Lecture 4: Graphics, Events, and ArrayLists

CS 62
Fall 2015
Kim Bruce & Michael Bannister

Quiz Friday!!

Lab today

- Silver Dollar game
- Come prepared
  - ArrayList
  - graphics

Correction on Graphics

- Posted on Piazza
- Bottom line:
  - repaint works as explained: event on queue, update (erase), paint
  - obscuring and revealing window handled by OS (no paint)
  - minimizing and opening does not erase, but does call paint
- Has no impact on your lab, because you call repaint after each move of a coin.

Lab Stuff

- toString returns String
  - Does not print anything!
- See definition of interface List<E> on-line
  - You will be using ArrayList (see next example) rather than an array (mainly just to get used to it!)

Lab Stuff

- toString returns String
  - Does not print anything!
- See definition of interface List<E> on-line
  - You will be using ArrayList (see next example) rather than an array (mainly just to get used to it!)
**PostItStdApplication**

- Sophisticated
  - JFrame contains two JPanels
  - JFrame uses BorderLayout, so add controls to JPanel in SOUTH, canvas in CENTER of contentPane of JFrame
    - See GUI cheat sheet for details
    - DrawingCanvas extends JPanel — contains paint method

**PostIt class**

- Represents the rectangles being dragged:
  - Contains accessor and mutator methods to allow it to be manipulated by drawing program.
  - Could add features (title bar, go-away box) without affecting PostItApplication code.

**PostItApplication**

- PostItApplication class responsible for
  - setting up the GUI
  - Responding to button pressed and menu selections
  - Sets up ArrayList of items on canvas.
- Class has 3 inner classes
  - DrawingCanvas
  - DrawingMouseListener
  - DrawingMouseMotionListener
  - Inner classes have access to private features of containing class

**Inner classes**

- DrawingPanel extends JPanel
  - Associates listeners for mouse actions on the panel
  - Responsible for repainting the screen
- DrawingMouseListener and DrawingMouseMotionListener
  - Responsible for responding to mouse actions by changing the items in the ArrayList.
Event-Driven Programming

Handling Mouse Events

- If want program to react to mouse press, click, or release on a component
  - send `addMouseListener(mlo)` to component (usually in the constructor of the component)
  - See `PostItApplication.java`
  - For motion or drag, send `addMouseMotionListener(mlo)`
- When user presses mouse on a component
  - Computer looks for registered “MouseListener” for component or its containers.
  - If found, sends `mousePressed(evt)` to listener

Listener

- object designated as mouse listener must
  - implement `MouseListener` (& implement `mousePressed`, `mouseReleased`, & `mouseClicked`) or
  - extend `MouseAdapter` (which has default implementations of all 3)
- Second is easier unless class already extends another. *Can only extend one class in Java*
- Similarly, for mouse motion listener
  - implement `MouseMotionListener` or
  - extend `MouseMotionAdapter`

GUI Objects & Events

- Similar to handling mouse events, but must also install components in a container.
- See GUI cheat sheet in Documentation & Handouts.
Listeners in PostItApplication

- Main class (this) is listener for button and choice. Set up when GUI items constructed.
- Special listener objects for mouse actions. Set up by DrawingCanvas since listening for actions on that object.

List Operations

- Review list operations from library interface List<E> in Java 8 documentation.
  - Bailey's List<E> is slightly different.
  - Think about how implement with array.

ArrayList

- Focus on implementation
- See Bailey's ArrayIndexList<E>
  - Similar to Java 8's ArrayList
  - Instance variables:
    - elts: array instance variable,
    - eltsFilled: number of slots filled.
- Some operations very cheap:
  - size, isEmpty, get, set take constant time (no search)
- Others more expensive