

CS1007 lecture #18 notes

tue 2 apr 2002

- news
- interfaces
- events
- listeners
- quick intro to GUIs
- reading: ch 5.4-5.6, 6.6

cs1007-spring2002-sklar-lect18

1

news.

- midterm #2 TUE APRIL 9
- hw#5 will be posted this week and will be due TUE APRIL 16
- hw#3 tournament results:
 - jmc2010
 - mbc2002
 - jpg2001
 - details will be posted on the web page

cs1007-spring2002-sklar-lect18

2

interfaces (1).

- an *interface* is a group of *abstract methods* that are defined by all classes that implement the interface
- an *abstract method* is one that does not have an implementation, i.e., there is no body of code for the method
- for example:
 - Speaker.java
 - Philosopher.java
 - Dog.java
 - Talking.java

cs1007-spring2002-sklar-lect18

3

interfaces (2).

- *polymorphism* means “having many forms”
 - lets us use different implementations of a single interface
 - *binding* happens when a particular implementation is locked to an interface
 - this can happen at compile time or at run time
 - an example of a run-time or *dynamic binding* is:
`((Philosopher)current).pontificate();`
from the earlier example

cs1007-spring2002-sklar-lect18

4

events (1).

- an *event* represents some action on the part of the user
- user-generated events are entered either through the *mouse* or the *keyboard*
- examples:
 - mouse pressed
 - mouse released
 - mouse clicked
 - mouse entered
 - mouse exited
 - mouse moved
 - mouse dragged

csl007-spring2002-sklar-lect18

5

listeners (1).

- a *listener* is a part of a program that captures these events for processing in the program
- frequently, a *listener interface* is created
- for example, `java.awt.event.MouseListener`:
 - `void mousePressed( MouseEvent evt );`
 - `void mouseReleased( MouseEvent evt );`
 - `void mouseClicked( MouseEvent evt );`
 - `void mouseEntered( MouseEvent evt );`
 - `void mouseExited( MouseEvent evt );`
- what is a `MouseEvent`?
 - `Point` `getPoint();`
 - `int` `getX();`
 - `int` `getY();`
 - `int` `getClickCount();`

csl007-spring2002-sklar-lect18

6

listeners (2).

- `MouseListener` examples:
 - `Dots.java`
 - `Dots2.java`
 - `Dots3.java`

csl007-spring2002-sklar-lect18

7

listeners (3).

- another example, `java.awt.event.KeyListener`:
 - `void keyPressed( KeyEvent evt );`
 - `void keyReleased( KeyEvent evt );`
 - `void keyTyped( KeyEvent evt );`
- what is a `KeyEvent`?
 - `char` `getKeyCode();`

csl007-spring2002-sklar-lect18

8

listeners (4).

- KeyListener examples:
 - Dots4.java
 - Dots5.java

cs1007-spring2002-sklar-lect18

9

GUIs (1).

- Graphical User Interface
- topics:
 - components
 - containers
 - layout managers
 - events
 - listeners

cs1007-spring2002-sklar-lect18

10

GUIs (2).

- *components*
- a component is a building block of any GUI
- here are some examples:
 - Label
 - TextField, TextArea
 - PushButton
 - CheckBox
 - RadioButton
 - ComboBox
 - List
 - PulldownMenu
 - ... and many more!!

cs1007-spring2002-sklar-lect18

11

GUIs (3).

- *containers*
- a container is a special component that can hold other components
- here are some examples:
 - Applet
 - Frame
 - Panel

cs1007-spring2002-sklar-lect18

12

GUIs (4).

- *layout managers*
- a layout manager describes where the components are laid out within a given container
- you need to “set” the layout manager for each container
- you can “nest” containers (and their layout managers)
- **BorderLayout** — simplest layout manager
- looks like this:

north		
west	center	east
south		