From Domain Classes and Use Cases to Design Classes

Exercise

- **I have a sealed room in which some triangles are magically suspended in the air. Their position and orientation is random. There is also a ball in the room. It has some position and velocity at time t.**
- **Describe how the ball moves between time t and t+Δt (use case)**

Exercise

- **I have a sealed room in which some triangles are magically suspended in the air. Their position and orientation is random. There is also a ball in the room. It has some position and velocity at time t.**
- **Build a domain model for this system. (Class diagram)**

CRC Cards Technique (Responsibility-Driven Design)

- Informal, non-detailed
- Used for group brain-storming
- End result is a first cut at **classes** for an object-oriented model
- Not intended to provide a **complete** design

CRC

- **C**: **C**lasses
- **R**: **R**esponsibilities
- **C**: **C**ollaborations

The Basic Idea

- Develop set of index cards.
- Each card represents one class.
- A card contains:
  - The name of the class.
  - The responsibilities of the class.
  - Collaborations: other classes with which this class inter-operates, in conjunction with the attendant responsibility.

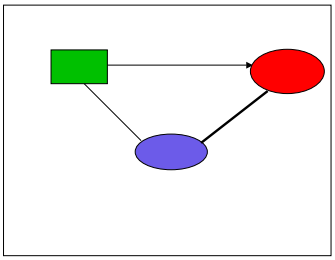
Image of CRC cards

Class Name	super class sub-classes
Responsibilities	Collaborations
_____	_____
_____	_____
_____	_____

Limiting the size of a card is an attempt at preventing the class from becoming too complex.

Sample Application: A graph-drawing program

Possible screen image



Typical Application Use-Cases:

- **Draw shape**
- **Move shape**
- **Resize shape**
- **Connect shapes**
- **Erase shape**
- **Erase connector**

Example of CRC card for a graph-drawing program (1)

Class → Shape

Example of CRC card for a graph-drawing program (2)

Responsibilities →

Shape
Remember size
Remember position
Remember fill color
Remember border
Remember connectors
Change size
Change position

Example of CRC card for a graph-drawing program (3)

Shape	
Remember size	
Remember position	
Remember fill color	
Remember border	→ Line
Remember connectors	→ Connector
Change size	
Change position	

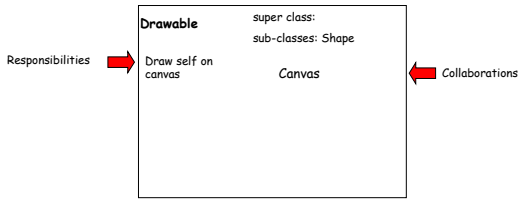
← Collaborations

Example of CRC card for a graph-drawing program (4)

Shape	super class: Drawable sub-classes: Rect, Oval, Group
Remember size	
Remember position	
Remember fill color	
Remember border	Line
Remember connectors	Connector
Change size	
Change position	

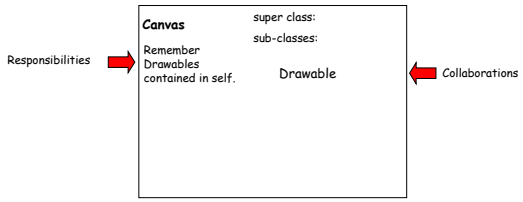
← Super- and Sub-classes

Example of CRC card
for a graph-drawing program (5)



Note: The Drawable doesn't necessarily need to *remember* a Canvas, since the Canvas could be passed as an argument to the *draw* method.

Example of CRC card
for a graph-drawing program (6)



Note:

- Responsibilities are usually for ***members*** (objects) of the class rather than the class itself, although
- Class-wide responsibility is possible (corresponding to **static** method)

Attribute Value vs. Object

- An object of a class typically has one or more ***attributes***.
- Attributes have ***values*** that specify or describe the object.
- A value might or might not deserve the distinction of being an object itself.
- A would-be attribute that is object-valued is actually a ***collaboration***.

Once the CRC cards are constructed …

- Team can engage in **role-playing** to verify that use-case **scenarios** make sense for chosen CRC.
- Each person can role-play one or more class cards.
- If something doesn't work, change the class accordingly.
- Revision of use-cases might also be indicated.

Exercise

• I have a sealed room in which some triangles are magically suspended in the air. Their position and orientation is random. There is also a ball in the room. It has some position and velocity at time t.

•Construct CRC cards for this system.