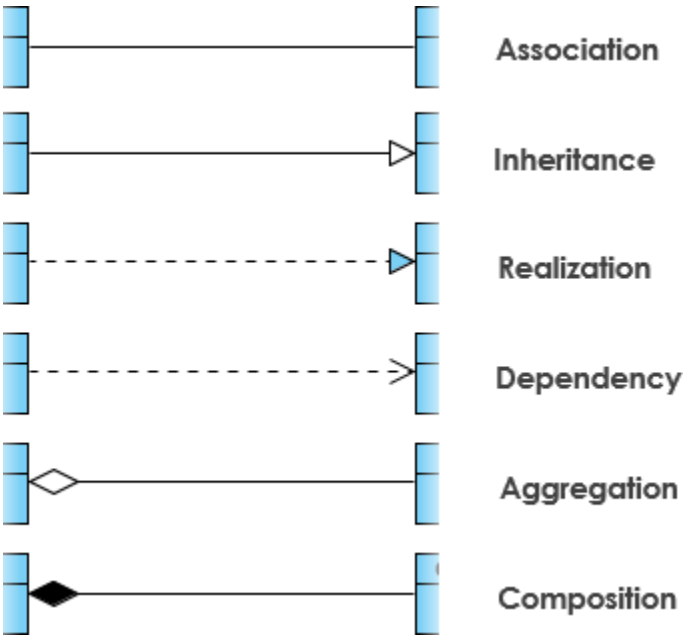
UML Class Diagrams

- ✓ UML (Unified Modeling Language) is a standardized family of notations for modeling and design of information systems.
- ✓ It was derived from various existing notations to provide a standard for software engineering.
- ✓ It comprises of several different diagrams representing different aspect of the system, and one of them being a Class Diagram that can be used for **data modeling.**
- ✓ Class diagrams are equivalent of ERDs in relational world and are mostly used to design classes in object-oriented programming languages (such as Java or C#).

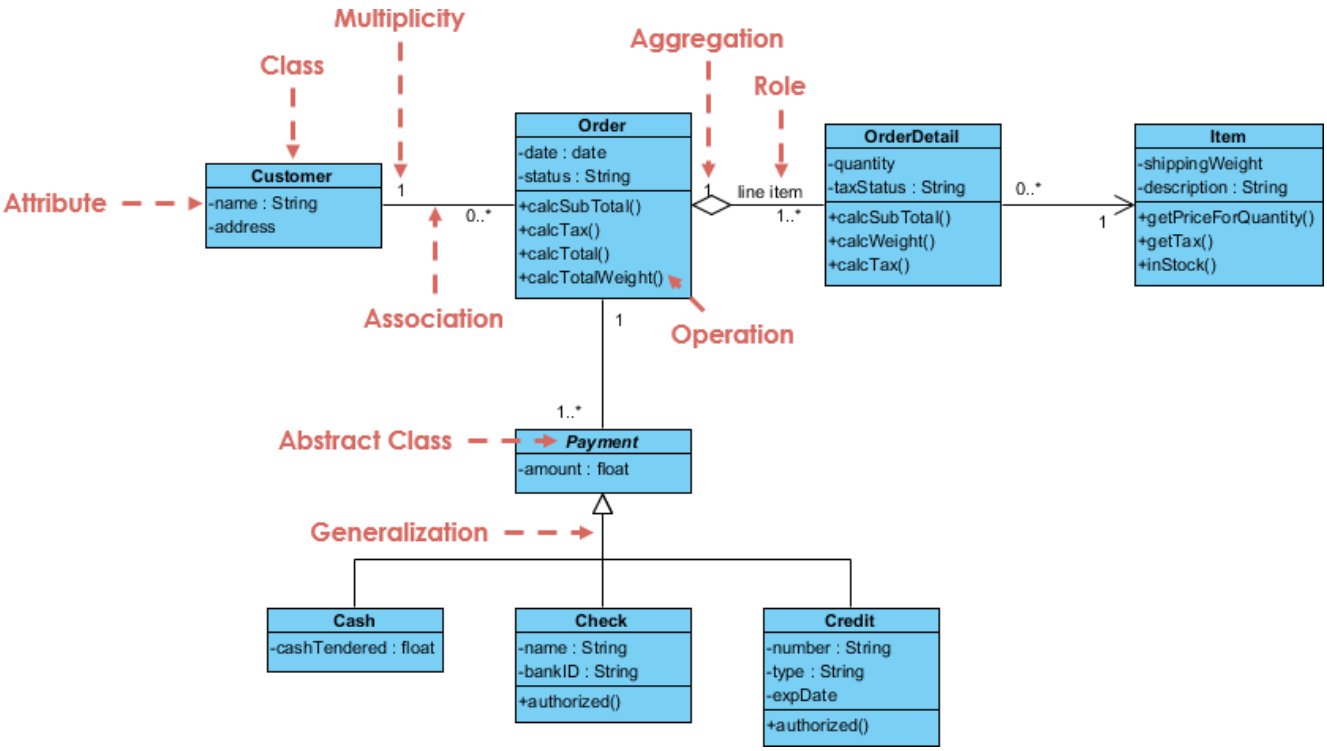
In class diagrams architects define:

- **Class Name** (equivalent of entity in relational world)
- **Attributes** (same as in an ERD) including data type
  - ✓ The attributes are written along with its visibility factors, which are public (+), private (-), and protected (#)
  - ✓ The accessibility of an attribute class is illustrated by the visibility factors.
- **Methods** representing its behavior (in relational world those would be stored procedures)
- **Relationships:** A class may be involved in one or more relationships with other classes. A relationship can be one of the following types:



- ✓ We can use class diagrams to design a tabular data (such as in RDBMS), but were designed and are used mostly for object-oriented programs (such as Java or C#).

Example:



How UML diagram assist in data modeling? [PU: 2019 fall]

UML diagrams can assist in data modeling in a number of ways, including:

- ✓ **Communicating the data model to stakeholders**. UML diagrams are a visual way to represent the data model, which can make it easier for stakeholders to understand. This is especially important for business users who may not be familiar with technical terms or concepts.
- ✓ **Identifying and documenting data requirements**. UML diagrams can be used to identify and document the data requirements for a system. This can help to ensure that the system meets the needs of the users and that the data is properly organized.
- ✓ **Modeling complex data relationships**. UML diagrams can be used to model complex data relationships, such as one-to-many, many-to-many, and self-referential relationships. This can help to ensure that the data is properly organized and that the relationships between the data elements are clear.
- ✓ **Generating code from the data model**. Some UML tools can generate code from the data model. This can save time and effort in the development process.

Here are some specific UML diagrams that can be used for data modeling:

Class diagrams: Class diagrams are used to represent the classes of objects in a system and the relationships between them. They can be used to model the data elements in a database.

Object diagrams: Object diagrams are used to represent the objects in a system at a specific point in time. They can be used to show how the data elements are related to each other.

Sequence diagrams: Sequence diagrams are used to show the interactions between objects in a system over time. They can be used to show how the data elements are used by the objects.

Overall, UML diagrams can be a valuable tool for data modeling. They can help to communicate the data model to stakeholders, identify and document data requirements, model complex data relationships, and generate code from the data model.

References:

https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/