Modeling
Administrative stuff

• Quiz
• Project proposal due this Thursday
Outline

• Review
  • What are requirements?
  • Use case diagram
  • Activity diagram

• Today’s
  • OO
  • Class diagram
  • Sequence diagram
What are requirements?

- The services the software should provide
  +
- The constraints the software should follow
  +
- Functional requirements
  +
- Non-functional requirements
What are non-functional requirements?
How to find out the requirements
System modeling
Use case diagrams

- A diagram includes
  - Actors
  - Use cases
  - System boundary

Use case text

• Use case name
• Use case priority
• Main scenario
  • Steps
• Extensions
  • Steps
Activity diagram

- Start, end
- Flow
- Fork/join
- Decision
- ...
Design

OO
Class Diagram
Sequence Diagram
What is the first P.L. you learned?
Object-Oriented Programming, Classes

• Class
  • Data + Operation

• Encapsulation
• Polymorphism
• Inheritance

• Enhance modularity!
Encapsulation

• “the packing of data and functions into a single component. The features of encapsulation are supported using classes. It allows selective hiding of properties and methods in a class by building an impenetrable wall to protect the code from accidental corruption.”

• Example
Polymorphism

• “to process objects differently depending on their data type or class. More specifically, it is the ability to redefine methods for derived classes”
• “the provision of a single interface to entities of different types.”
• Examples
Inheritance

• “a mechanism for code reuse and to allow independent extensions of the original software via public classes and interfaces.”

• Examples
Class diagram

• Describes the types of objects in the system
• Describes the static relationships among them
How to decide/design classes?

- Data+operation
Components of class diagrams

• Class name
• Class properties
  • Attributes
  • Associations (could be bi-directional)
    visibility name : type [multiplicity] = default {property-string}
• Class operations
  Visibility name (parameter list) : return-type {property-string}
• Generalization
  • Inheritance (subclass, super class, interface, …)
• Dependency
• Constraints { }
How to turn class diagram to code

• A private attribute → ??
• A * attribute/association → ??
• Class declaration
  • Some attributes may not map to fields
Example
Advanced Class-Diagram Features

• Composition◆ vs. Association◇
• Abstract class
• Template class
What are the constraints to set?

- Assertion
  - Pre-condition
  - Post-condition
  - Invariant
Sequence diagram

• Describes how objects collaborate/interact with each other in one scenario
Components of sequence diagram

• Participants
• Life-line
• Activation bar
• Message

• Creating and deleting object
• Loops and conditionals

http://en.wikipedia.org/wiki/Sequence_diagram
Example
Other diagrams

• CRC card
• State diagram
  • A single object across multiple use cases
Summary

• Requirement specification template
• Use-case diagram
• Activity diagram
• Class diagram
• Sequence diagram
Example 1: facebook-ish forum

• What are the use cases?
• What are the steps and scenarios of “send message”
• What is the activity diagram for that?
• What is the class diagram for that?
• What is the sequence diagram?
Example 2: Amazon-ish software

• What are the actors and use cases?
• What are the scenarios and steps for “check-out”?
• What is the activity diagram?
• What is the class diagram?
• What is the sequence diagram?