Class Diagrams

Dr. Isaac Griffith Idaho State University

Outcomes

Idaho State University | Computer Science

After today's lecture you will:

- Have an understanding of the different types of UML Diagrams
- Understand the types of relationships between classes/objects
- Be capable of using these relationships in Class Diagrams
- Be capable of translating basic Class Diagrams to working code.

UML Class Diagrams

CS 2263

What is UML?

Idaho State University | Computer Science

- UML: pictures of an OO system
  - programming languages are not abstract enough for OO design
  - UML is an open standard, lots of companies use it
- What is legal UML?
  - a *descriptive* language: rigid formal syntax (like programming)
  - a *prescriptive* language: shaped by usage and convention
  - it's okay to omit things from UML diagrams if they aren't needed by team/supervisor/instructor

Uses for UML

Idaho State University | Computer Science

- As a sketch: to communicate aspects of a system
  - **Forward Design:** doing UML before coding
  - **Backward Design:** doing UML after coding as documentation
  - Often done on whiteboard or paper, but with current collaborative tools, it has become more and more digital
  - Used to get rough selective ideas
- As a blueprint: a complete design to be implemented
  - Sometimes done with CASE (Computer-Aided Software Engineering) tools
- As a programming language: with the right tools, code can be auto-generated and executed from UML
  - Only good if this is faster than coding in a “real” language

UML Class Diagrams

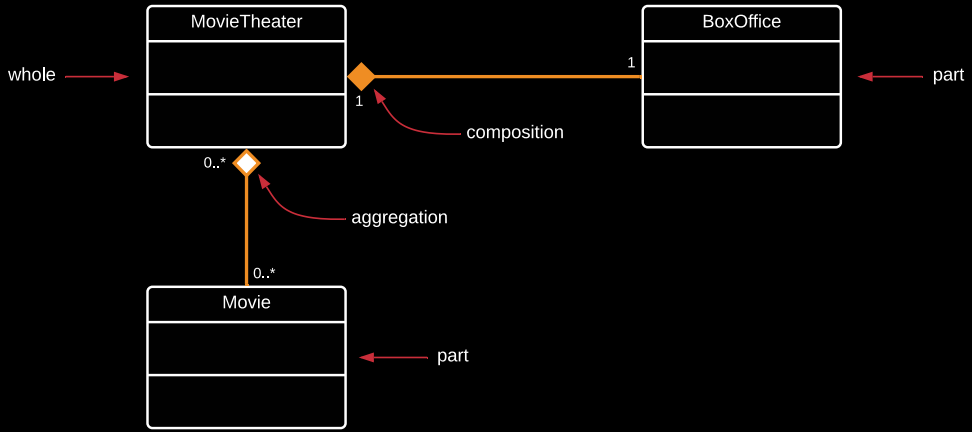
Idaho State University | Computer Science

- **UML class diagram:** a picture of
  - the classes in an OO system
  - their fields and methods
  - connections between the classes
    - that interact or inherit from each other
- Not represented in a UML class diagram:
  - details of how the classes interact with each other
  - algorithmic details; how a particular behavior is implemented

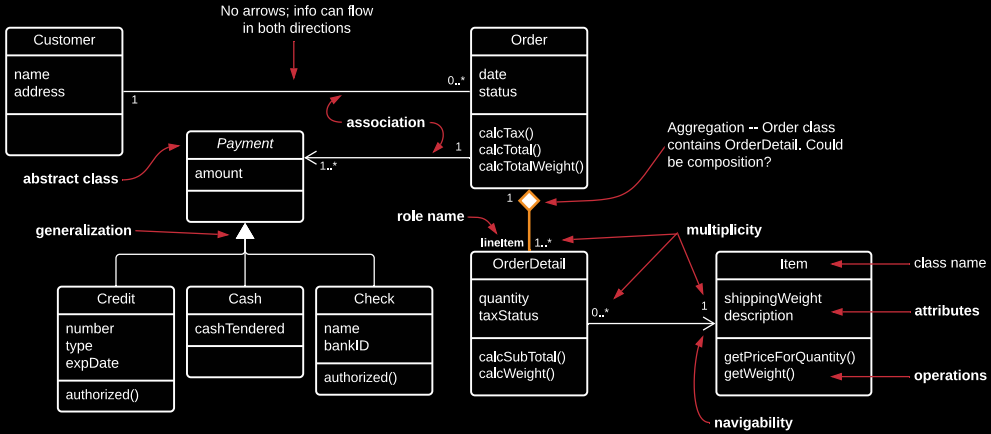
Examples

CS 2263

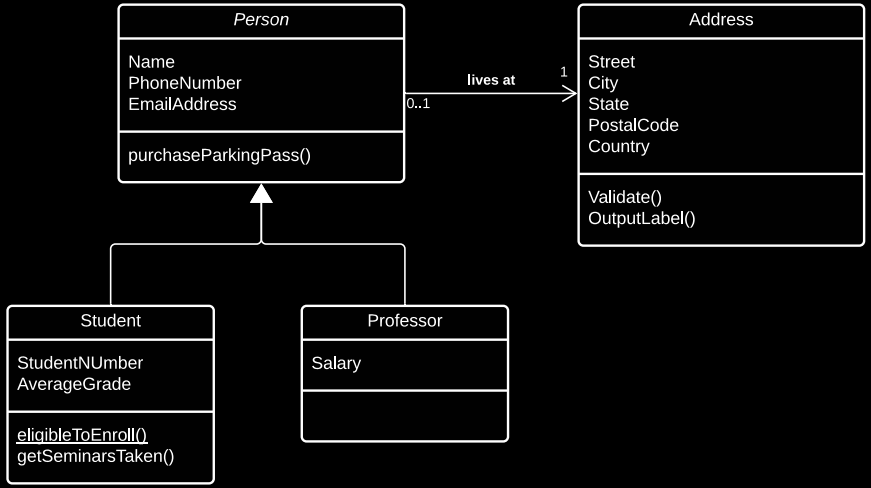
Movie Theater



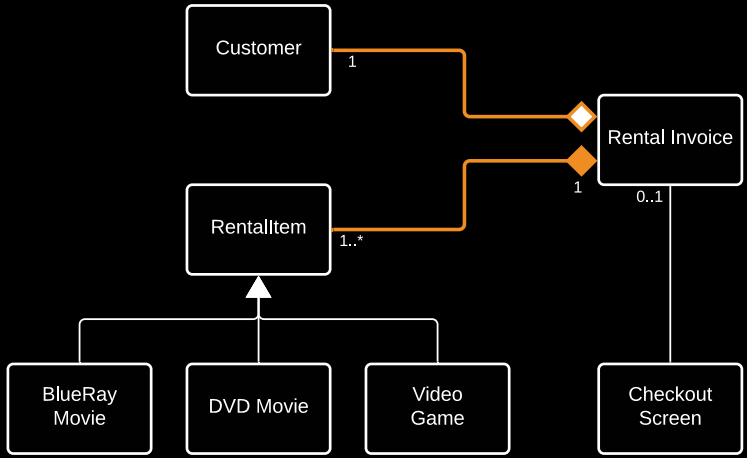
Order Handling



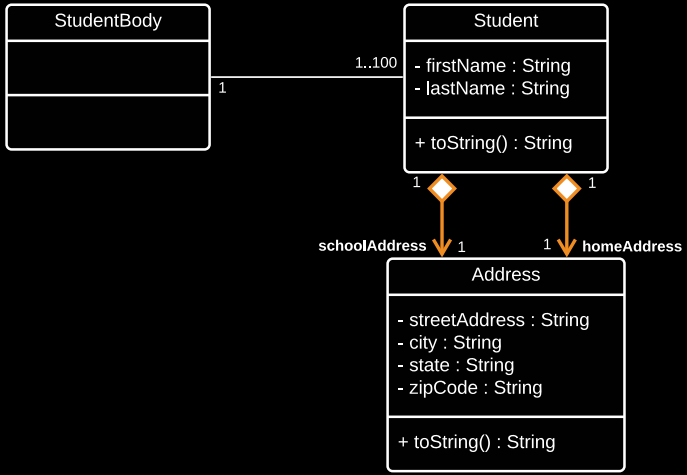
People


Person
Name
PhoneNumber
EmailAddress
purchaseParkingPass()
lives at
0..1
1
Address
Street
City
State
PostalCode
Country
Validate()
OutputLabel()
Student
StudentNUmber
AverageGrade
eligibleToEnroll()
getSeminarsTaken()
Professor
Salary

Retro Video Store



Students


StudentBody
Student
- firstName : String
- lastName : String
+ toString() : String
1..100
1
1
1
schoolAddress
1
1
homeAddress
Address
- streetAddress : String
- city : String
- state : String
- zipCode : String
+ toString() : String

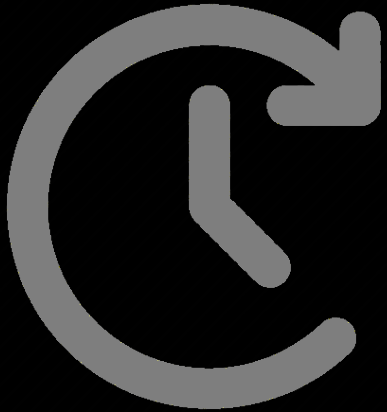
Common Pitfalls in Modeling

Idaho State University | Computer Science

- Confusing basic class relationships (i.e., IS-A, HAS-A, IS-Implemented-Using)
- Poor use of Inheritance
  - Violating encapsulation and/or increasing coupling
  - Base classes do too much or too little
  - Not preserving base class invariants
  - Confusing interface inheritance with implementation inheritance
  - Using multiple inheritance to invert IS-A

For Next Time

- Review this Lecture
- Watch the video on creating class diagrams with LucidCharts
- Come to Class
- Read Chapter 4.1 - 4.4
- Read Java 9 Modules Tutorial
- Continue working on Homework 02



Are there any questions?