UML Diagrams

Java Software Solutions
Foundations of Program Design
Sixth Edition

by
Lewis & Loftus
UML Diagrams

• UML stands for the *Unified Modeling Language*

• *UML diagrams* show relationships among classes and objects
  – Provide a visual representation of a project

• There are several kinds of UML diagrams
  – Class diagrams
  – Object diagrams
  – Collaboration diagrams
  – Sequence diagrams
UML Class Diagrams

• Used to illustrate relationships between classes
• Used to show the details of a particular class
UML Class Diagrams

- A UML class diagram consists of one or more classes, each with sections for the class name, attributes (data), and operations (methods).

- Lines between classes represent associations:
  - A dotted arrow shows that one class uses the other (calls its methods).
The uses relationship

• One class uses another when a method creates and uses an object of the second type.
  – A main class uses other classes

• A UML class diagram for the RollingDice program:

```
<table>
<thead>
<tr>
<th>RollingDice</th>
<th>Die</th>
</tr>
</thead>
<tbody>
<tr>
<td>main (args : String[]) : void</td>
<td>faceValue : int</td>
</tr>
<tr>
<td></td>
<td>roll() : int</td>
</tr>
<tr>
<td></td>
<td>setFaceValue (int value) : void</td>
</tr>
<tr>
<td></td>
<td>getFaceValue() : int</td>
</tr>
<tr>
<td></td>
<td>toString() : String</td>
</tr>
</tbody>
</table>
```
Class Diagram for Pig Latin

PigLatin

+ main (args : String[]) : void

PigLatinTranslator

+ translate (sentence : String) : String
- translateWord (word : String) : String
- beginsWithVowel (word : String) : boolean
- beginsWithBlend (word : String) : boolean
Aggregation

• This is a has-a relationship
• One class has an instance variable whose type is another class
  – A Photo has a Date
  – A Photo has a File
• Represented by a solid line with a diamond on the class that holds an object of the other class
Aggregation in UML

StudentBody

+ main (args : String[]) : void

Student

- firstName : String
- lastName : String
- homeAddress : Address
- schoolAddress : Address
+ toString() : String

Address

- streetAddress : String
- city : String
- state : String
- zipCode : long
+ toString() : String
Inheritance

• This creates an *is-a* relationship, meaning the child *is* a more specific version of the parent
  – Subclass extends Superclass
• Inheritance relationships are shown in a UML class diagram using a solid arrow with an unfilled triangular arrowhead pointing to the parent class
Example Class Diagram

```
Book
# pages : int
+ pageMessage() : void

Dictionary
- definitions : int
+ definitionMessage() : void

Words
+ main (args : String[]) : void
```
Class Hierarchies

• A child class of one parent can be the parent of another child, forming a class hierarchy

```
Business
    /    \
RetailBusiness  ServiceBusiness
    /    \
KMart  Macys  Kinkos
```
Abstract classes

- Use abstract for the stereotype to show a class is abstract
Interfaces

• Implementing an interface also creates an is-a relationship
• Use a dashed arrow with an outlined triangular head
• Use interface for the stereotype
# UML Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Visibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>public</td>
</tr>
<tr>
<td>-</td>
<td>private</td>
</tr>
<tr>
<td>#</td>
<td>protected</td>
</tr>
<tr>
<td>~</td>
<td>package</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Relationship</th>
<th>Connector</th>
</tr>
</thead>
<tbody>
<tr>
<td>inheritance (is-a)</td>
<td>open arrow</td>
</tr>
<tr>
<td>association (uses)</td>
<td>solid line</td>
</tr>
<tr>
<td>aggregation/composition (has-a)</td>
<td>open/solid diamond</td>
</tr>
<tr>
<td>inner class</td>
<td>circle with cross</td>
</tr>
</tbody>
</table>
Object Diagrams

- There are several kinds of diagrams that show the objects in a program and how they interact
  - Object diagrams show the state of the objects at a particular point in the program
  - Collaboration diagrams show the interactions between objects by showing method calls
  - Sequence diagrams show interactions between objects as a function of time
Object diagram

• An object is also represented as a box
• The name of the object (optional) and the name of the class are underlined
• Attribute values may be shown

```
c : Color
  redValue  51
  greenValue  102
  blueValue  204
```
Collaboration Diagram

• Shows interactions (e.g. method calls) between objects
• Active object is bold
Sequence Diagram

- Show interactions between objects as a function of time
- Time axis goes down
Tools for UML Diagrams

- Available on onyx
  - dia (can also download for Windows, free)
  - umbrello
  - Violet (a jar file)
- Eclipse plug-in