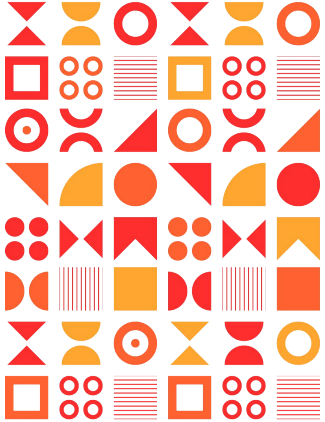


OBJECT ORIENTED DESIGN

semicolon

What we will cover

- Concisely define each of the following key data modelling terms: Object, state, behavior, object class, class diagram, operation, encapsulation, association role, polymorphism, aggregation and composition
- Draw a class diagram to represent business situations

OBJECTS

When using object-oriented approach in designing systems, you see the system from the lens of real world objects.

What are objects: An object is an entity that has a well defined role in the application domain, it has state, behavior and **identity characteristic**

OBJECTS

An object has a **STATE** which emcompasses it properties (attributes and relationships with other objects) and the values of those properties

An object also have behavior which represents how an object acts and reacts, these behaviors by conventions are represented by VERBS (action words)

OBJECTS: EXAMPLE

Consider an example of a student called Sade Ciroma, who is represented as on object in a system, the student will be characterized by its attributes, such as *name, class, age* as well as the **values** for these attributes.

In this case, the values for the attributes will for example; name= Sade Ciroma, class= SS3, age=22.

OBJECTS: EXAMPLE

This student object will also have behavior which are actions she can take such as:

cal_gpa to calculate GPA

register_crse to register for a course

OBJECT CLASS

An object class or CLASS is a logical grouping of objects that have the same (or similar) attributes, relationships and behaviours.

So far we have been using an example of a student called Sade Ciroma. We can now say **student** is a class(object class) and Sade Ciroma is an object of the class STUDENT.

OBJECT IN RELATION TO CLASS

Another way of defining OBJECTS is to say **an object is a SINGLE OCCURENCE of a class.**

Or we say an object is an INSTANCE of a class

OBJECT IDENTITY

- No two objects are the same
  - In our previous example of a **student** class. If there are two occurences of students with the name = Sade Ciroma, class = SS3, age = 22. They are two distinct objects and maintain such througout the life of the system. If one of the Sade Ciroma decide to go back one class, only the value for her class changes and it does not affect the other Sade Ciroma, even though they share the same attributes.

CLASS DIAGRAM

A class diagram shows the static structure of an object oriented model, the object classes, their internal structure, and the relationships in which they participate.

CLASS DIAGRAM

In UML notation, the class is represented by a rectangle with three compartment seperated by horizontal lines.

The top of the compartment contains the Class name, the second compartment holds the attributes for the class while the last compartment holds the behaviour (operations) for the class

CLASS DIAGRAM

OPERATIONS

Operations are tha actions or functions provided by all the instances of a class to invoke a behavior.

Operations provides an external interface to a class

An operation allows the internal details of a class to be hidden, that is the ‘actor’ does not need to know the structure and implementation details of the class. But only ‘interface’ to it through operations.

The concept of hiding internal implementation details on an object is known as ENCAPSULATION or INFORMATION HIDDING.

OPERATIONS

Operations can be classified into three parts

- Constructor operation
- Query operation
- Update operation

Constructor operation

A constructor operation creates an instance (occurence) of a class. So for our STUDENT CLASS, we could have a constructor operaion called *create-student()* which creates a new student and initializes its state.

Query operation

A query operation is an operation that has no side effects; it accesses the state of the object but does not alter the state.

Query operation

In our previous example of a student class, imagine we have an operation known as *get_age()* which returns the age of the student.

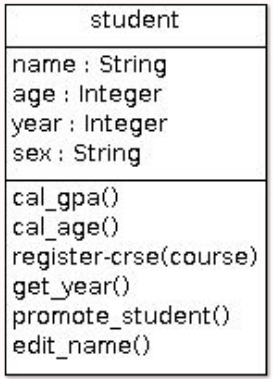
Class work:

Does this operation affect the state(age) of the object?

Then get_age() is a query operation.

Identify all the query operations in the next slide

Query operation



UPDATE OPERATION

An update operations alters the state of an object when it is invoked. That means it has a side effect. Imagine we have an operation called *promote_student()* what this will do is to change the year of the student by incrementing it by one.

Class work

Identify the update operation in the previous slide.

Why do some operation have explicit argument?

CLASS-SCOPE OPERATION

A class-scope operation is an operation that applies to a class rather than an object instance. For example, we may have a function *avg_gpa()* that calculates the average GPA of of students in our class. This function is said to be a class scope operations, since it needs every student GPA to make its calculation. In UML notation, class-scope operation are identified by undelining them

ASSOCIATIONS

An association is a relationship among instances of classes.

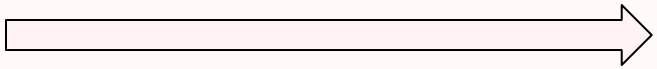
An association is represented by a solid line between the participating classes

Generic Association

This is represented by a straight thick line

Generalization

This is a “is a” relationship. In programming languages such as JAVA, it depicts inheritance. It is represented by an arrow



Aggregation

This is a “part of” association, which means multiple owners.

This is represented by an arrow with a diamond head

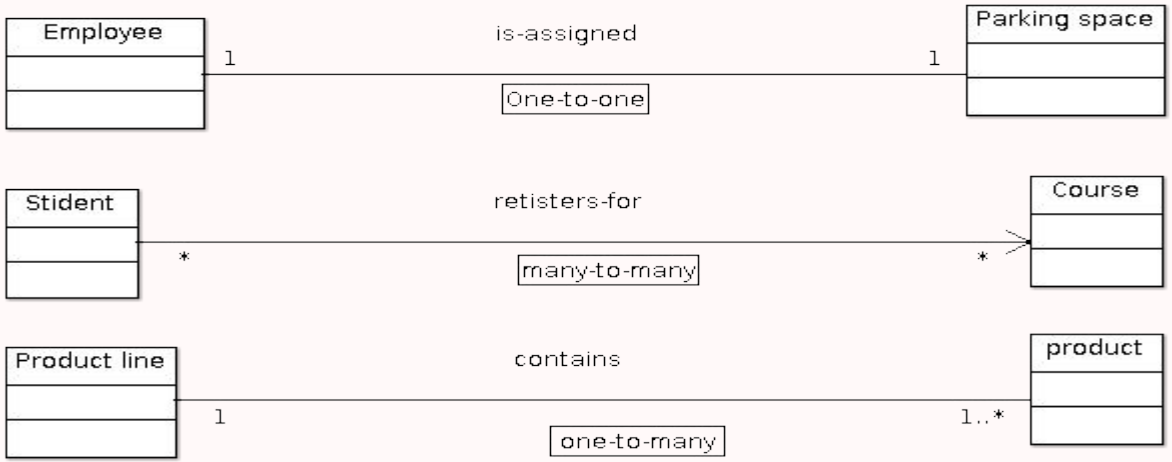
Composition

This is a special case of aggregation which means “exactly one owner” it is represented by an arrow with a diamond head, but unlike aggregation, the head is filled

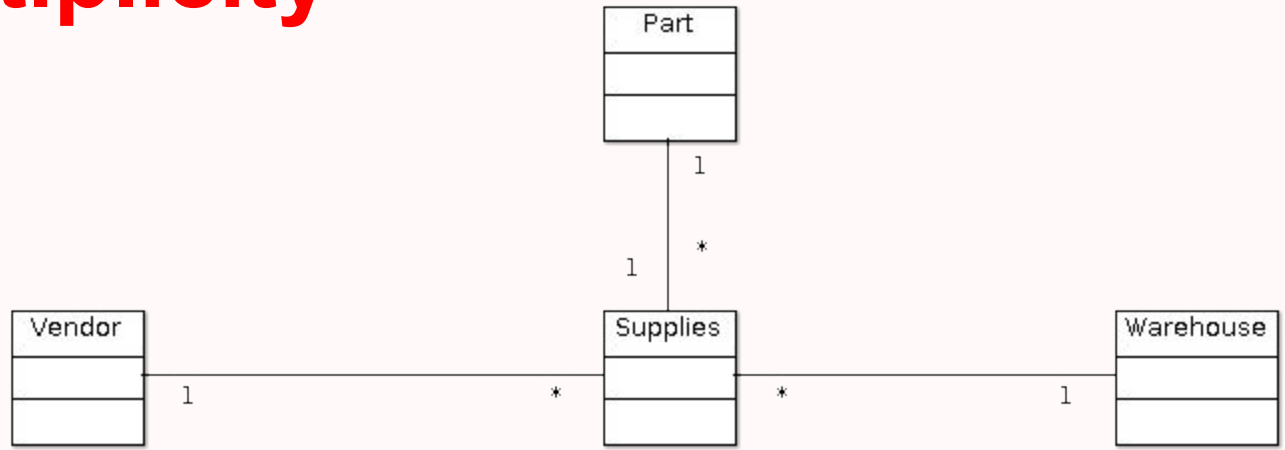
Multiplicity

This is a specification that indicates how an object participate in a relationship

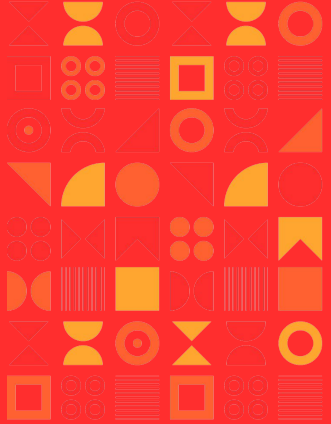
Multiplicity



Multiplicity



Multiplicity



semicolon