# UML

UML – Unified Modified Language

* Structure Diagrams
* Behavioral Diagrams

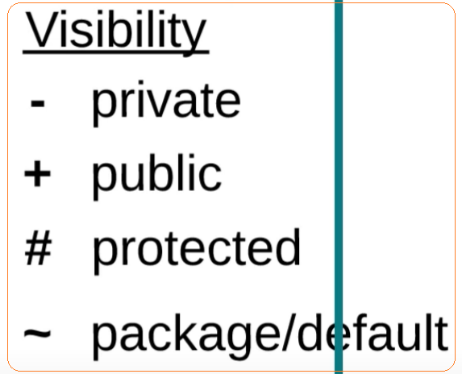
**Structured diagram:**

* ***Class diagram*** – it describes system. It displays classes, their attributes and relationship between classes. It’s static diagram

**Behavior diagram**:

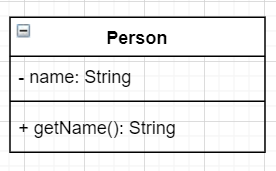
* ***Sequence diagram***. It’s dynamic diagram(describes life cycle of system)
* ***Collaboration diagram*** – visa verse of sequence diagram. It helps to display dependencies
* ***StateChart Diagram*** – describes states of system. It uses quite rare
* ***Deployment diagram*** -

# CLASS DIAGRAM



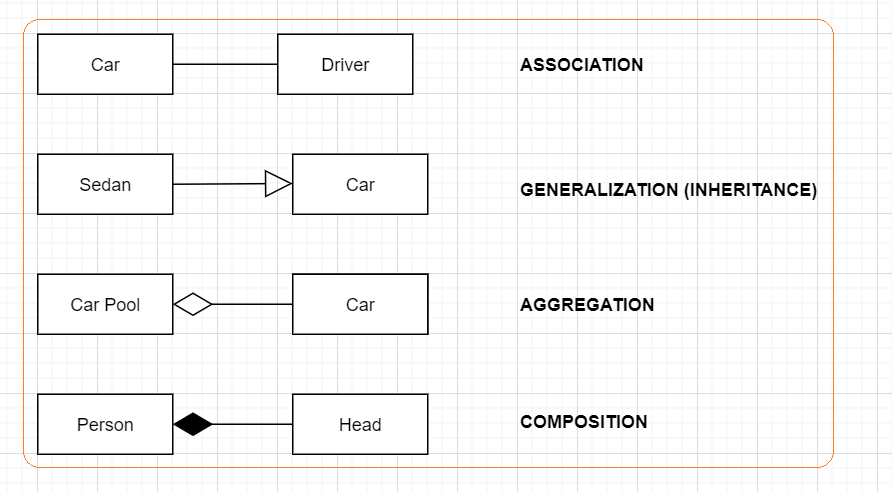
**Class Person**

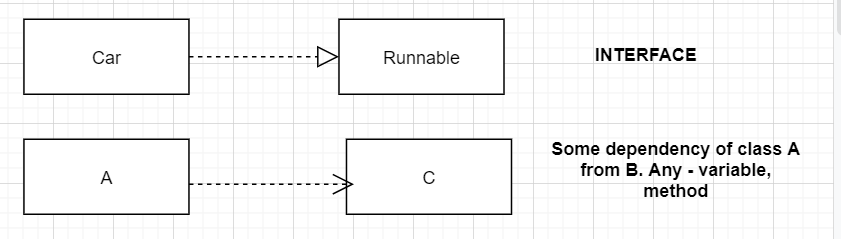
Field [name] is



**Interface**

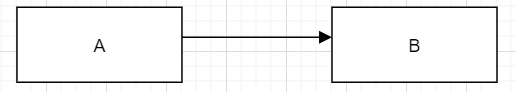






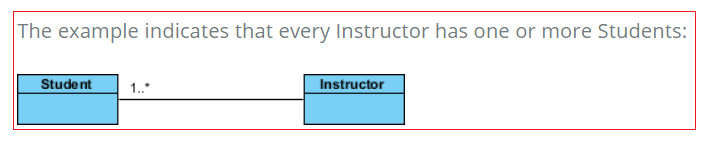
**Association** is a relationship between classes two and more classes. If two classes need to communicate each other we need to use [association]. In other words,

Class A depends on class B or class A calls methods of class B



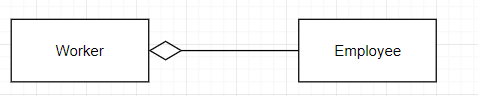
Association can be

* One way (odnonapravlennaya) “->”v
* Round way (dvunapravlennaya)

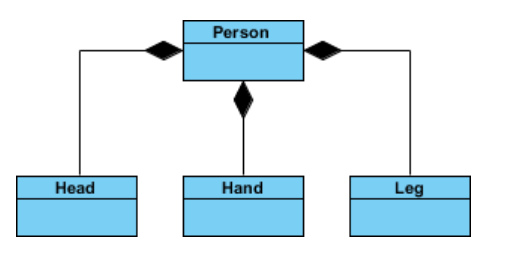


Note: association –is general term. [Aggregation] an [Composition] are part of [Association]

**Aggregation = “has a”:**



**Composition = “is a” (“or part of”):**



!!!! Difference between Aggregation and Composition

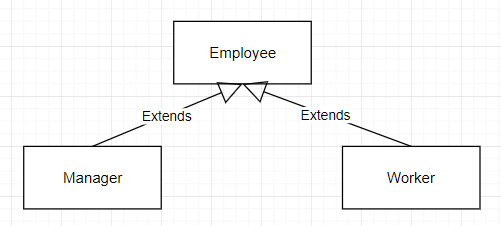
* *In Aggregation case child can* ***exists*** *without parent. For example, [Worker]-[Employee]*

*If you delete [Employee] class then [Worker] will still exists*

* *In Composition case child cannot* ***exist*** *without parent. For example, [Hand]-[Body].*

*If you delete [Body] then [Hand] will not exist*

**Generalization** – it indicates that child receives all attributes from parent



## Difference between Aggregation and Composition

