Subclass / base class

Unified Modeling Language

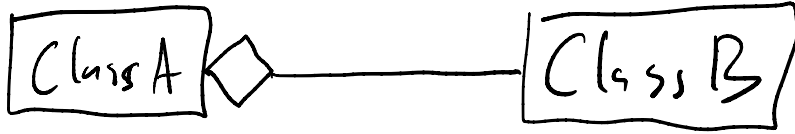
- Standard language for diagrams



Class B inherits from Class A



Class A is abstract



- Class A stores an object of type Class B, Class B can survive if the Class A object dies (stack allocated and the function returns, freed from the heap, or garbage collected)
- Aggregation



- Class A stores an object of type Class B, the Class B object dies if the Class A object dies
- Composition

