

Lecture 4: Lists and listeners

CS 62

Spring 2018

Alexandra Papoutsaki & William Devanny

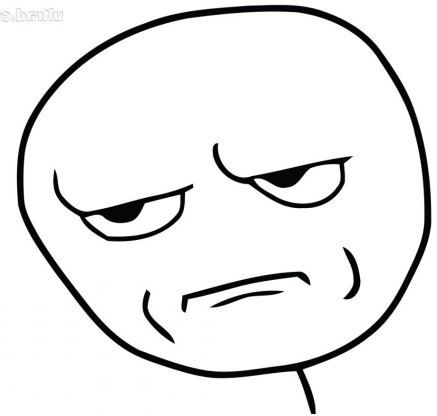
Lab and Assignment 1

- Strip with 12 squares and 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 textbook.

Arrays

- `int arr[] = new int[10]`
- Hold a sequence of primitives or objects.
- Public instance variable `length`
- Fixed length
- Don't play nice with generics

td.moe.brain



ArrayList

- `import java.util.ArrayList`
- `class ArrayList<E> implements List<E>`
- Important methods:
 - `add`, `get`, `set`, `indexOf`, `isEmpty`, `remove`, `size`, `contains`, `clear`
 - `size`, `isEmpty`, `get`, `set` → constant time
 - `add(E e)` → "amortized constant" time
- See javadoc at: <https://docs.oracle.com/javase/8/docs/api/>
- Text uses `Vector` instead of `ArrayList`.
 - `ArrayList` more efficient if no concurrency

Event-Driven Programming

PostItApplication

- More sophisticated.
- **JFrame** contains two **JPanel**s.
- **JFrame** uses **BorderLayout**, so add controls to **JPanel** in **SOUTH**, drawing canvas in **CENTER** of the **JFrame**.
- **DrawingCanvas** extends **JPanel** -- contains paint method
 - Note use of **ArrayList** to hold **PostIts**.

PostIt

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

Java.awt.event

- Classes: `ActionEvent`, `MouseEvent`, `KeyEvent`
- Listener Interfaces: `ActionListener`, `MouseListener`
- Listener Adapters: `MouseAdapter`, `KeyAdapter`

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*

Action Listeners

- `class MyClass implements ActionListener{...}`
- Register an instance of the event handler class as a listener on one or more components.
- `someComponent.addActionListener(instanceOfMyClass);`
- Implement the methods in listener interface.
- `public void actionPerformed(ActionEvent e) {
...//code that reacts to the action... }`

Inner Classes

- **DrawingCanvas** extends **JPanel**
 - Associates listeners for mouse actions on the canvas
 - Responsible for repainting the screen
- **DrawingMouseListener** and **DrawingMouseMotionListener**
 - Responsible for responding to mouse actions by changing the items in the **ArrayList**.

Handling Mouse Events

- If you 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**

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** in Java 8 documentation.
 - Bailey's List is slightly different.
- Think about how to implement with array.