CS1007 lecture #17 notes

thu 7 nov 2002

- news
- drawing
- reading: ch 8.5-8.9, 9.1-9.4

applets (1).

- Java programs can run as *applications* or *applets*
- *application:*
  - executed using the *java* command
  - server and client can be the same machine or different machines
  - client invokes JVM which interprets classes and runs them
- *applet:*
  - must be executed using a browser, like Netscape, or the *appletviewer* command
  - server sends applet to the client, in the form of class files; applet invokes JVM which interprets classes and runs them on the client
  - there are two parts:
    - an HTML file used to invoke the applet
    - Java class file(s) that contain the applet code

applets (2).

- `java.awt` package
  - *Abstract Windowing Toolkit* (AWT)
  - classes that support graphical user interfaces (GUI)
  - includes `java.awt.Component` method:
    - `public void paint()`
- `java.applet.Applet` class
  - `public void init()`
  - `public void start()`
  - `public void stop()`

graphics.

- `java.awt.Graphics` class
- X-windows coordinate system
- drawing primitives:
  - lines
  - Strings
  - rectangles
  - ovals
  - arcs
- color

graphics (2).

- simple methods from the `java.awt.Graphics` class
  - `void drawLine( int x1, int y1, int x2, int y2 );`
    - draws a line connecting (x1,y1) and (x2,y2);
  - `void drawString( String str, int x, int y );`
    - draws the text in “str”, with its lower left corner at (x,y)

graphics (3).

```
import java.awt.*;
import java.applet.Applet;

public class ex16a extends Applet {

  public void paint ( Graphics g ) {
    g.drawString( "hello world!",10,10 );
    g.drawLine( 0,400, 400,0 );
  } // end of paint()

} // end of class ex16a()
```

graphics (4).

- bounding rectangles
  - coordinates of origin (upper left corner)
  - extent (width and height)
- arcs
  - measured in degrees
  - starting from 0° (along positive X-axis, like hw#3)
  - extent (total angle of arc)

graphics (5).

- more methods from the `java.awt.Graphics` class
  - `void drawRect( int x, int y, int width, int height );`
    - draws a rectangle with its upper left corner at (x,y), extending the specified "width" and "height"
  - `void drawOval( int x, int y, int width, int height );`
    - draws an oval circumscribed in the bounding rectangle with its upper left corner at (x,y), extending the specified "width" and "height"
  - `void drawArc( int x, int y, int width, int height, int startAngle, int arcAngle );`
    - draws an arc whose oval is circumscribed in the bounding rectangle with its upper left corner at (x,y), extending the specified "width" and "height", where the arc starts at the "startAngle", measured in degrees (where 0°) is horizontal along the positive x-axis), extending for "arcAngle" degrees

graphics (6).

```
import java.awt.*;
import java.applet.Applet;

public class ex16b extends Applet {

  public void paint ( Graphics g ) {
    g.drawRect( 10,300,25,25 );
    g.drawOval( 10,250,25,25 );
    g.drawArc( 10,200,25,25,45,90 );
  } // end of paint()

} // end of class ex16b()
```

graphics (7).

- `java.awt.Color` class
- color is defined using the “RGB” methodology
- “Red”, “Green”, “Blue”
- each is an integer between 0 and 255, where 0 means no color and 255 means maximum color
- so white is: red=255 green=255 blue=255 or the ordered triple (255,255,255)
  - and black is: red=0 green=0 blue=0
  - and red is: red=255 green=0 blue=0
  - and green is: red=0 green=255 blue=0
  - and blue is: red=0 green=0 blue=255
- make up your own colors...

graphics (8).

- even more methods from the `java.awt.Graphics` class
  - `void setColor( Color color );`
    - sets the foreground (pen) color to the specified color
  - `void fillRect( int x, int y, int width, int height );`
    - draws a filled rectangle with its upper left corner at (x,y), extending the specified "width" and "height"
  - `void fillOval( int x, int y, int width, int height );`
    - draws a filled oval circumscribed in the bounding rectangle with its upper left corner at (x,y), extending the specified "width" and "height"
  - `void fillArc( int x, int y, int width, int height, int startAngle, int arcAngle );`
    - draws a filled arc whose oval is circumscribed in the bounding rectangle with its upper left corner at (x,y), extending the specified "width" and "height", where the arc starts at the "startAngle", measured in degrees (where 0°) is horizontal along the positive x-axis), extending for "arcAngle" degrees

examples.

- `snowman.java`
- `bullseye.java`
- `snowman2.java`, `man.java` (with helper class)
- `snowman3.java` (animated)
- `snowman4.java` (another way of doing animation)