// Exercise 16.15 Solution: Ball.java

// Program bounces a ball around the applet

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class Ball extends JApplet implements Runnable

{

private Thread blueBall;

private boolean xUp, yUp, bouncing;

private int x, y, xDx, yDy;

private final int MAX\_X = 200, MAX\_Y = 200;

public void init()

{

// initialize values

xUp = false;

yUp = false;

xDx = 1;

yDy = 1;

bouncing = false;

// let Ball Applet be its own MouseListener

addMouseListener(

new MouseListener()

{

public void mousePressed( MouseEvent event )

{

createBall( event ); // delegate call to ball //starter

}

public void mouseExited( MouseEvent event ) {}

public void mouseClicked( MouseEvent event ) {}

public void mouseReleased( MouseEvent event ) {}

public void mouseEntered( MouseEvent event ) {}

} // end of MouseListener

);

setSize( MAX\_X, MAX\_Y ); // set size of Applet

} // end of init

// creates a ball and sets it in motion if no ball exists

private void createBall( MouseEvent event )

{

if ( blueBall == null ) {

x = event.getX();

y = event.getY();

blueBall = new Thread( this );

bouncing = true; // start ball's bouncing

blueBall.start();

}

} // end of createBall method

// called if applet is closed. by setting blueBall to null,

// threads will be ended.

public void stop()

{

blueBall = null;

}

// draws ball at current position

public void paint( Graphics g )

{

super.paint( g );

if ( bouncing ) {

g.setColor( Color.blue );

g.fillOval( x, y, 10, 10 );

}

} // end of paint method

// action to perform on execution, bounces ball

// perpetually until applet is closed.

public void run()

{

while ( true ) {

// sleep for a random interval

try {

blueBall.sleep( 20 );

}

// process InterruptedException during sleep

catch ( InterruptedException exception ) {

System.err.println( exception.toString() );

}

// determine new x position

if ( xUp == true )

x += xDx;

else

x -= xDx;

// determine new y position

if ( yUp == true )

y += yDy;

else

y -= yDy;

// randomize variables for creating next move

if ( y <= 0 ) {

yUp = true;

yDy = ( int ) ( Math.random() \* 5 + 2 );

}

else if ( y >= MAX\_Y - 10 ) {

yDy = ( int ) ( Math.random() \* 5 + 2 );

yUp = false;

}

if ( x <= 0 ) {

xUp = true;

xDx = ( int ) ( Math.random() \* 5 + 2 );

}

else if ( x >= MAX\_X - 10 ) {

xUp = false;

xDx = ( int ) ( Math.random() \* 5 + 2 );

}

repaint();

} // end while

} // end method run

} // end class Ball