NetBeans and Active Objects

2 – 4 pm Tuesday 8/19/2008 @JD2211

1

Agenda

- Review
  - Event, Listener, and Action
- NetBeans
  - Design GUI and add functionalities
- Active objects

2

NetBeans

- www.netbeans.org
- www.netbeans.org/features/index.html

3

Number Addition -- Tutorial

- Using NetBeans, do 8-14 lab assignment

4

Active Objects

- Passive objects
- Active objects

5

Concurrent Program

- A program in which there is more than one active object

6

Java & Ada

- Support the concurrent programs

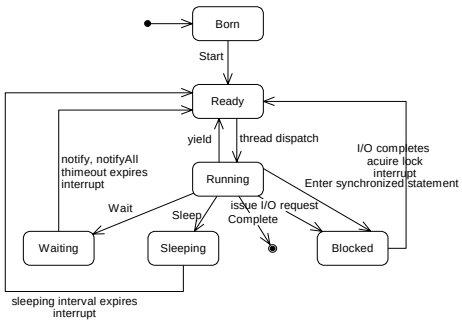
7

Thread

- java.lang.Object
  - java.lang.Thread
- public class Thread extends Object implements Runnable
- Thread a1 = new Thread();
- a1.start();

8

Thread Life-Cycle UML Statechart Diagram



9

```
class CounterThread extends Thread {
  public boolean stopped = false;
  int count = 0;

  public void run() {
    while (!stopped) {
      try {
        sleep(1000);
      } catch (InterruptedException e) {}
    System.out.println(count++);
    }
  }
}
```

10

```
public class MainClass {
  public static void main(String[] args) {
    CounterThread thread = new CounterThread();
    thread.start();
    try {
      Thread.sleep(10000);
    } catch (InterruptedException e) {}
    thread.stopped = true;
    System.out.println("exit");
  }
}
```

11

```
class CounterThread implements Runnable {
  public Thread thread = new Thread(this);
  public boolean stopped = false;

  int count = 0;

  public void run() {
    while (!stopped) {
      try {
        Thread.sleep(1000);
      } catch (InterruptedException e) {}
      System.out.println(count++);
    }
  }
}
```

12

```
public class RunnableTest {
  public static void main(String[] args) {
    CounterThread t1 = new CounterThread();
    t1.thread.start();
    try {
      Thread.sleep(10000);
    } catch (InterruptedException e) {}
    t1.stopped = true;
    System.out.println("exit");
  }
}
```

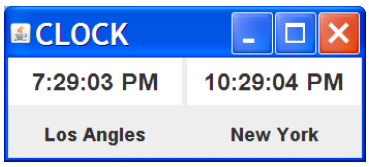
13

Dynamic Swing Components

- Class `Timer`
- `javax.swing.Timer`
- `java.util.Timer`

14

Exercise



15

Timer

- A Timer object can generate regularly time events of type ActionEvent
- The frequency with which these events are generated and who listens to the events are specified as parameters to the constructor

```
Timer t = new Timer (1000, l);

t.start();
```

16

DateFormat

- `java.text.DateFormat`
- getTimeInstance()
- setTimeZone()
- http://java.sun.com/j2se/1.4.2/docs/api/java/text/DateFormat.html

17

TimeZone

- `java.util.TimeZone`
- `getTimeZone(zone)`
  - `TimeZone.getTimeZone("America/Los_Angeles");`

18

Lab Assignment

- http://www.csun.edu/~twang/Java/Lab/8-19-Lab.pdf

19