

CLASS AND RELATION DIAGRAMS

We will represent the patterns using the UML notation. Then, the patterns stay simples : we only use a subset of the UML notation. We also avoid the possible ambiguities.

Classes definition

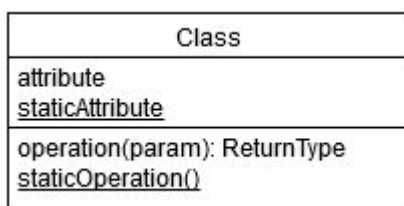
Classes

A class is described by his name, his attributes, and his operations.

When the attributes and the operations are static, they are underlined.

When it is typed, we indicate the type after a colon.

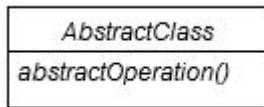
If we need to indicate restrictions to the attributes and operations, we can add + (public), = (protected) or - (private) behind the attribute or operation's name.



Abstract classes

If a class doesn't provide an implementation for any operation, it is abstract.

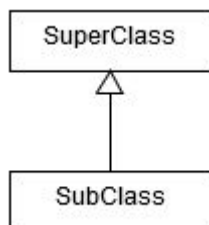
Abstract classes and operations are in italics.



Relations

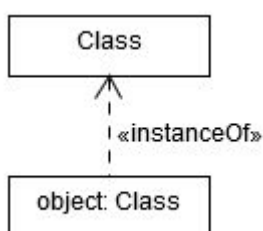
Inheritance

A superclass and his subclasses (or a super-interface and his sub-interfaces) have an inheritance relation.



Instances

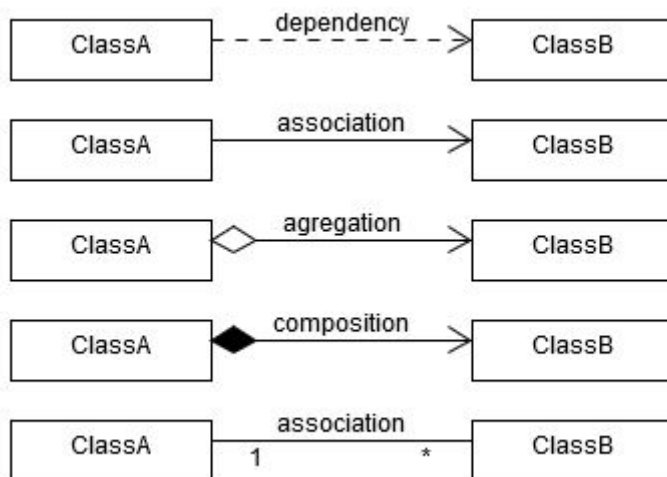
An instantiated object have an "instance of" relation with his class.



Relation between classes

- Dependency: ClassB is just used by ClassA (it can be an import). Most of the case, it is a local variable, or a parameter or return type.
- Association: the instances of these classes are associated. It can be "ClassA references ClassB". It is the same as "A person has an address".

- Agregation: it is a strong association, dissymetric with an object. It can be "ClassA have a ClassB". It is the same as "A car has tires".
- Composition: it is an agregation where the compounds can't exist by themselves. The lifecycles are strongly linked. It can be "ClassA is composed of ClassB". It is the same as "A person have a head". You can't have people without a head, and a head can't live by itself.
- We can add multiplicities on the link. It is "ClassA references n ClassB", and "ClassB is referenced by only 1 ClassA".



SEQUENCE DIAGRAMS

Here, the different objects are typed, it is instances. The sequence diagram shows object interactions arranged in time sequence. The different boxes (called Activation boxes) represent the processes who are performed by each object. Synchronous calls are represented by solid arrows, open arrows represent asynchronous messages, and dashed lines represent reply messages.

