Lecture 22. Listening to events on a GUI
(and development of a loop)
Sec. 17.4 contains this material. Corresponding lectures on ProgramLive CD is a better way to learn the material.

Developing a loop
// Set x to length of largest segment of equal values in b[0..n-1].
// Precondition: x is sorted (in ascending order)
0 1 2 3 3 3 5 5 5 7 8 8 8 8 9
n 14
x

Listening to events: mouseclick, mouse movement into or out of a window, a keystroke, etc.
• An event is a mouseclick, a mouse movement into or out of a window, a keystroke, etc.
• To be able to “listen to” a kind of event, you have to
  1. Write a method that will listen to the event.
  2. Let Java know that the method is defined in the class.
  3. Register an instance of the class that contains the method as a listener for the event.

We show you how to do this for clicks on buttons, clicks on components, and keystrokes.

A JPanel that is painted
• The content pane has a JPanel in its CENTER and a “reset” button in its SOUTH.
• The JPanel has a horizontal box b, which contains two vertical Boxes.
• Each vertical Box contains two instances of class Square.
• Click a Square that has no pink circle, and a pink circle is drawn.
• Click a square that has a pink circle, and the pink circle disappears.
• Click the rest button and all pink circles disappear.
• This GUI has to listen to:
  (1) a click on a Button
  (2) a click on a Square

why men think “computer” should be a feminine word
1. No one but their creator understands their internal logic.
2. The native language they use to talk with other computers is incomprehensible to everyone else.
3. Even the smallest mistakes are stored in long term memory for possible later retrieval.
4. As soon as you commit to one, half your paycheck goes for accessories for it.

why women think “computer” should be a masculine word
1. In order to do anything with them, you have to turn them on.
2. They have a lot of data but still can’t think for themselves.
3. They are supposed to help you solve problems, but half the time they ARE the problem.
4. As soon as you commit to one, you realize that if you had waited a little longer, you could have gotten a better model.

```java
// Set x to length of largest segment of equal values in b[0..n-1].
// Precondition: x is sorted (in ascending order)
0 1 2 3 3 3 5 5 5 7 8 8 8 8 9
n 14
x
```
import java.awt.event.*;  
import java.awt.*;  
import java.swing.*;  
A class that listens to a mouse click in a Square

```java
class Square {
  public static final int HEIGHT = 70; // height and width of square
  private int x, y; // Coordinates of square on board
  private boolean hasDisk = false; // = "square has pink disk"

  /** Constructor: a square at (x,y) */
  public Square(int x, int y) {
    this.x = x;
    this.y = y;
    setPreferredSize(new Dimension(WIDTH, HEIGHT));
  }

  /** Complement the "has pink disk" property */
  public void complementDisk() {
    hasDisk = ! hasDisk;
  }

  /** Paints a pink disk or removes it. */
  public void paint(Graphics g) {
    if ((x+y)%2 == 0) g.setColor(Color.green);
    else g.setColor(Color.red);
    g.fillRect(0, 0, WIDTH - 1, HEIGHT - 1);
    if (hasDisk) {
      g.setColor(Color.pink);
      g.fillRoundRect(7, 7, WIDTH - 14, HEIGHT - 14);
    }
    g.setColor(Color.black);
    g.fillRect(0, 0, WIDTH - 1, HEIGHT - 1);
    g.setColor(Color.green);
    g.drawOval(7, 7, WIDTH - 14, HEIGHT - 14);
  }

  /** An instance is a JPanel of size (WIDTH,HEIGHT). Green or red depending on whether the sum of constructor parameters is even or odd. */
  public void mousePressed(MouseEvent e) {
    int x = e.getX() - 3;
    int y = e.getY() - 3;
    hasDisk = ! hasDisk;
    repaint(); // Ask the system to repaint the square
  }
}
```

```java
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;

Class Square

A class that listens to a mouse click in a Square

```java
public class MouseDemo2 extends JFrame {
  public MouseDemo2() {
    super("Square Demo");
    setResizable(false);
    pack(); setVisible(true);
  }

  public void actionPerformed(ActionEvent e) {
    if (e.getSource() instanceof Square) {
      Square b = (Square) e.getSource();
      b.addKeyListener(new SquareKeyListener());
      b.addMouseMotionListener(new SquareMouseMotionListener());
      b.addMouseListener(new SquareMouseListener());
    }
  }
}
```