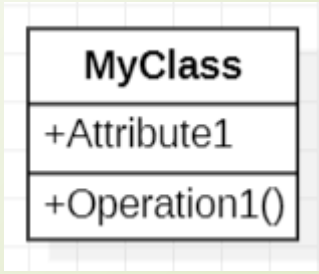
Class Diagram

Amir Dirin

Class Diagram

- A *class* is a description of a set of objects that share the same attributes, operations, relationships, and semantics.
- Graphically, a class is rendered as a rectangle, usually including its name, attributes, and operations in separate, designated compartments.



Class Attributes

- Attributes can be:

  +public

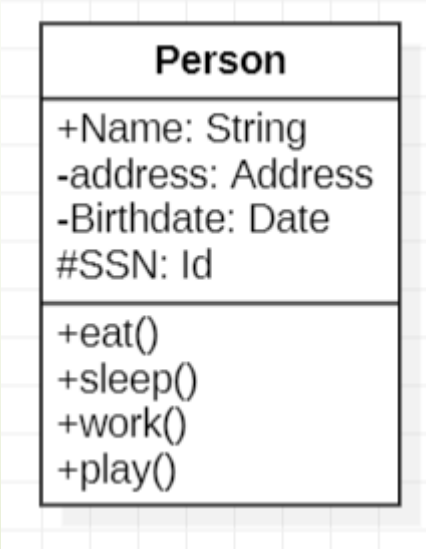
  #protected

  -private

  /derived

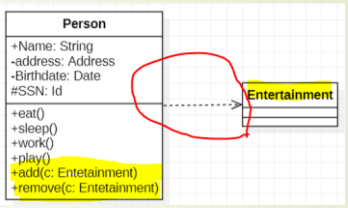
Operations

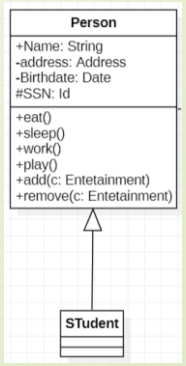
- You can specify an operation by stating its signature: listing the name, type, and default value of all parameters, and, in the case of functions, a return type.



Relationships

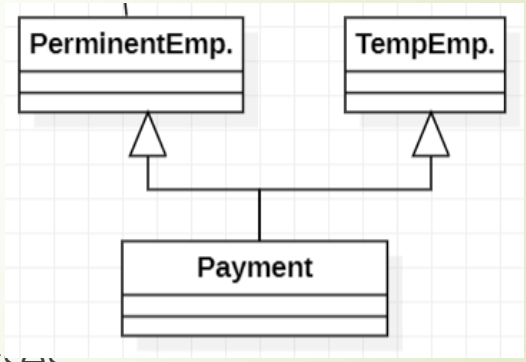
- In UML , object interaction are modeled as relationships.
  - Dependencies
    - Semantic relationship between two or more element
  - Generalizations
    - A generalizations connects a subclass to its superclass. It denoted an inheritance of attributes and behavious from superclass



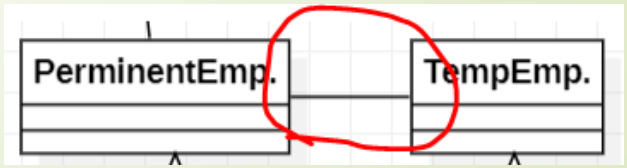


Relationships

- Generalization Relationships (Cont'd)
  - UML permits a class to inherit from multiple supperclass although some programminge.,g java does not allow

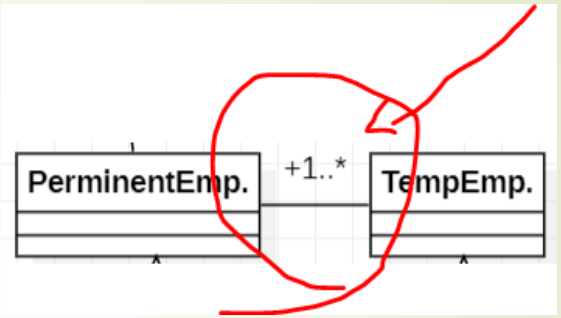


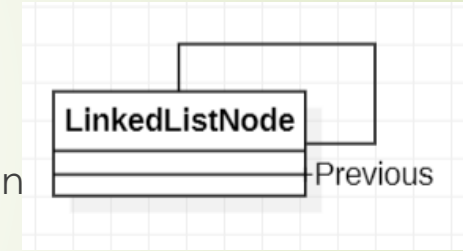
- Association Relationships
  - Communication between two classes

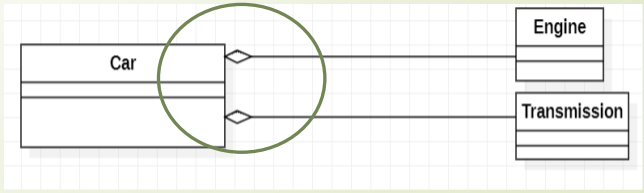


Relationships

- Association Relationships
  - A Class can have a self association
  - Aggregation
    - A whole-part relationship between an aggregate and consitutuen However, part can exist independently from aggregate.

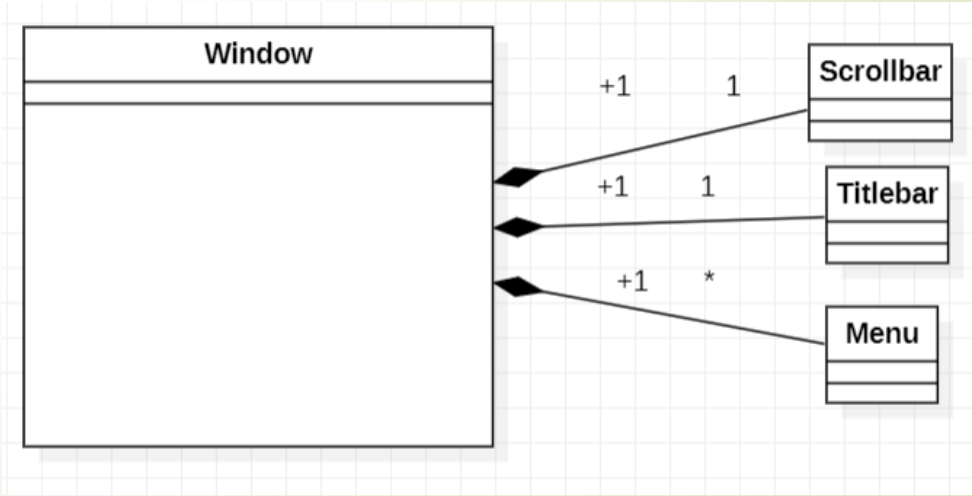






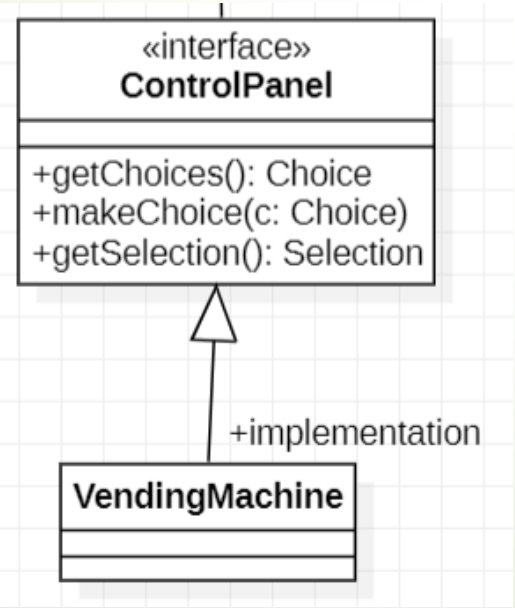
Association Relationships

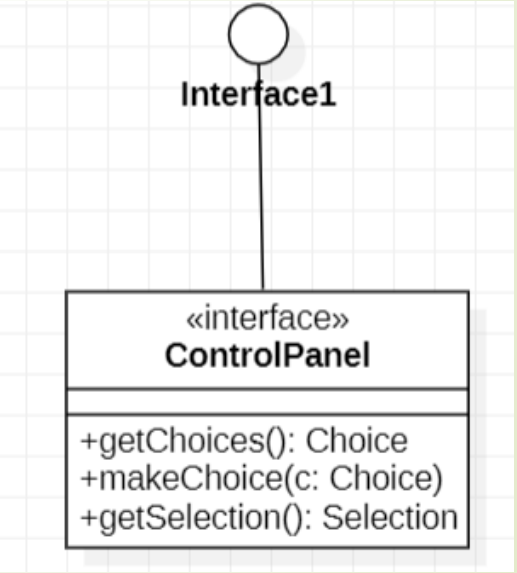
- A *composition* indicates a strong ownership and coincident lifetime of parts by the whole (*i.e.,* they live and die as a whole). Compositions are denoted by a filled-diamond adornment on the association.



Interface

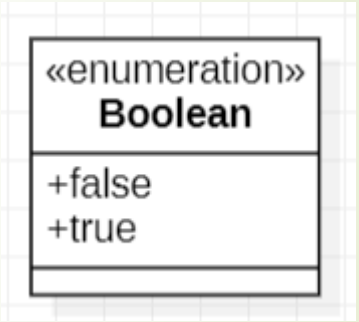
- An interface is a named set of operations that specifies the behaviou of objects without showing their inner structure.





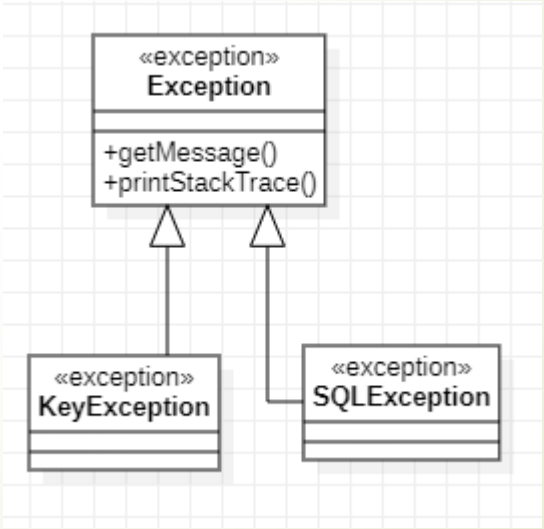
Enumeration

- An enumeration is a user-defined data type that consists of a name and and ordered list of enumeration literals.



exceptions

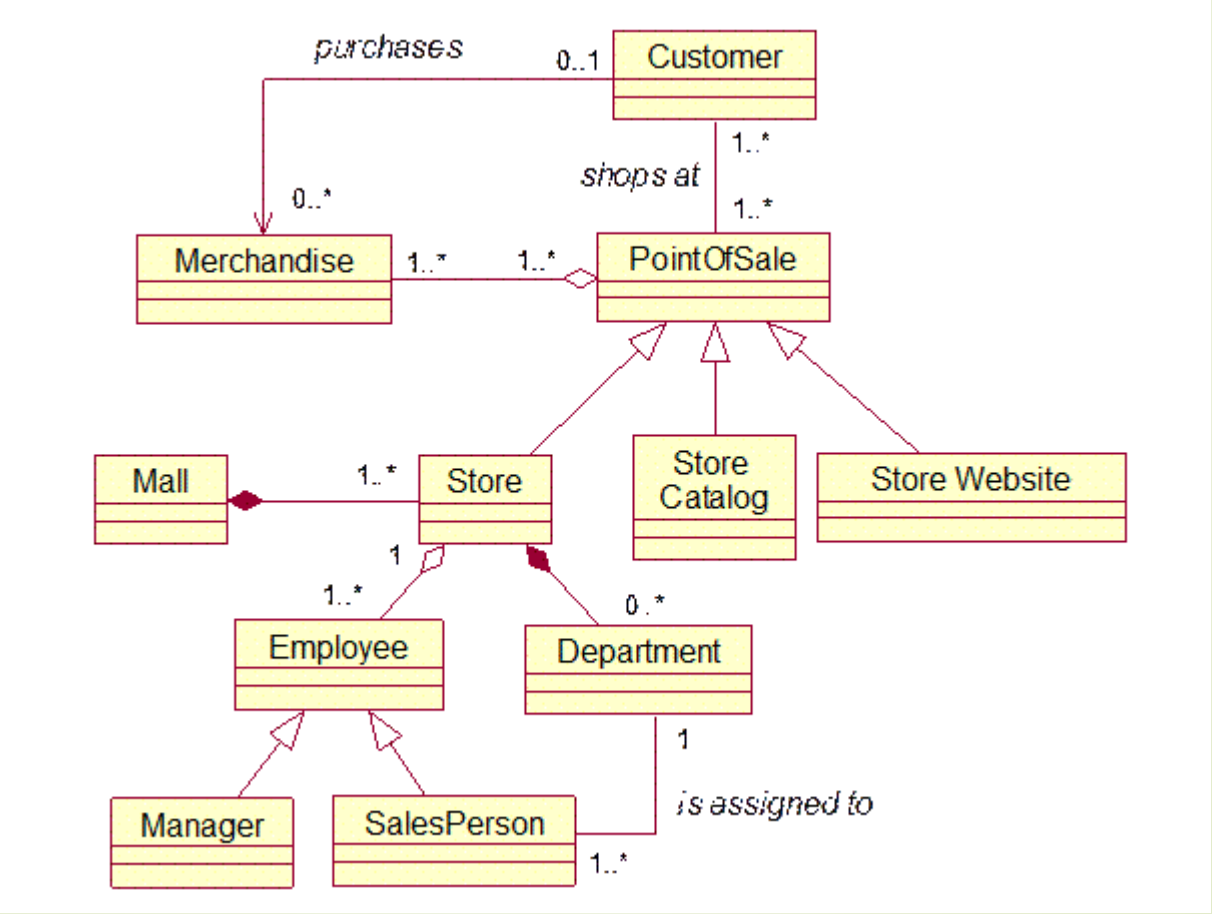
- *Exceptions* can be modeled just like any other class.
- Notice the <<exception>> stereotype in the name compartment.



Exercise 1

- Draw a UML class diagram that models the relationships between the following classes: *Mall, Store, Sales Person, Department, Manager, Merchandise, Store Catalog, Store Website, and Customer*.

Answer


purchases
0..1
Customer
1..*
shops at
1..*
0..*
Merchandise
1..*
1..*
PointOfSale
Mall
1..*
Store
Store Catalog
Store Website
1
1..*
0..*
Employee
Department
1
is assigned to
Manager
SalesPerson
1..*

Exercise 2

- Draw a UML class diagram that models the relationships between the classes in each of the following lists. Model using aggregation, association, composition and inheritance relationships.

    - Bank, Savings Account, Loan, Teller, ATM, Customer, Checking Account