Object-Oriented Design VectorGraphics

CSSE 221

Fundamentals of Software Development Honors

Rose-Hulman Institute of Technology

ROSE-HULMAN INSTITUTE OF TECHNOLOGY

Announcements

- Turn in HW4 and Fifteen UML now
- Exam 1 returned
  - Solution posted outside my office
  - Opportunity for questions tomorrow
- Capsules round 1 returned
  - Again, great work researching!
- Start capsules round 2 tomorrow

Schedule

- OO software development in Java.
- 18 chapters in text!
  - Ch. 1-16.4, 18, 20
  - Only 6 more left…
- Lots of programming, including:
  - Each week' structured around a prog. assignment
  - 1 bigger team project
- Researching and presenting course material to classmates
- Intro to C

	Topic	Project	Indep
1	Interfaces	BigRational	
2	Inher & Poly	BallWorlds	Research
3	GUI	Fifteen	Research
4	Lists	VectorGraphics	Demo
5	Data Structs	Markov	Demo
6	Simulation	Simulation	
7	Sorting	Simulation	Present
8	Searching	Simulation	Present
9	C Basics	C Projects	
10	Linked Lists	Linked Lists	

This week: VectorGraphics

- Monday:
  - More about software design and planning
  - Project workday
- Tuesday:
  - Lists and Iterators (capsule)
  - Review big-Oh
- Thursday:
  - Threads (capsule)
  - Project workday

A team project to create a scalable graphics program.

Vector Graphics Assignment

http://www.rose-hulman.edu/class/csse/binaries/VideoDemos/VectorGraphics220.mov

ROSE-HULMAN
INSTITUTE OF TECHNOLOGY

Work time today

- Now:
  - Read the specification
  - Sketch out some screen layouts
- Design (CRC cards and UML) due Thursday
- Code due Monday
- In ~15 minutes
  - How to create CRC cards
  - Review of UML

A practical technique

Object-Oriented Design

ROSE-HULMAN
INSTITUTE OF TECHNOLOGY

Object-Oriented Design

- We won't use full-scale, formal methodologies
  - Those are in later SE courses
- CRC cards → UML class diagram
- Like any design technique, the key to success is practice

Key Steps in Our Design Process

1. **Discover classes** based on requirements
   - Come from **nouns** in the problem description
2. **Determine responsibilities** of each class
   - Come from **verbs** associated with the classes
3. **Describe relationships** between classes:

   is-a, has-a

May…

Represent single concepts

`Circle, BigRational`

Represent visual elements of the project

`ColoredPanel`, `GameButton`

Be abstractions of real-life entities

`BankAccount`, `TicTacToeBoard`

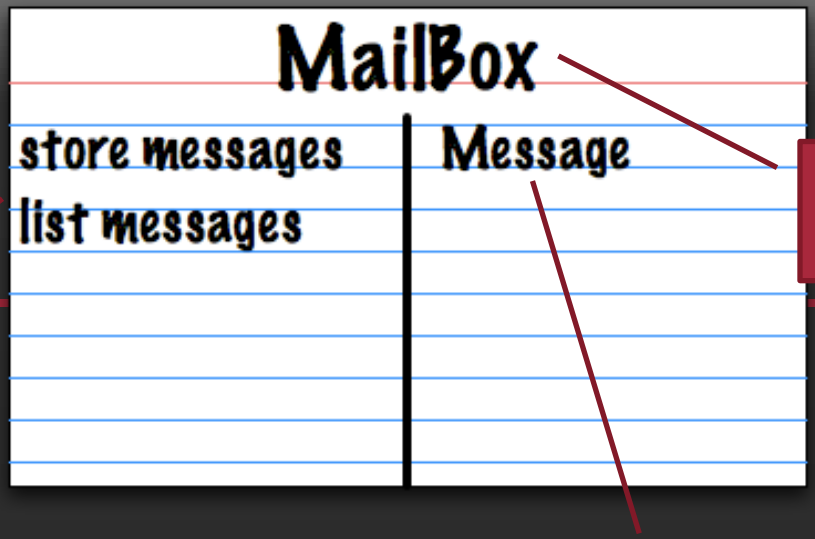
Be actors

`Scanner`

Be utilities

`Math`

CRC Card Technique



1. Pick a responsibility of the program
2. Pick a class to carry out that responsibility
   - Add that responsibility to the class's card
3. Can that class carry out the responsibility by itself?
   - Yes → Return to step 1
   - No →
     - Decide which classes should help
     - List them as collaborators on the first card
     - Add additional responsibilities to the collaborators' cards

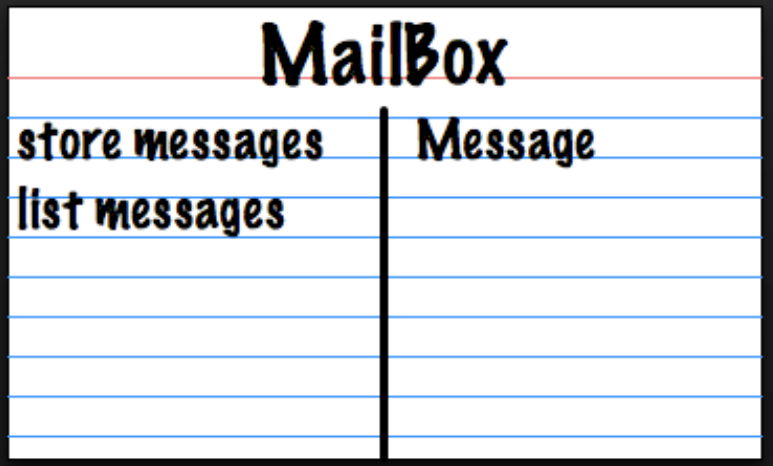
CRC Card Tips

- **Spread the cards out** on a table
  - Or sticky notes on a whiteboard instead of cards
- **Use a "token"** to keep your place
  - A quarter or a magnet
- **Focus on high-level responsibilities**
  - Some say < 3 per card
- **Keep it informal**
  - Rewrite cards if they get to sloppy
  - Tear up mistakes
  - Shuffle cards around to keep "friends" together

Make CRC cards for your VectorGraphics project

1. Pick a responsibility of the program
2. Pick a class to carry out that responsibility
   - Add that responsibility to the class's card
3. Can that class carry out the responsibility by itself?
   - Yes → Return to step 1
   - No →
     - Decide which classes should help
     - List them as collaborators on the first card
     - Add additional responsibilities to the collaborators' cards

- **High cohesion**
- **Low coupling**
- **Immutable where practical**
  - **Document where not**
- **Inheritance for code reuse**
- **Interfaces to allow others to interact with your code**



Convert your CRC Cards to a UML class diagram

- Classes stay classes
- Responsibilities become properties (methods)
- If attributes (fields) are obvious, add them
  - Who stores the list of shapes?
- Collaborators are usually has-a relationships
- If is-a relationships are obvious, add them

Summary of UML Class Diagram Arrows