CE204 Object-Oriented Programming

UMPLE - Part 1

Author: Asst. Prof. Dr. Uğur CORUH

## CE204 Object-Oriented Programming

## Week-6 (UMPLE - Part 1)

#### Spring Semester, 2021-2022

Download [DOC-PDF](ce204-week-6.en.md_doc.pdf), [DOC-DOCX](ce204-week-6.en.md_word.docx), [SLIDE](ce204-week-6.en.md_slide.pdf), [PPTX](ce204-week-6.en.md_slide.pptx),

## UMPLE

### Common Scope

* What is UMPLE?
* What is its purpose?
* How to create a UML model with UMPLE?
* What is philosophy of UMPLE?

### Common Scope

* How to use UMPLE?
  + UMPLE Online
  + Command-Line
  + Eclipse Plugin
  + Visual Studio Code Plugin

### Common Scope

* How to learn UMPLE?
  + Online Documentations
  + Video Tutorials
  + UMPLE Community

### Common Scope

* Overview of the basics of UMPLE
* Associations in UMPLE
* State machines in UMPLE
* Product lines in UMPLE: Mixins and Mixsets
* Other separation of concerns mechanisms: (Aspects and traits) and their code generation
* Other advanced features of UMPLE
* Hands-on exercise developing versions of a concurrent system using state machines and product lines.
* UMPLE as written in itself: A case study.

### Common Scope

* Introduction:
* Overview of Model-Driven Development
  + Languages / Tools / Motivation for UMPLE
* Class Modeling
  + Tools / Attributes / Methods / Associations / Exercises / Patterns
* Modeling with State Machines
  + Basics / Concurrency / Case study and exercises
* Separation of Concerns in Models
  + Mixins / Aspects / Traits
* More Case Studies and Hands-on Exercises
  + UMPLE in itself / Real-Time / Data Oriented
* Conclusion

### Outline - UMPLE Part 1

* Introduction to UMPLE
* Motivation for developing UMPLE
* Some key UMPLE innovations
* Using UMPLE
* UMPLE Philosophy
* UMPLE Class Modeling

### Outline - UMPLE Part 1

* UMPLE Online Usage
* UMPLE Attributes
* UMPLE Generalization and interfaces
* UMPLE Methods
* UMPLE Associations

## **Introduction to UMPLE**

### UMPLE: Simple, Ample, UML Programming Language

* **Open source textual modelling tool set for 3 platforms**
  + Command line compiler
  + Web-based tool (UMPLEOnline) for demos and education
  + Eclipse plugin
* **Code generator for UML ++**
  + Infinitely nested state machines, with concurrency
  + Proper referential integrity and multiplicity constraints on associations
  + Traits, mixins, aspects for modularity
  + Text generation templates, patterns, traits
* **Pre-processor to add UML, patterns and other features on top of Java, PhP, C++ and other languages**

### UMPLE: Simple, Ample, UML Programming Language

* Open source textual modeling tool and code generator
  + Adds modeling to Java,. C++, PHP
  + A sample of features
    - Referential integrity on associations
    - Code generation for patterns
    - Blending of conventional code with models
    - Infinitely nested state machines, with concurrency
    - Separation of concerns for models: mixins, traits, mixsets, aspects
* Tools
  + Command line compiler
  + Web-based tool (UMPLEOnline) for demos and education
  + Plugins for Eclipse and other tools

### What Are we Going to Learn About in This Tutorial? What Will You Be Able To Do?

* Modeling using **class diagrams**
  + Attributes, Associations, Methods, Patterns, Constraints
* Modeling using **state diagrams**
  + States, Events, Transitions, Guards, Nesting, Actions, Activities
  + Concurrency
* **Separation of Concerns** in Models
  + Mixins, Traits, Aspects, Mixsets
* Practice with a examples focusing on **state machines** and **product lines**
* Building a complete system in UMPLE

### What Technology Will You Need?

* As a minimum: Any web browser.
* For a richer command-line experience
  + A computer (laptop) with Java 8-14 JDK
  + Mac and Linux are the easiest platforms, but Windows also will work
  + Download UMPLE Jar at http://dl.UMPLE.org
* You can also run UMPLE in Docker: http://docker.UMPLE.org

### Key Websites

* Entry-point: https://www.UMPLE.org
  + Everything you need to get started with UMPLE
* Github: https://github.com/UMPLE/UMPLE
  + Source code and examples for UMPLE
* UMPLE Online: https://try.UMPLE.org
  + Online application for UMPLE

### Key Websites (Another way)

The UMPLEOnline web interface is at try.UMPLE.org

The user manual is at manual.UMPLE.org

The UMPLE home page is at www.UMPLE.org

UMPLE download page: dl.UMPLE.org

## **Motivation for developing UMPLE**

### Motivation for developing UMPLE (1)

Designers want the best combination of features:

* Textual editing and blending with other languages
* Ability to use in an agile process
  + Write tests, continuous integration, versioning
  + Combine the best of agility and modeling
* Excellent code generation
  + A complete generation of real systems (including itself)
* Multi-platform (command line, Eclipse, VsCode, Web)
* Practical and easy to use for developers
  + Including great documentation
* Open source

### Motivation for developing UMPLE (2)

Many existing tools:

* Lacked in usability
  + Awkward to edit diagrams
  + Many steps to do a task
  + Lengthy learning process
* Lack in ongoing support
* Could be enhanced by us perhaps, but we would be tied to key decisions (e.g. Eclipse-only)

## **Some key UMPLE innovations**

## Some key UMPLE innovations

* Model is code
  + Traditional code is embedded in model
* No need to edit generated code
  + No *round-trip engineering*

## **Using UMPLE**

### Using UMPLE

* We will mostly be using
  + UMPLEonline
    - In a web browser: http://try.UMPLE.org
    - Or in Docker: http://docker.UMPLE.org
  + UMPLE on the command line: http://dl.UMPLE.org
    - Needs Java 8 JDK on the command line:
      * http://bit.ly/1lO1FSV
        + Java 9 works well too

### Docker Container Experimental

mkdir ~/src && cd ~/src && git clone git@github.com:UMPLE/UMPLE.git

docker run -i -t -v `pwd`:/src UMPLE/UMPLE:0.4.0 bash

### Using UMPLE

* Optional:
  + UMPLE in Eclipse
    - https://github.com/UMPLE/UMPLE/wiki/InstallEclipsePlugin
  + cmake and gcc for compiling C++ code

## **UMPLE Philosophy**

### UMPLE Philosophy 1-4

* P1. Modeling is programming and vice versa
* P2. An UMPLE programmer should never need to edit generated code to accomplish any task.
* P3. The UMPLE compiler can accept and generate code that uses nothing but UML abstractions.
  + The above is the inverse of the following
* P4. A program without UMPLE features can be compiled by an UMPLE compiler.
  + e.g. input Java results in the same as output

### UMPLE Philosophy 5-8

* P5. A programmer can incrementally add UMPLE features to an existing program
  + Umplification
* P6. UMPLE extends the base language in a minimally invasive and safe way.
* P7. UMPLE features can be created and viewed diagrammatically or textually
* P8. UMPLE goes beyond UML

## **UMPLE Class Modeling**

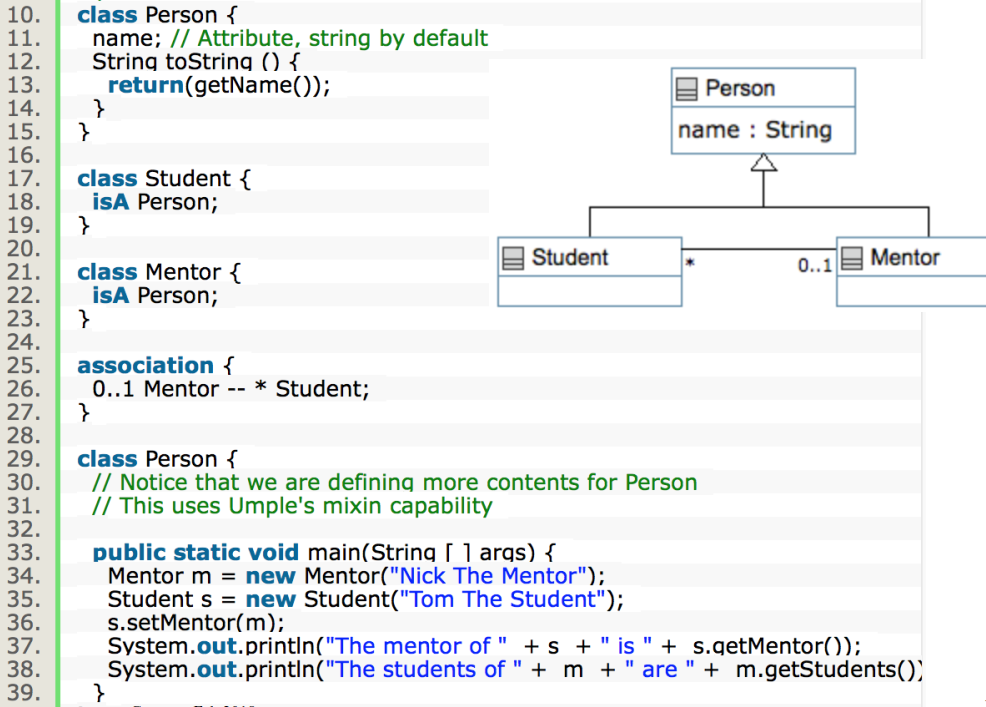
### UMPLE Class Models - Quick Overview

* Key elements:
  + Classes
  + Attributes
  + Associations
  + Generalizations
  + Methods
* We will look at all these using examples via UMPLE ONLINE
* UMPLE code/models are stored in files with suffix **.ump**

### Exercise: Compiling and changing a model

* Look at the example at the bottom of
  + http://helloworld.UMPLE.org (also on next slide)
    - Observe: attribute, association, class hierarchy, mixin
* Click on Load the above code into UMPLEOnline
  + Observe and modify the diagram
  + Add an attribute
  + Make a multiplicity error, then undo
  + Generate code and take a look
  + Download, compile and run if you want

### Hello World Example 2 in the User Manual

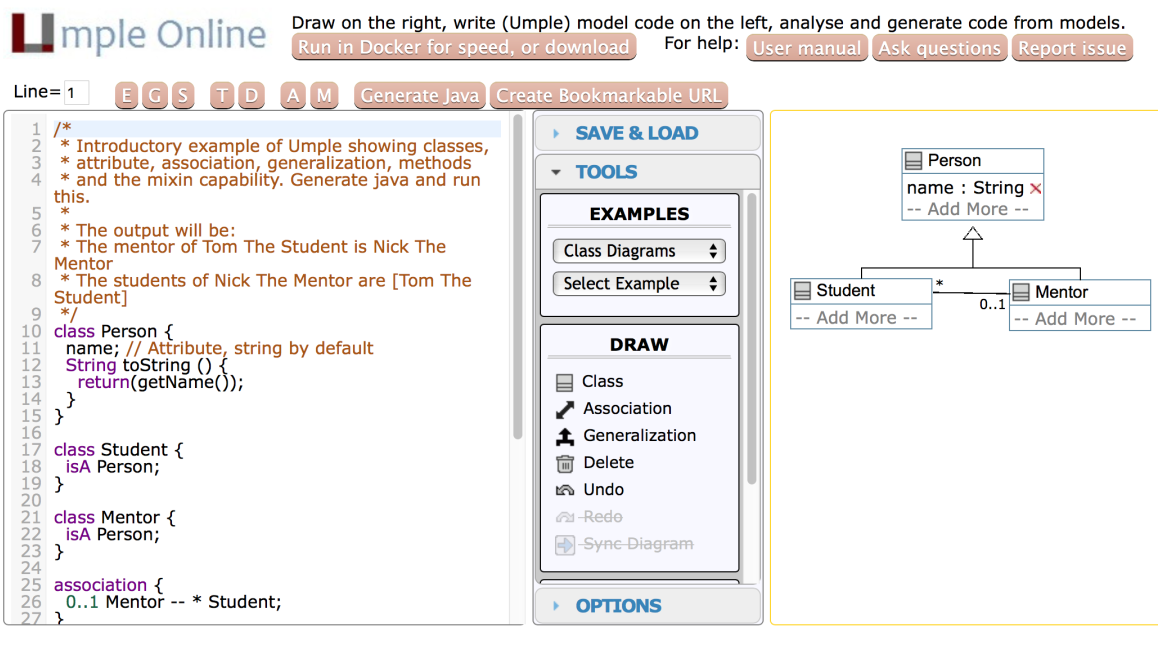


center h:550

### **Key tools:**

* UMPLE Online
* Command-Line
* User Manual

### Hello World example 2 in UMPLEOnline



center h:500

### Exploration of UMPLEOnline

* Explore class diagram examples
* Options
  + T or Control-t (**hide and show text**)
  + D or Control-d (**hide and show diagram**)
  + A, M to **hide and show attributes, methods**
  + Default diagram types
    - G/Control-g (**Graphviz**), S/Control-s (**State Diagram**)
    - E/Control-e (**Editable class diagram**)
* Generate code and look at the results
  + In UMPLE you never should modify generated code
  + It is designed to be readable for educational purposes

### Use of the UMPLEOnline Docker image

* UMPLE’s server can handle 80,000 transactions per hour
  + Code generations, edits
* But needs a good Internet connection (sometimes hundreds of students have assignments due)
* To maximize speed of UMPLEOnline run it in your local machine:
  + Follow the instructions at http://docker.UMPLE.org

### Demo of compiling on the command line

* To compile on the command line you will need Java 8
* Download UMPLE from http://dl.UMPLE.org
* Basic compilation

java -jar UMPLE.jar model.ump

* Help for features and commands

java -jar UMPLE.jar --help

* To generate and compile the java to a final system

java –jar UMPLE.jar model.ump -c -

### Quick walkthrough of the user manual

* http://manual.UMPLE.org

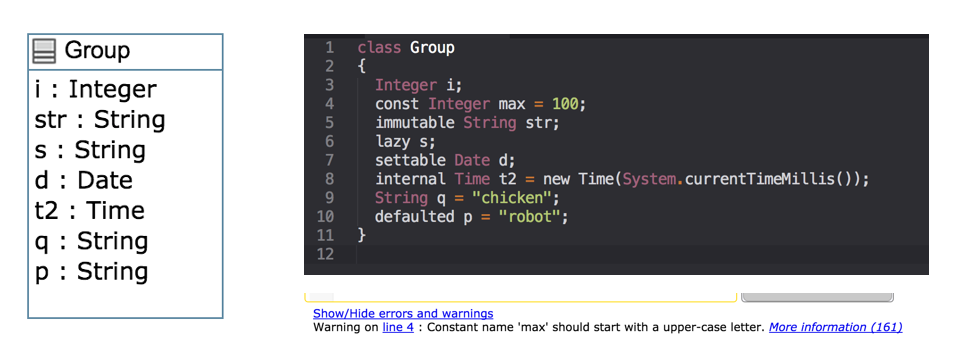
**Note in particular**

* Key sections:
  + attributes,
  + associations,
  + state machines
* Grammar
* Generated API
* Errors and warnings
* Editing pages in github

## **UMPLE Attributes**

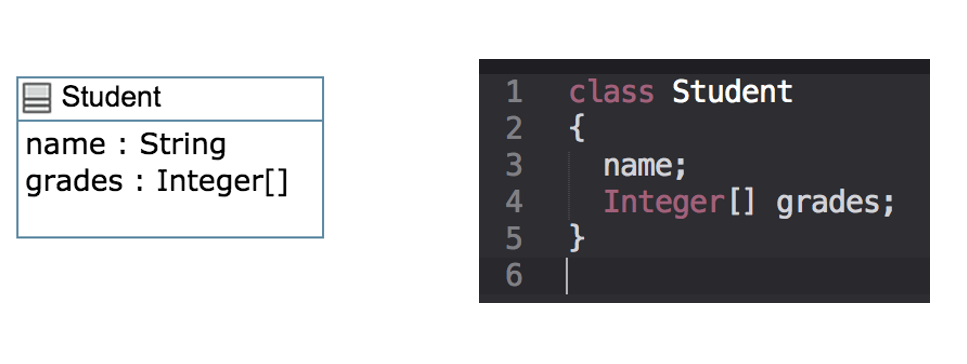
* More than just variables
  + http://attributes.UMPLE.org

### Attributes



center h:400

### Attributes Exercise #1



center h:400

### Attributes

* “*Instance variables*”
  + Part of the state of an object
  + Simple data that will always be present in each instance
* Specified like a Java or C++ field or member variable
* But, intended to be more abstract!
  + **Example**, with an initial value

a = "init value";

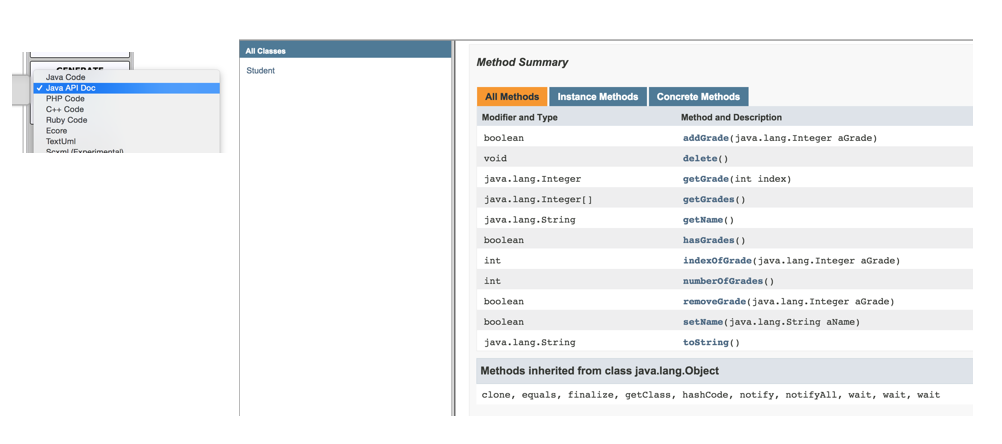
### Attributes

* As in UML, more abstract than instance variables
  + Always private by default
  + Should only be accessed get, set methods
  + Can be stereotyped (upcoming slides) to affect code generation
  + Can have aspects applied (discussed later)
  + Can be constrained (discussed later)

### Code generation from attributes

* Default code generation
  + Generates a getName() and setName() method for name
    - public
* Creates an arguments in the class constructor by default
* An attribute is private to the class by default
  + *Should only be accessed get, set methods*

### Code Generation (JavaDocs)



center h:500

### Code Generation Patterns

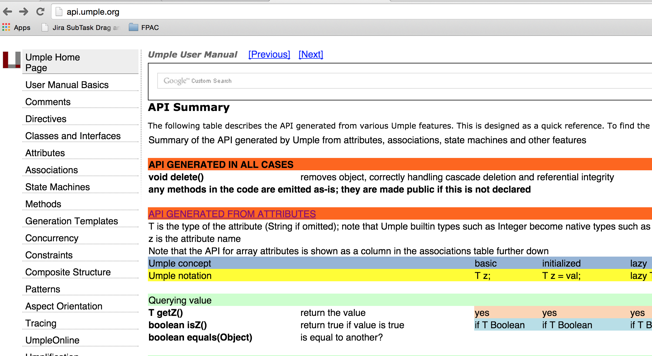
* Attributes
  + Set/Get (UB = 1)
  + Add/Remove/NumberOf/IndexOf/Get (UB > 1)
  + Lazy immutability
  + Default values
  + Constants
  + Before / After cod
* UB = upper bound

### Code Generation Patterns

* Associations
  + Set/Get (UB = 1)
  + Add/Remove/NumberOf/IndexOf/Get (UB > 1)
  + Referential Integrity
  + Multiplicity Constraints
  + 42 different cases
* UB = upper bound

### Code Generation (Semantics)

* http://api.UMPLE.org/



center h:500

### UMPLE builtin datatypes

String // (default if none specified)  
Integer  
Float  
Double  
Boolean  
Time  
Date

* The above will generate appropriate code in Java, C++ etc.
  + e.g. Integer becomes int
* Other (native) types can be used but without guaranteed correctness

### Attribute stereotypes (1)

* Code generation can be controlled through stereotypes:
  + lazy - **don’t add a constructor argument**

lazy b; // sets it to null, 0, “” depending on type

* Defaulted – *can be reset*

defaulted s = "def"; // resettable to the default

### Attribute stereotypes (2)

* autounique – provide a unique value to each instance

autounique x; // sets attribute to 1, 2, 3 …

* internal – don’t generate any methods

internal i; // doesn’t generate any get/set either

### Immutability

* Useful for objects where you want to guarantee no possible change once created
  + e.g. a geometric point
* Generate a constructor argument and get method but no set method

immutable String str;

* No constructor argument, but allows setting just once.

lazy immutable z;

### Lets explore attributes by example

* Go to
  + http://attributes.UMPLE.org

### Derived attributes

* These generate a get method that is calculated.

class Point  
{  
// Cartesian coordinates  
Float x;  
Float y;  
  
// Polar coordinates  
Float rho =  
{Math.sqrt(Math.pow(getX(), 2) + Math.pow(getY(), 2))}  
Float theta =  
{Math.toDegrees(Math.atan2(getY(),getX()))}  
  
}

### Multi-valued attributes

* Limit their use. Associations are generally better.

class Office {  
Integer number;  
Phone[] installedTelephones;  
}  
  
class Phone {  
String digits;  
String callerID;  
}

### Keys

* Enable UMPLE to generate an equals() and a hashcode() method

class Student {  
Integer id;  
name;  
key { id }  
}

* The user manual has a sports team example showing keys on associations too
* Note how this feature is not inherited from UML

## **UMPLE Generalization and interfaces**

### Generalization in UMPLE

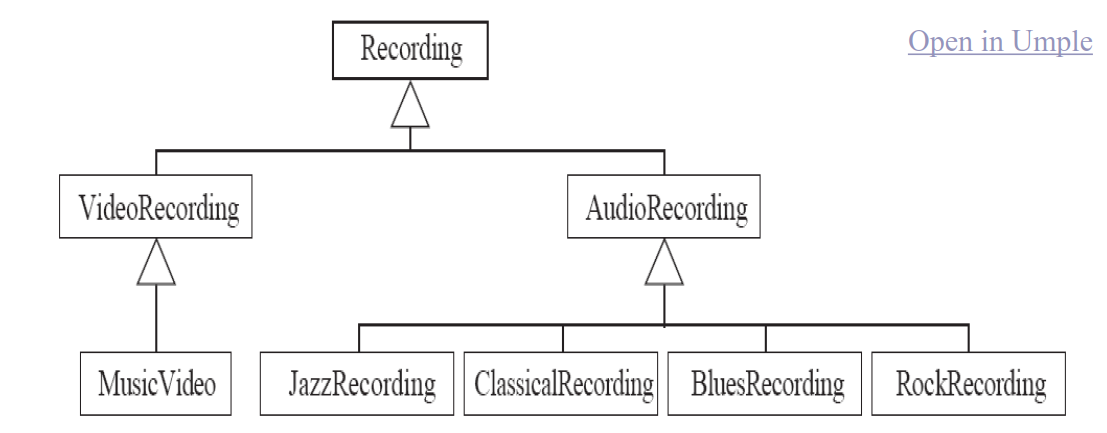
* UMPLE uses the isA keyword to indicate generalization
* Used to indicate superclass, used trait, implemented interface

class Shape {  
colour;  
}  
class Rectangle {  
isA Shape;  
}

### Avoiding unnecessary generalizations

[Open in UMPLE](https://cruise.UMPLE.org/UMPLEonline/?text=class%20Recording%7B%20%20*%20--%201%20RecordingCategory%20category;%7Dclass%20RecordingCategory%7B%20%200..1%20--%20*%20RecordingCategory%20subcategory;%7D//$?%5BEnd_of_model%5D$?class%20Recording%7B%20%20position%20157%2030%20109%2045;%20%20position.association%20Recording__RecordingCategory%2062,46%2075,0;%7Dclass%20RecordingCategory%7B%20%20position%20149%20135%20133%2045;%7D)

* Inappropriate hierarchy of Classes
* What should the model be?



center h:300

### Interfaces

* Declare signatures of a group of methods that must be implemented by various classes
* Also declared using the keyword isA
* Essentially the same concept as in Java
* *Let’s explore examples in the user manual …*

## **UMPLE Methods**

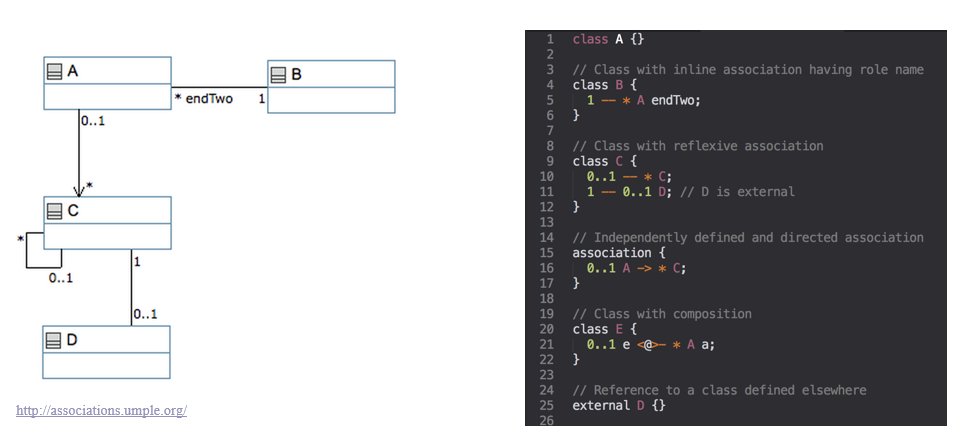
### User-written **Methods** in UMPLE

* Methods can be added to any UMPLE code.
* UMPLE parses the signature only; the rest is passed to the generated code.
* You can specify different bodies in different languages
* *We will look at examples in the user manual …*

## **UMPLE Associations**

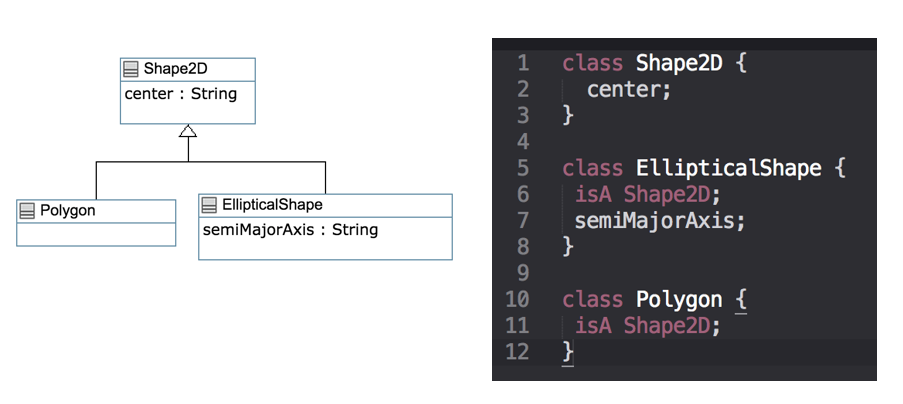
* <http://associations.UMPLE.org>
  + Notice the inline and independent state machines

### Associations



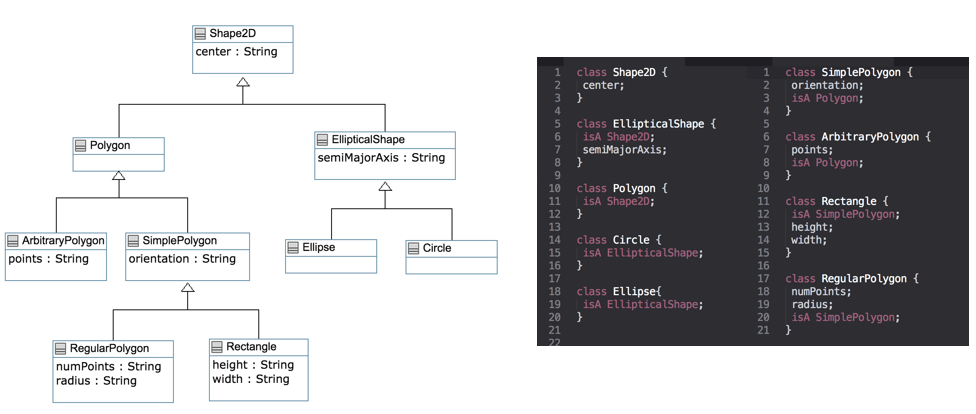
center h:500

### Associations Exercise #1



center h:500

### Associations Exercise #2

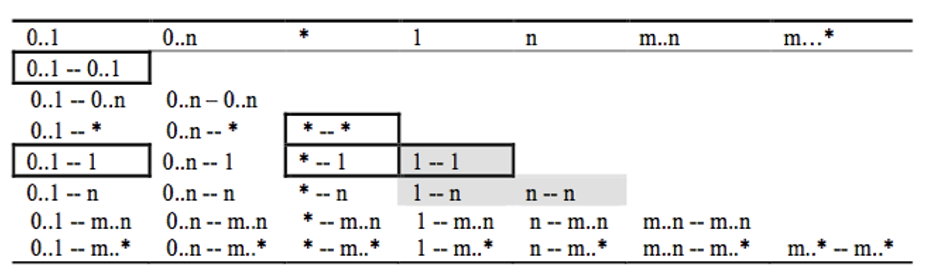


center h:500

### Associations

* Describe how instances of classes are linked at runtime
  + Bidirectional -- or
  + Unidirectional ->
* Multiplicity:
  + Bounds on the number of linked instances
* \* Or 0..\* 0 or more
* 1..\* 1 or more
* 1 Exactly 1
* 2 Exactly 2
* 1..3 Between 1 and 3
* 0..2 Up to 2

### Association Relationships



center h:400

### Association Relationships

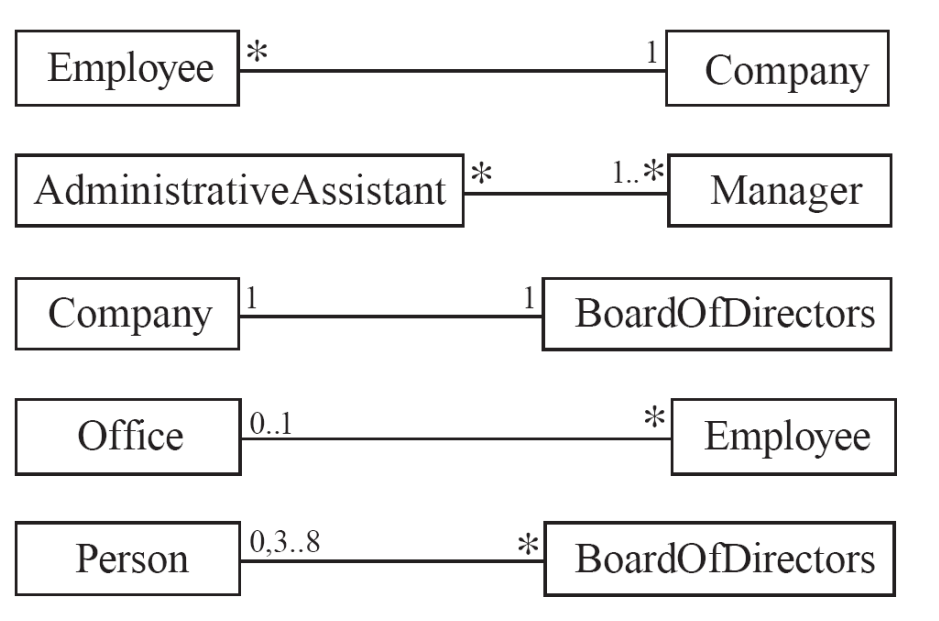
* Directional Associations

\* -> 0..1, \* -> 1, \* -> \*, \* -> m..n, \* - >n, \*->m..\* and\*->0..n.

* Symmetric Reflexive

0..1, 0..n, \*, 1, n, m..n,m..\*

### Basic UML associations



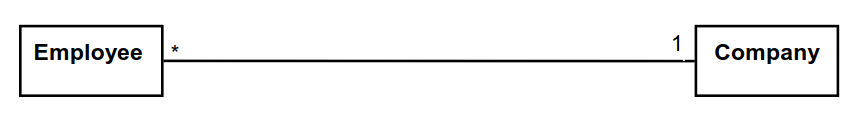
center h:500

### Many-to-one associations (1)

class Employee {  
id;  
firstName;  
lastName;  
}  
  
class Company {  
name;  
1 -- \* Employee;  
}

### Many-to-one associations (2)

* A company has many employees,
* An employee can only work for one company.
  + This company will not store data about the moonlighting activities of employees!
* A company can have zero employees
  + E.g. a ‘shell’ company
* It is not possible to be an employee unless you work for a company
* Let’s draw and write this in UMPLEOnline:



center h:150

### Role names (optional, in most cases)

* Allow you to better label either end of an association

class Person{  
id;  
firstName;  
lastName;  
}  
  
class Company {  
name;  
1 employer -- \* Person employee;  
}

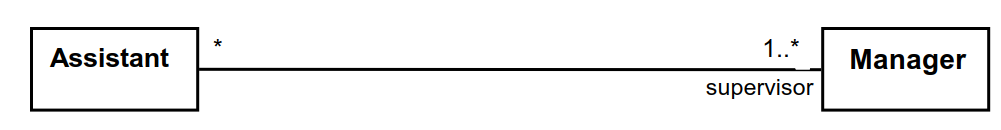
### Referential Integrity

* When an instance on one side of the association changes
  + The linked instances on the other side know …
  + And vice-versa
* This is standard in UMPLE associations, which are bidirectional

### Many-to-Many Associations

* An assistant can work for many managers
* A manager can have many assistants
* Assistants can work in pools working for several managers
* Managers can have a group of assistants
* Some managers might have zero assistants.
* Is it possible for an assistant to have, perhaps temporarily, zero managers?

[Open in UMPLE](https://cruise.UMPLE.org/UMPLEonline/?text=class%20Assistant%20%7B%7Dclass%20Manager%20%7B%20%201..*%20supervisor%20--%20*%20Assistant;%7D//$?%5BEnd_of_model%5D$?class%20Assistant%7B%20%20position%2049%2030%20109%2045;%7Dclass%20Manager%7B%20%20position%2073%20127%20109%2045;%7D)

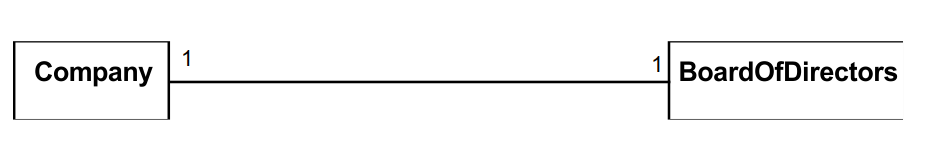


center h:150

### One-to-One Associations (Use cautiously)

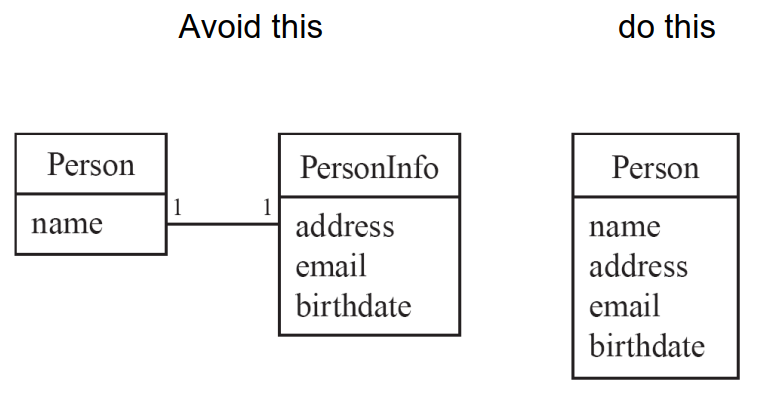
* For each company, there is exactly one board of directors
* A board is the board of only one company
* A company must always have a board
* A board must always be of some company

[Open in UMPLE](https://cruise.UMPLE.org/UMPLEonline/?text=class%20Company%20%7B%7Dclass%20BoardOfDirectors%20%7B%7Dassociation%20%7B%20%201%20Company%20--%201%20BoardOfDirectors;%7D//$?%5BEnd_of_model%5D$?class%20Company%7B%20%20position%2050%2030%20109%2045;%7Dclass%20BoardOfDirectors%7B%20%20position%2050%20130%20109%2045;%7D)



center h:150

### Typical erroneous use of one-to-one



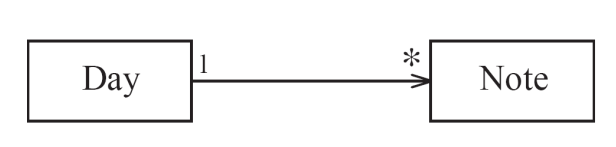
center h:500

### Unidirectional Associations

* Associations are by default bi-directional
* It is possible to limit the direction of an association by adding an arrow at one end
* In the following unidirectional association
  + A Day knows about its notes, but a Note does not know which Day is belongs to
  + Note remains ‘uncoupled’ and can be used in other contexts

class Day {  
\* -> 1 Note;  
}  
class Note {}

[Open in UMPLE](https://cruise.UMPLE.org/UMPLEonline/?text=class%20Day%20%7B%20%20*%20-%3E%201%20Note;%7Dclass%20Note%20%7B%7D//$?%5BEnd_of_model%5D$?class%20Day%7B%20%20position%2050%2031%20109%2045;%20%20position.association%20Day__Note%2030,46%2030,0;%7Dclass%20Note%7B%20%20position%2050%20131%20109%2045;%7D)



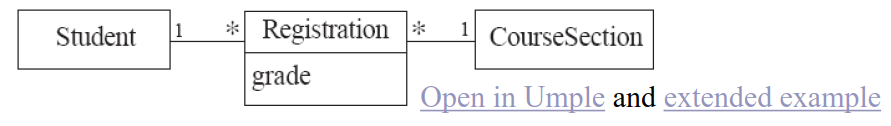
center h:150

### Association Classes

* Sometimes, an attribute that concerns two associated classes cannot be placed in either of the classes

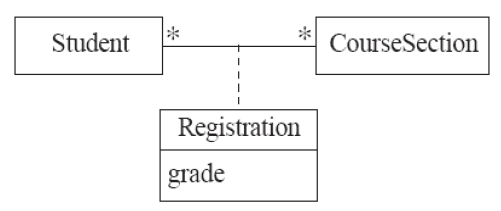
[Open in UMPLE](https://cruise.UMPLE.org/UMPLEonline/?text=class%20Student%20%7B%7Dclass%20CourseSection%20%7B%7Dclass%20Registration%20%7B%20%20*%20--%201%20Student;%20%20*%20--%201%20CourseSection;%7D//$?%5BEnd_of_model%5D$?class%20Student%7B%20%20position%2050%2030%20109%2045;%7Dclass%20CourseSection%7B%20%20position%2097%20203%20109%2045;%7Dclass%20Registration%7B%20%20position%2067%20123%20109%2045;%20%20position.association%20CourseSection__Registration%2084,45%2029,0;%7D)

and [Extended Example](https://cruise.UMPLE.org/UMPLEonline/?text=class%20Student%20%7B%7D%0A%0Aclass%20CourseSection%20%7B%20%0A%0A%201%20--%20*%20Evaluation%3B%0A%7D%0A%0Aclass%20Registration%20%7B%0A%20%20finalGrade%3B%0A%20%20*%20--%201%20Student%3B%0A%20%20*%20--%201%20CourseSection%3B%0A%7D%0A%0Aclass%20ComponentGrade%0A%7B%0A%20%20Integer%20value%3B%0A%20%20*%20--%201%20Registration%3B%0A%20%20*%20--%201%20Evaluation%3B%0A%7D%0A%0Aclass%20Evaluation%0A%7B%0A%20%20description%3B%0A%20%20Float%20weight%3B%0A%7D%0A%0A%2F%2F%24%3F%5BEnd_of_model%5D%24%3F%0A%0Aclass%20Student%0A%7B%0A%20%20position%2050%2030%20109%2045%3B%0A%7D%0A%0Aclass%20CourseSection%0A%7B%0A%20%20position%2088%20229%20109%2045%3B%0A%20%20position.association%20CourseSection__Evaluation%20110%2C43%202%2C0%3B%0A%7D%0A%0Aclass%20Registration%0A%7B%0A%20%20position%2052%20123%20145%2063%3B%0A%20%20position.association%20CourseSection__Registration%2071%2C63%2029%2C0%3B%0A%20%20position.association%20Registration__Student%2030%2C0%2030%2C46%3B%0A%7D%0A%0Aclass%20ComponentGrade%0A%7B%0A%20%20position%20305%20124%20124%2046%3B%0A%20%20position.association%20ComponentGrade__Registration%200%2C27%20146%2C26%3B%0A%20%20position.association%20ComponentGrade__Evaluation%2064%2C63%20102%2C0%3B%0A%7D%0A%0Aclass%20Evaluation%0A%7B%0A%20%20position%20252%20293%20149%2063%3B%0A%7D)



center h:100

* The following are nearly equivalent
  + The only difference:
    - in the association class there can be only a single registration of a given Student in a CourseSection



center h:200

### Association classes (cont.)

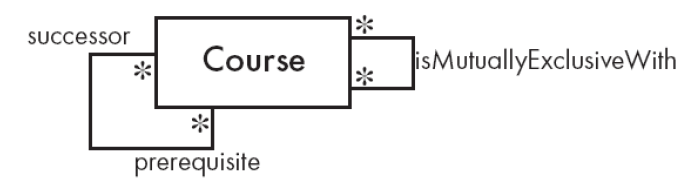
* UMPLE code

class Student {}  
class CourseSection {}  
associationClass Registration {  
\* Student;  
\* CourseSection;  
}

* Open in UMPLEOnline, and then generate code

### Reflexive Associations

* An association that connects a class to itself



center

[Open in UMPLE](https://cruise.UMPLE.org/UMPLEonline/?text=class%20Course%20%7B%20%20%20*%20self%20isMutuallyExclusiveWith;%7Dassociation%20%7B%20%20%20%20%20*%20Course%20successor%20--%20*%20Course%20prerequisite;%7D//$?%5BEnd_of_model%5D$?class%20Course%7B%20%20position%20122%2025%20109%2045;%7D)

class Course {  
\* self isMutuallyExclusiveWith; // Symmetric  
}  
  
association {  
\* Course successor -- \* Course prerequisite;  
}

### Inline vs. Standalone associations

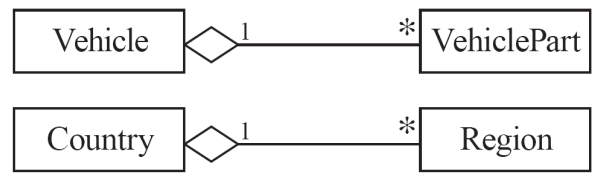
* The following are equivalent to allow flexibility:

class X {}  
class Y {  
1 -- \* X;  
  
}

class X {}  
class Y {}  
association {  
1 Y -- \* X;  
  
}

### Aggregation

* Aggregations are ordinary associations that represent part-whole relationships.
  + The ‘whole’ side is often called the assembly or the aggregate
  + This is a shorthand for association named isPartOf
  + UMPLE has no special syntax currently

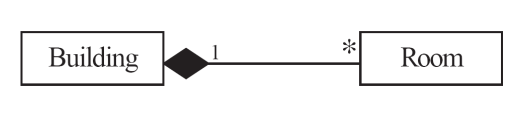


center

class Vehicle {  
1 whole -- \* VehiclePart part;  
}  
class VehiclePart{  
}

### Composition

* A composition is a strong kind of aggregation
  + If the aggregate is destroyed, then the parts are destroyed as well



class Building {  
1 <@>- \* Room;  
}  
class Room{  
}

### Sorted Associations

* Order objects in the association according to a specific key

class Academy {  
1 -- \* Student registrants sorted {id};  
}  
  
class Student {  
Integer id;  
name;  
}

* We will look at a more complete example in the User Manual

### A final word on associations

* More help and examples are in the user manual online at
  + http://associations.UMPLE.org

## References

* [UMPLE Tutorials](https://github.com/UMPLE/UMPLE/wiki/Tutorials)
* [UMPLE Github](https://github.com/UMPLE/UMPLE)
* [UMPLE Online](https://UMPLE.org)
* [UMPLE Documentation](https://cruise.UMPLE.org/UMPLE/)
* [UMPLE CSI5112– February 2018](http://www.site.uottawa.ca/~mgarz042/files/CSI5112-UMPLE.pdf)
* [UMPLE Tutorial: Models 2020 Web](https://cruise.UMPLE.org/presentations/UMPLEModels2020Tutorial/)
* [UMPLE Tutorial: Models 2020 Pdf](https://cruise.UMPLE.org/presentations/UMPLEModels2020Tutorial/UMPLETutForModels2020.pdf)

## References

* [Getting Started in UMPLE](https://cruise.UMPLE.org/UMPLE/GettingStarted.html)
* [Experiential Learning for Software Engineering Using Agile Modeling in UMPLE (Youtube)](https://www.youtube.com/watch?v=yif1clbrXnI&ab_channel=CSEETconf)
* [Experiential Learning for Software Engineering Using Agile Modeling in UMPLE (Slide)](https://cruise.UMPLE.org/presentations/UMPLETutorialCSEET2020.pdf)
* [Tomassetti Code Generation](https://tomassetti.me/code-generation/)