Slide 1

Administrivia

- Homework 5 on Web later today. Due dates next week.

Slide 2

Minute Essay From Last Lecture

- Here's a line of code that can throw two kinds of exceptions:

  ```
  FileInputStream fis = new FileInputStream("someFile");
  ```

  Write a few lines of code to catch these exceptions and print out a meaningful error message.

  ```
  try {
      FileInputStream fis = new FileInputStream("someFile");
  }
  catch (FileNotFoundException e) {
      System.out.println(e);
  }
  catch (SecurityException e) {
      System.out.println(e);
  }
  ```

Slide 3

Recap — Error Handling with Exceptions

- Not the only way to deal with errors, but a good one, and widely used in Java.
- Clearing up a point from last time . . .

  `try` block is followed by zero or more `catch` blocks and an optional `finally` block.

  If no exceptions are thrown in `try` block, we skip the `catch` blocks, do the `finally` block, and go on.

  If an exception that matches a `catch` is thrown, we skip immediately to that block, execute it, do the `finally` block, and go on.

  If an exception that doesn't match any `catch` is thrown, we do the `finally` block and exit to caller.

  `finally` is always done (except when `System.exit` is called).

Slide 4

Java GUI Libraries

- Java being an evolving language, it has two groups of GUI-related classes:
  - Abstract Window Toolkit (AWT) — older, “look and feel” consistent with platform's windowing system.
  - Swing — newer, more extensive, look and feel more aimed at being consistent across platforms. Makes use of AWT components.
- Many, many classes to build GUIs:
  - GUI elements —- buttons, labels, text boxes, menus, etc., etc., etc., etc.
  - “Containers” to group elements and arrange them for display.
  - “Listeners” and “events” to allow program to respond to user input.
- Programs are “event-based” or “event-driven”, can seem a little different from traditional text-in/text-out programs.

Slide 5

Some GUI Classes

- `Component` — base class.
- `Container` – component that can contain other components
- `JFrame` — window with titlebar, etc.; usually the “main” window for an application.
- `JDialog` — popup dialog box.
- `JPanel` – very simple container, useful for grouping things, providing custom graphics.
- `JMenuBar`.
- Etc., etc., etc., etc. — far more than we can cover in this course! Read the API. Some classes have links to online tutorials too.

Slide 6

Using the GUI Classes — Appearance

- When using predefined components, key issue is how they're grouped into container and how things are laid out within each container.
- Preferred method is to use a layout manager — places elements in some reasonable way, does something reasonable if container is resized.
  - `FlowLayout` — sort of like text flows in a paragraph.
  - `GridLayout` — two-dimensional grid, all elements the same size.
  - `BorderLayout` — five regions (north, south, etc.).
  - Others exist — `GridBagLayout` is most complex/flexible.

  Some of them expand components to fits, others lay them out at their minimum size.
- Often makes sense to group elements hierarchically — `JPanel` is useful for that.

Slide 7

Using the GUI Classes — Behavior

- Under the hood, Java programs using GUI classes are multithreaded (a thread to deal with user input, a thread to display output, etc.)
- The thread that deals with user input translates each user action (keyboard or mouse input) into an “event” and then calls method(s) in “event listener” objects.
- So, to tell the runtime system what should happen when, e.g., a `JButton` is clicked, call button's `addActionListener` with an object `listenerj` that implements `ActionListener` interface. Now when the button is clicked, `listener`'s `actionPerformed` method is called.
- Several approaches to defining listener objects. One is to have “main” class implement required interface. Another is to use anonymous inner classes.

Slide 8

Minute Essay

- The game framework will allow you to add panels to any or all four sides of your game. You can display info (text is easiest) or include GUI components for additional user input (e.g., click a button to speed up the player). You can also add to the menu bar.

  How might this be helpful for your game?