Lecture 2: OO and UML

Today's Plan

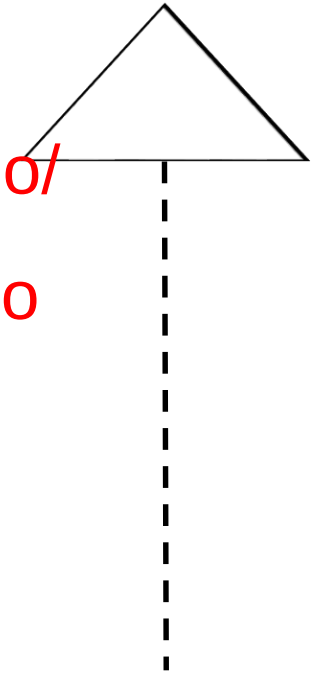
- 10:05-10:55:
  - Quiz 2 (assessed, correctness)
  - Important
- Break
- 11:05-11:55:
  - PI 2 (assessed, participation)
  - UML sequence diagram
  - Case-study: Airline
- Long break
- 13:00-14:00 OR 14:00-15:00 in BO1028: Lab 2 UML

QUIZ

- Go to TACTS
  - https://cl
- Join session: **1140**
  - prepare for quiz.
  - raise you hand if your are experiencing problems
  - 180 sec per question (I'll call out 2 min, 1 min and 30 sec)

Q2.1: UML Syntax [solo]

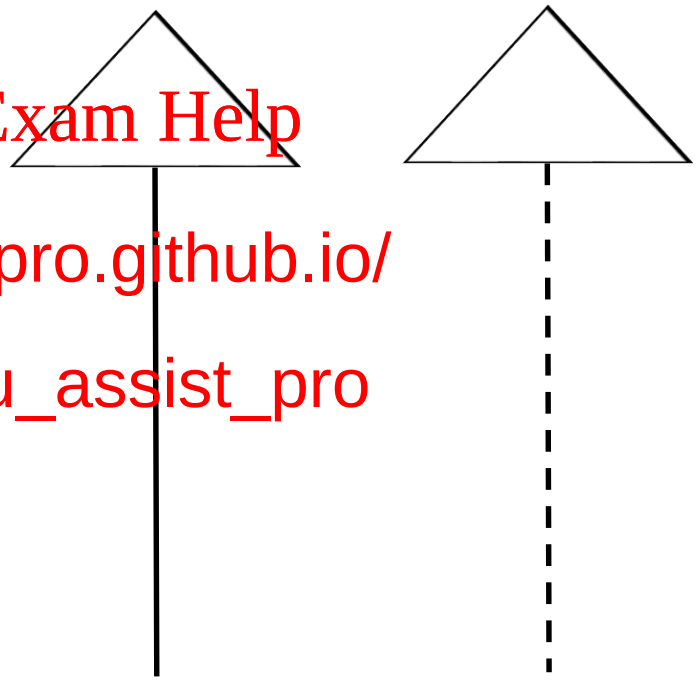
What does this arrow represent in UML?



- A: Inheritance relationship
- B: Interface relationship
- C: Uses relationships
- D: Requires relations
- E: Don't know

Q2.1: Interfaces Versus Inheritance (solution)

- Don't confuse arrows for inheritance and interfaces.
- Just need to memorise and get used to identifying different arrows.



Extends

B: Implements

Q2.2: UML Syntax [solo]

This box is used to describe:

- A: Attributes
- B: Instances
- C: Methods
- D: Objects
- E: Don't know

Q2.2: UML Syntax (solution)

This box is used to describe:

- A: Attributes
- B: Instances
- C: **Methods**
- D: Objects

Q2.3: UML Syntax [solo]

Interfaces are indicated in UML by:

- A: Italic Cl

- B: Bold ClassName
- C: Interface Label above Class Name
- D: A diamond-shaped arrow
- E: Don't know

Q2.3: UML Syntax (solution)

Interfaces are indicated in UML by:

- A: Italic Cl <– Incorrect: classes
- B: Bold ClassName <– All should be bold
- C: Interface Label above Class Name <– All interfaces should be labeled as such (correct)
- D: A diamond shaped arrow <– this describes a different kind of relationship: Aggregation and Composition

UML

- Class diagrams
  - describe classes and their relationships
- Interacti
  - show rms of how objects interact with each other
- State diagrams and activit
  - show how systems behave internally
- Component and deployment diagrams
  - show how the various components of systems are arranged logically and physically

Why is UML Important?

- One of the few ways to add tangibility to software
  - structure
  - docume
  - communication/collaboration
  - review
  - …
- Standard method of communication (also in Industry)
- Implementation
  - Can help in auto-generating code

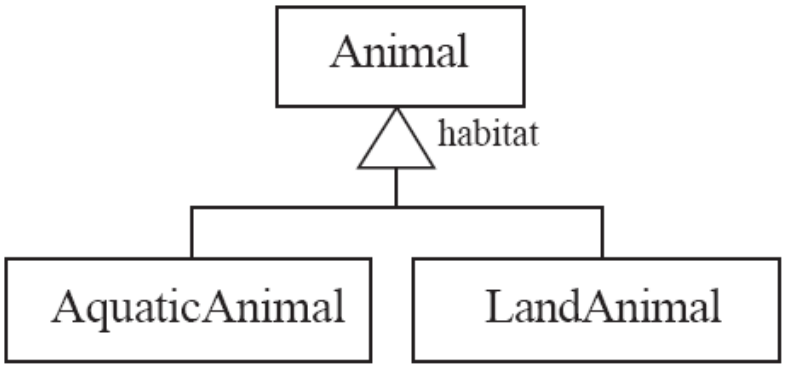
Usage

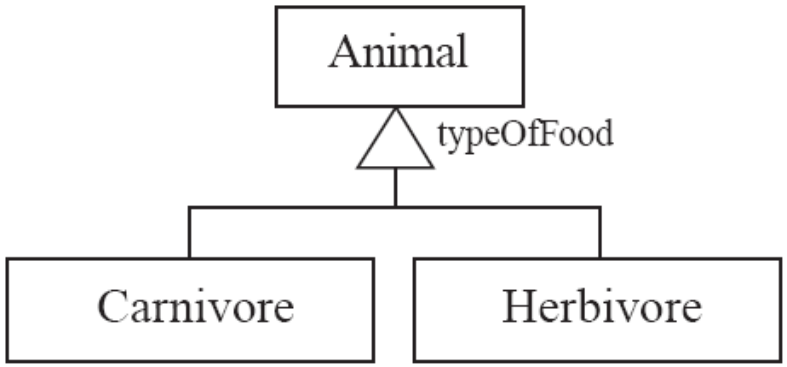
UML

- *The main symbols shown on class diagrams are:*
  - *Classes*
    - represent sets
  - *Generaliz*
    - group classes into inh rchies
  - *Associations*
    - represent linkages between instances of classes
  - *Attributes*
    - are simple data found in classes and their instances
  - *Operations*
    - represent the functions performed by the classes and their instances

Generalization

- Specializing a superclass into two or more subclasses
  - A *gene* roup of
    genera uperclass
  - The label (sometimes c *criminator*)
    describes the criteria u ecialization





Java: `extends`

Associations

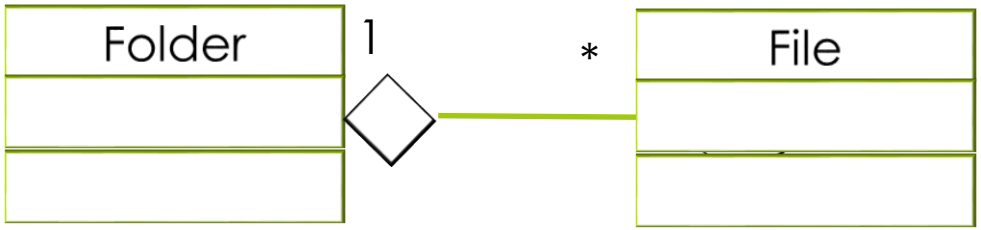
- One-to-many

- Many-to-many

- One-to-one – Consider is these should really be two entities

Associations in Java

Composition and Aggregation



Aggregation in Java

- A mechanism where an operation in an aggregate is implem egate perform that operati

- At the same time, prop arts are often propagated back to th

- Propagation is to aggregation as inheritance is to generalization.
  - The major difference is:
    - inheritance is an implicit mechanism
    - propagation has to be programmed when required

Associations v generalization

- **Associations** describe the relationships that will exist between
  - When a diagram generated from a class diagram, there will be an instance of *both* classes joined by an association
- **Generalizations** describe relationships between *classes* in class diagrams.
  - They do not appear in instance diagrams at all.
  - An instance of any class should also be considered to be an instance of each of that class's superclasses

PI 2 Question Time

- Please Join PACRS now:
  - https://
- Join session: **1141**

PI 2.1A: Question [solo!]

A course registration system has classes for ***Course*** *and* ***Student.***

These should be represented

A: Composition

B: Attribute

C: Aggregation

D: Inheritance

PI 2.1B: Question [group!]

A course registration system has classes for ***Course*** *and* ***Student.***

These should be represented

A: Composition

B: Attribute

C: Aggregation

D: Inheritance

PI 2.1: Solution

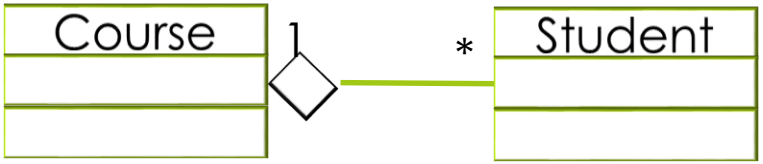
A course registration system has classes for ***Course*** *and* ***Student.***

These should b

A: Composition ← Does course ts, i.e. no course no students?

B: Attribute

C: Aggregation ← Are students part of a course. Can a student exist without a course?



D: Inheritance

PI 2.2A: Question

A grocery store is trying to refactor its system to better use generalization for categories its products. Which structure would you us

A: A generalization for each ed on the aisles in the store (for example Aisle 1,

B: A generalization for each product based on the category (for example Fruit, Dairy, Dry Goods, etc..)

C: A generalization called Product, which all products extend

D: Something else…

PI 2.2B: Question

A grocery store is trying to refactor its system to better use generalization for categories its products. Which structure would you us

A: A generalization for each ed on the aisles in the store (for example Aisle 1,

B: A generalization for each product based on the category (for example Fruit, Dairy, Dry Goods, etc..)

C: A generalization called Product, which all products extend

D: Something else…

PI 2.2: Solution

A grocery store is trying to refactor its system to better use generalization for categories its products. Which structure would you use?

A: A generalization for each product based on the aisles in the store, for exa "is a" relationship does not work here and they are sold on aisle 5, you can't say and an 5

B: A generalization for each product based on the category, for example Fruit, Dairy, Dry Goods, etc.. <- Do these categories behave differently in meaningful ways (if so then correct); but do not go overboard with subclasses

C: A generalization called Product, which all products extend
<- typically this would be a sensible generalization; here the specific product info would be saved on as attributes

PI 2.3A: Question

A credit card company has a database of all its customers, including those who no longer hold credit account with the company. Customers can have multiple credit accounts.

How would you annotate the following association?

A.

B.

C.

D.

PI 2.3: Solution

A credit card company has a database of all its customers, including those who no longer hold credit account with the company. Customers can have multiple credit accounts.

How would you annotate the following association?

A.

B.

C.

D.

Sequence Diagrams

- Modeling behavior in an interactive system
- Instances o

- Messages between instance actors

- In general, sequence diagrams come after class modeling is basically complete

- ***Notice:*** Necessary syntax included here (and in following lectures), but you may want to read OOSE Chapter 8.

Simple Sequence Diagram

Key Syntax

- Instances are underlined, labeled like this:
  - `instance`

- Lifeline is a

- Messages are horizontal line
  - Synchronous message with filled arrowhead
  - Asynchronous message with unfilled arrowhead
- Your books also recommends dash line for "create" message, but it's not standard by IBM specification
  - A good overview at http://www.ibm.com/developerworks/rational/library/3101.html

Class Responsibilities

- Setting and Getting
- Creating and In
- Loading and Sa
- Destroying
- Adding and Deleting (associations)
- Copying, converting, transforming, transmitting, outputting
- Computing
- Navigating and Searching
- Specialised Work

Case-study: Airline Reservation System (OOSE: Appendix C)

- *Ootumlia Airlines runs sightseeing flights from Java Valley, the capitol tion system keeps track of pa in specific seats on various flights, as well as p ill form the crew. For the crew, the system m what everyone does, and who supervised whom. Ootmulia Airlines runs several daily numbered flights on a regular schedule. Ootumlia Airlines expects to expand in the future, and therefore the system needs to be flexible; in particular, it will be adding a frequent flier plan.*

Domain Modeling: Classes

- Identify all the **nouns and noun phrases**.
- Which sho

Domain modelling: Attributes & Associations

- Which attributes and associations should we include?

Associations

Generalization

Domain modelling: Responsibilities and Operations

- What Responsibilities does each class have?
  - Identify v

Responsibilities

- Creating a new Regular Flight
- Searching
- Modifying t Flight
- Creating a Specific Flight
- Cancelling a Specific Flight
- Cancelling a Booking
- …

You can review this complete example at home

- See Appendix C description of "Airline Reservation System" an 5.5 for more details

Summary - UML

- How readable is the diagram? Can you easily figure out "what's going on"?
- By looking at th clear vision of the underlying Java
- Does each class in the diagram orrespondence to a class in Java?
- For non-specific (not implements or extends) relationships, how would these be represented in code? Are they implicit or explicit?
- Can you see coupling and cohesion in the code? Can you describe it?

Lab 2: Domain Modeling with UML

Lab 2: Available on Moodle (12:30)

- 13:00-14:00

- 14:00-15:00 BO1028

Next Week (Week 3)

- Design Pattern mania
- Reading:
- Quiz on ***Design Patterns (n*** ***ill follow on Moodle)***

<EOL>