Experiment 7

Class Diagram

Software engineering

Introduction

Class diagrams are the main building blocks of every object-oriented methods. The class diagram can be used to show the classes, relationships, interface, association, and collaboration. Since classes are the building block of an application that is based on OOPs, so as the class diagram has appropriate structure to represent the classes, inheritance, relationships, and everything that OOPs have in its context. It describes various kinds of objects and the static relationship in between them.

# The main purpose to use class diagrams are:

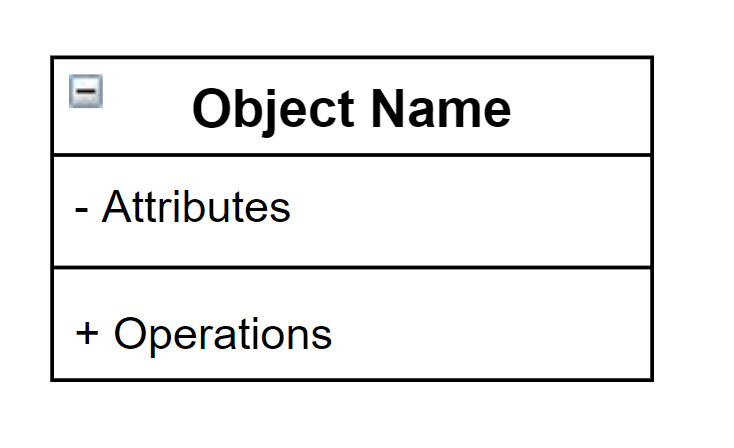
# This is the only UML which can appropriately depict various aspects of OOPs concept.

# Proper design and analysis of application can be faster and efficient.

# It is base for deployment and component diagram.

# Class Model

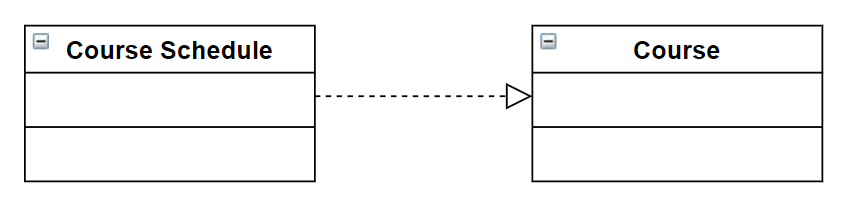
The class model consists od three things class name, Attributes, Operations.



Types of Relationships

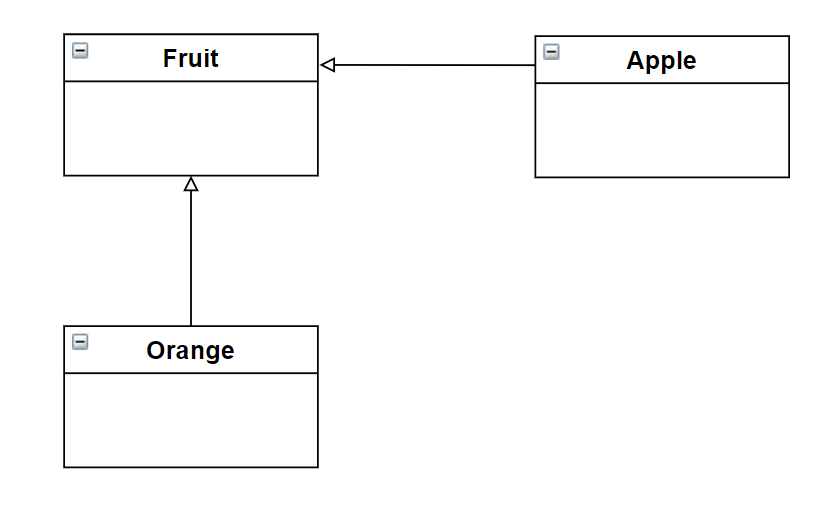
1. Dependency

A dependency means the relation between two or more classes in which a change in one may force changes in the other. However, it will always create a weaker relationship. Dependency indicates that one class depends on another.



1. Generalization

A generalization helps to connect a subclass to its superclass. A sub-class is inherited from its superclass. Generalization relationship can't be used to model interface implementation. Class diagram allows inheriting from multiple super-classes.



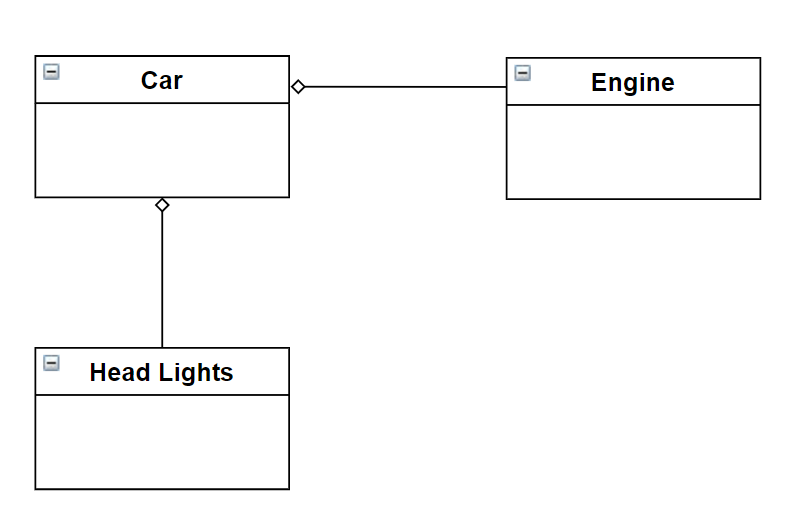
1. Association

This kind of relationship represents static relationships between classes A and B. For example; an employee works for an organization.

* Only one-one 1-1
* Zero or many 0...\*
* One or many 1...\*
* Zero or one 0 . \*
* Some range 2...5

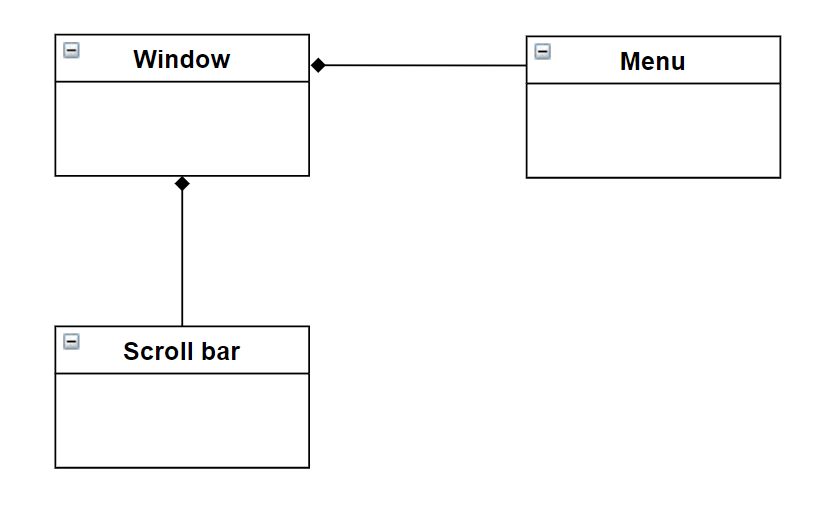
1. Aggregation

Aggregation is a special type of association that models a whole- part relationship between aggregate and its parts.



1. Composition

The composition is a special type of aggregation which denotes strong ownership between two classes when one class is a part of another class.



# CLASS DIAGRAM

A close up of a map

Description automatically generated