Topic 5

Introduction to UML Diagrams

Objectives

- To introduce UML Diagrams
  - A diagrammatic way of showing the relationships among classes
  - This will help our understanding of the definitions of our collections and the usage of our collections in applications

UML Diagrams

- ***Unified Modeling Language (UML)*** is a standard notation for object-oriented design
  - Used to ***model*** object-oriented designs
  - Shows overall design of a solution
    - Shows class specifications
    - Shows how classes interact with each other
  - Diagrams use specific icons and notations
  - It is ***language independent***

UML Class Diagram

- A class is represented in a UML diagram by a rectangle divided into **3** sections:
  - ***name*** of the class
  - ***attributes*** of the class (i.e. the data fields of the class, including variables and constants)
  - ***operations*** of the class (essentially equivalent to a Java method or a C++ function)

Example: UML Class Diagram

Person
firstName lastName email
getName() getEmail() setEmail( ) equals( ) toString()

Example: UML Class Diagram

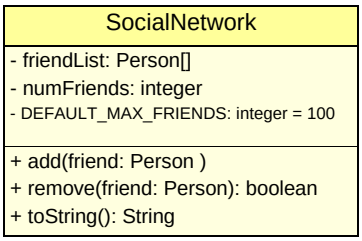
SocialNetwork
friendList numFriends DEFAULT_MAX_FRIENDS
add( ) remove() toString()

Features of UML Class Diagrams

- Attributes and operations may include:
  - ***visibility***: public (+) or private (-)
  - ***type*** of attribute or operation
  - ***parameter list*** for operations
- Including this information is of the form:

visibility variable_name: type
visibility variable_name: type = default_value
visibility method_name(parameter_list): return_type {property}

Example: UML Class Diagram



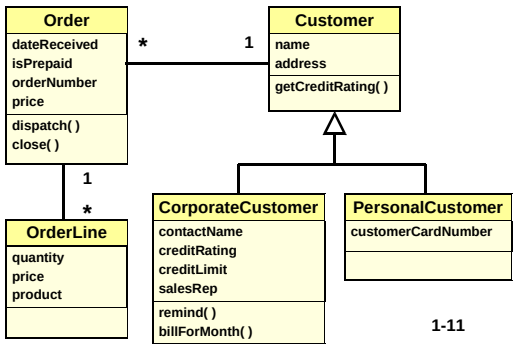
Features of UML Class Diagrams

- Attributes and operations may be left incomplete, and completed as design is developed

Set of UML Class Diagrams

- A set of UML class diagrams shows:
  - The classes used in the system
  - The ***relationships*** among classes
  - The ***constraints*** on the connections among classes

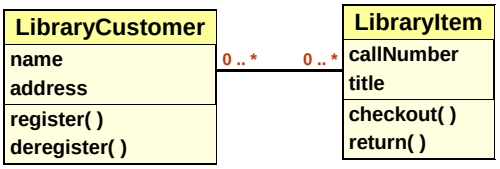
Example: UML Diagram for Order Processing



Features of Set of UML Diagrams

- ***Association between classes***:
  - Represents a relationship between objects of those classes
  - Indicated with a *solid line* between the classes
  - Can be annotated with ***cardinality***: indicates a numeric association between classes, such as:
    - **one-to-one**
    - **one-to-many (1..*)**
    - **many-to-many (*..*)**
    - **zero-to-many (0..*)**
    - **zero-to-5 (0..5)**
    - **etc.**

Example: Association Between Classes



Association Between Classes

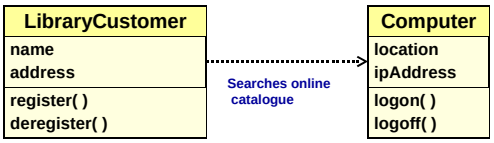
- What is the Order-Customer relationship in our Order Processing System?
- How would we annotate that a Library Customer can not check out more than 5 library items?

Features of Set of UML Diagrams

- ***Usage of another class***:
  - Broken line with an arrow indicates that one class makes use of the other

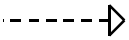
  - Line can be labeled with a message indicating the type of usage

Example: One Class Indicating its Use of Another



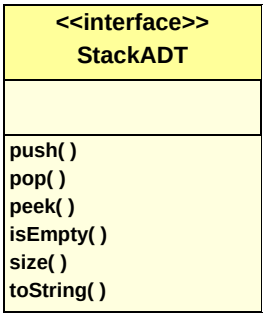
Features of Set of UML Diagrams

- ***Implementation of an interface***:
  - Indicated by a broken line with an open arrow

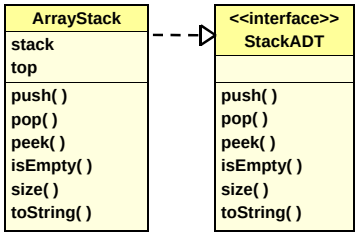


  - ***UML diagram for an interface*** is much like the UML diagram for a class

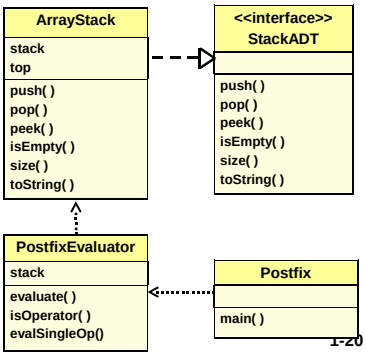
UML Diagram for StackADT Interface



UML Diagram for ArrayStack Implementation of StackADT



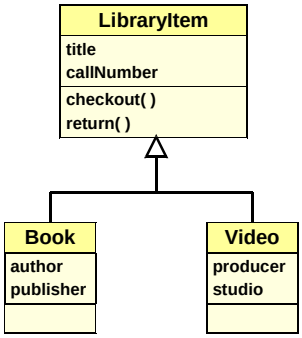
UML Diagram for Postfix Expression Program



Features of Set of UML Diagrams

- ***Inheritance:***
  - An arrow on an association line indicates that one class is derived from the other

Example: Inheritance Relationships



1-22

Example: Inheritance Relationships