UML and Classes, Objects and Relationships [2]

Defining Domain Models Using Class Diagrams

---

Association - Multiplicity

- A Student can take many Courses and many Students can be enrolled in one Course.

![Diagram depicting the association between Student and Course with multiplicities.]

Notes

- One class can be relate to another in a
  - One-to-one
  - One-to-many
  - One-to-one or more
  - One-to-zero or one
  - One-to-a bounded interval (one-to-two through twenty)
  - One-to-exactly n
  - One-to-a set of choices (one-to-five or eight)

Association - Self

- An association that connects a class to itself is called a self association.

![Diagram illustrating self-association with a company having employees, where one manager is responsible for up to 10 workers.]

Notes

- Multiplicity can be expressed as,
  - Exactly one - 1
  - Zero or one - 0..1
  - Many - 0..* or *
  - One or more - 1..*
  - Exact Number - e.g. 3..4 or 6
  - Or a complex relationship - e.g. 0..1, 3..4, 6..* would mean any number of objects other than 2 or 5
Association - Multiplicity

- A cricket team has 11 players. One of them is the captain.
- A player can play only for one Team.
- The captain leads the team members.

Class Relationships

- Association
- Generalization
- Realization
- Dependency

Generalization (Inheritance)

- Child class is a special case of the parent class

Generalization (Inheritance) e.g.

Inheritance - Implementation

```
public class Circle {
    
}

public class GraphicCircle extends Circle {
    
}
```

Abstract Class

```
Shape

Circle

Rectangle
```
Abstract Methods (Operations)

```
public abstract class Shape {
    public abstract draw(); // declare without implementation
    ..........
}
public class Circle {
    public draw(){
        ..........
    }
    ..........
}
```

Class Relationships

- Association
- Generalization
- Realization
- Dependency

Realization - Interface

```
public interface TypeWriter {
    void keyStroke();
}
```

```
public class Keyboard implements TypeWriter {
    public void keyStroke(){
        ..........
    }
    ..........
}
```

Class Relationships

- Association
- Generalization
- Realization
- Dependency
Dependency

- Change in specification of one class can change the other class. This can happen when one class is using another class.

Class Diagrams

- The UML class diagram consists of several Classes, connected with Relationships.

Class Diagram - Example

- Draw a class diagram for a information modeling system for a school.
- School has one or more Departments.
- Department offers one or more Subjects.
- A particular subject will be offered by only one department.
- Department has instructors and instructors can work for one or more departments.
- Student can enrol in upto 5 subjects in a School.
- Instructors can teach upto 3 subjects.
- The same subject can be taught by different instructors.
- Students can be enrolled in more than one school.

Class Diagram - Example

- School has one or more Departments.
- Department offers one or more Subjects.
- A particular subject will be offered by only one department.

Class Diagram - Example

- Department has Instructors and instructors can work for one or more departments.
- Student can enrol in upto 5 Subjects.
Class Diagram - Example

- Instructors can teach up to 3 subjects.
- The same subject can be taught by different instructors.

![Class Diagram Example]

Class Diagram - Example

- Students can be enrolled in more than one school.

![Class Diagram Example]

Object Diagram

- Object Diagram shows the relationship between objects.
- Unlike classes objects have a state.

![Object Diagram - Example]

Object Diagram - Example

- `c1: Company`
- `d1: Department`
  - `name = "Sales"`
- `d2: Department`
  - `name = "R&D"`
- `p1: Person`
  - `name = "John"`
- `p2: Person`
  - `name = "David"`