Class Diagram

NOUR
Contents

1. Introduction
2. Representing a class
3. Classes Relationships
Introduction

- Class diagram is a way of showing system classes and the relations among them.

- Static diagrams, pick only existence and relationships but not actions.

- Where to get your classes?
Representing a class

- Class name
- Attributes
- Operations (methods)
## Attributes

### Syntax:

`attributeName : type`

### Ex:

- `firstName : String`
- `salary : Dollars`
Methods

Syntax:

methodName(parm) : return type

Ex:

setName(String)
getSalary() : Dollars
Representing a class

- Complete Example

<table>
<thead>
<tr>
<th>Employee</th>
</tr>
</thead>
<tbody>
<tr>
<td>firstName : String</td>
</tr>
<tr>
<td>lastName : String</td>
</tr>
<tr>
<td>employeeID : Integer</td>
</tr>
<tr>
<td>salary : Dollars</td>
</tr>
</tbody>
</table>

| hire() |
| setSalary(Integer) |
| getSalary() : Integer |
Visibility

You can show the visibility of attributes and methods by these signs

+ : public
- : private
# : protected
~ : package

<table>
<thead>
<tr>
<th>Employee</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ firstName : String</td>
</tr>
<tr>
<td>+ lastName : String</td>
</tr>
<tr>
<td># employeeID : Integer</td>
</tr>
<tr>
<td>- salary : Dollars</td>
</tr>
</tbody>
</table>

~ hire()
- setSalary(Integer)
+ getSalary() : Integer
Okay!

- We have learned how to create classes.
- Now let's learn how to connect them!
Classes Relationships

- Association
- Aggregation
- Composition
- Generalization

  - Multiplicity
If there is a conceptual connection between two classes we say that these classes are associated.
Multiplicity

An indication of numerical limit of a relation.
Aggregation & Composition

- Both describe the relation between a whole and its parts.

What is the difference?!
To test if you are dealing with aggregation of composition check:

- If the part belong only to one whole then its composition.

Classes Relationships

- Band
- Musician
- Catalog
- Product
- Building
- Room
Aggregation & Composition

- Aggregation

- Composition

- The diamond sign is always at the side of the whole.
Generalization

- A way of describing inheritance in Class Diagram.

- Employee
  - name: String
  - hire()

- Salaried Emp
- HourlyEmp
- TempEmp
Thank You!