MusicShop Program (ComponentListener Example)

File: MusicShop.java

```java
/*
 * File: MusicShop.java
 * --------------------
 * This program handles the data management for a music
 * shop, showing which albums are carried and how many copies
 * are in stock.  The program handles dynamic resizing of
 * the program window.
 */

import acm.program.*;
import acm.util.*;
import java.awt.event.*;
import java.io.*;
import java.util.*;
import javax.swing.*;

public class MusicShop extends Program {

   // Set up initial display with interactors and canvas
    public void init() {
       label = new JLabel("Album Name");
       albumName = new JTextField(20);
       add(label, SOUTH);
       add(albumName, SOUTH);

       canvas = new MusicShopDisplay();
       add(canvas);

       loadInventory();

       addActionListeners();
       albumName.addActionListener(this);
    }

    // Read file to get inventory information on all albums
    private void loadInventory() {
      try {
         BufferedReader rd = new BufferedReader(
                                  new FileReader("music-data.txt"));
         while (true) {
            String line = rd.readLine();
            if (line == null) break;
            Album album = parseLine(line);
            inventory.put(album.getAlbumName(), album);
         }
         rd.close();
      } catch(IOException ex) {
         throw new ErrorException(ex);
      }
    }
```

```
   // Parse a single line from inventory file and returns an Album
   // object that contains the information from the line
   private Album parseLine(String line) {
      int albumNameStart = line.indexOf("[") + 1;
      int albumNameEnd = line.indexOf("]");
      String albumName = line.substring(albumNameStart, albumNameEnd);

      int bandNameStart = line.indexOf("[", albumNameEnd + 1) + 1;
         int bandNameEnd = line.indexOf("]", albumNameEnd + 1);
      String bandName = line.substring(bandNameStart, bandNameEnd);

      int numStockedStart = line.indexOf(" ", bandNameEnd + 1) + 1;
      int numStocked = Integer.parseInt(
                          line.substring(numStockedStart));

      return (new Album(albumName, bandName, numStocked));
   }

   // Update the display whenever the user enters a new album name
   public void actionPerformed(ActionEvent e) {
      if (e.getSource() == albumName) {
         canvas.displayInventory(inventory.get(albumName.getText()));
      }
   }

   /* Private instance variables */
   private JLabel label;
   private JTextField albumName;
   private MusicShopDisplay canvas;
   private Map<String,Album> inventory = new HashMap<String,Album>();

}
```

File: MusicShopDisplay.java

```
/*
 * File: MusicShopDisplay
 * ----------------------
 * This file handles the display of album inventory for the music shop.
 * It provides dynamic redrawing of window contents when display is
 * resized.
 */

import acm.graphics.*;
import java.awt.event.*;

public class MusicShopDisplay extends GCanvas
               implements ComponentListener {

   // Constructor
   public MusicShopDisplay() {
      addComponentListener(this);
      lastAlbum = null;
   }
```

```
    // Display the album name, band name, and number in stock
    // for a single album if it is in our inventory.  Otherwise,
    // just clear the display.
    public void displayInventory(Album album) {
        removeAll();
        lastAlbum = album;
        if (album != null) {
            int numStocked = album.getNumStocked();
            add(new GLabel("Album [" + album.getAlbumName() + "] by ["
                           + album.getBandName() + "]"),
                10, (getHeight() - BAR_HEIGHT) / 2 - SPACER);

            // Display squares in dicating how many inventory
            double nextX = SPACER;
            for(int i = 0; i < numStocked; i++) {
                double barLength = (getWidth() / (double)MAX_INVENTORY)
                                     - SPACER;
                GRect rect = new GRect(nextX,
                        (getHeight() - BAR_HEIGHT) / 2,
                        barLength, BAR_HEIGHT);
                rect.setFilled(true);
                add(rect);
                nextX += barLength + SPACER;
            }

            GLabel label = new GLabel(numStocked + " in stock");
            add(label, 10, (getHeight() + BAR_HEIGHT) / 2 +
                        SPACER + label.getAscent());
        }
    }


    // Whenever we need to update the display, continue to
    // display the last album shown
    private void update() {
        displayInventory(lastAlbum);
    }


    public void componentHidden(ComponentEvent e) { }
    public void componentMoved(ComponentEvent e) { }
    public void componentResized(ComponentEvent e) { update(); }
    public void componentShown(ComponentEvent e) { }


    /* constants */
    private static final double BAR_HEIGHT = 20;
    private static final double SPACER = 10;
    private static final int MAX_INVENTORY = 20;


    /* private instance variables */
    private Album lastAlbum;
}
```

File: Album.java

```java
/*
 * File: Album.java
 * ----------------
 * Keeps track of all the information for one album
 * in the music shop, including its name, the number
 * in stock, and the band that its by.
 */

public class Album {

    // Constructor
    public Album(String album, String band, int stock) {
        albumName = album;
        bandName = band;
        numStocked = stock;
    }

    public String getAlbumName() {
        return albumName;
    }

    public String getBandName() {
        return bandName;
    }

    public int getNumStocked() {
        return numStocked;
    }

    // Returns a string representation of an album, listing
    // the album name, the band name, and the number in stock
    public String toString() {
        return ("\"" + albumName +
                "\" by " + bandName +
                ": " + numStocked + " in stock");
    }

    /* private instance variables */
    private String albumName;
    private String bandName;
    private int numStocked;
}
```

File: music-data.txt

```
[Snakes and Arrows] [Rush] 10
[Synchronicity] [The Police] 12
[Piece of Mind] [Iron Maiden] 2
[Plans] [Death Cab For Cutie] 10
[So] [Peter Gabriel] 20
```