

# Lecture 4: Standard Java Events & List

CS 62  
Fall 2017  
Kim Bruce & Alexandra Papoutsaki

## Lab & Assignment I

- Strip with 12 squares & 5 silver dollars placed randomly on the board.
- Move silver dollars to fill 5 leftmost squares
  - Coins move only to the left.
  - No coin may pass another.
  - No square may hold more than one coin.
- Last person to move wins.
- Complete description in text.

## Read Lab & Assignment Before Lab Today!

*Rest of today's lecture is info for lab and assignment!*

## ArrayList Specification

- Class `ArrayList<E>` implements `List<E>`
- Important methods:
  - `add`, `get`, `set`, `indexOf`, `isEmpty`, `remove`, `size`, `contains`, `clear`
  - `size`, `isEmpty`, `get`, `set` take constant time
  - `add` (to end) is “amortized constant” time
- See javadoc at
  - <http://docs.oracle.com/javase/8/docs/api/>

*See PostIt example later!*

## Last Time: Java Graphics

For details, see document on course web page associated with lecture  
Also see GUI cheat sheet in documentation and handouts section.

## Overview

- Graphical User Interfaces (GUI)
  - JFrame (*window*), JPanel (*grouping*) *Done!!*
  - JButton, JTextField, JSlider, JComboBox, ...
- Events:
  - Generated by mouse actions, button clicks, etc
  - Use ActionListener, MouseMotionListener, ActionListener, etc. to respond
- Graphics
  - Drawing items on the screen - *Done*

## General graphic applications

- Create an extension of component (either JPanel, JFrame, or JApplet) and implement paint method in the subclass.
  - See main method of demo to get window to show
  - At start of paint method cast g to Graphics2D to get access to new methods
- Call repaint() on component every time make a change.
  - Causes OS to schedule call of paint in event queue
  - Called automatically if window obscured and revealed

## PostItStdApplication

- More sophisticated
  - JFrame contains two JPanels
  - JFrame uses BorderLayout, so add controls to JPanel in SOUTH, drawing canvas in CENTER of the JFrame
    - Ignore controls for now.
    - See GUI cheat sheet for details
  - DrawingCanvas extends JPanel -- contains paint method
  - Note use of ArrayList to hold PostIts.

## 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

- DrawingCanvas 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 buttons 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.