Using Threads with Buttons in Java

Giuseppe Narzisi

V22.0102-003

Spring 2009
JBu1on and actionPerformed

• Desire:
  – When the button is pressed certain things need to be done which should update a display.

• Problem:
  – Display does not update until the actionPerform method has fully completed.

• Solution:
  – Java threads!
Thread.sleep()

- Use `Thread.sleep()` to pause in a repetitive process initiated by a button.
- To have the `pause()` operate properly, you have to start another thread. One way of doing this is with an anonymous inner class.

```java
Thread advanceThread = new Thread()
{
    public void run()
    {
        findStart();
    }
};
advanceThread.start();
```
Implementation details

• In the anonymous inner class, we write our own version of run() which overrides the run method of class Thread.
• In run() we call the method that contains the pause(), here findStart().
• We then call the start() method of Thread.
• The entire code given above should be placed in the scope of the actionPerformed method of the button class that implements ActionListener.
Remark

• To have the pause() operate properly the whole work of the action performed must be delegated to the new thread.

class ThreadListener implements ActionListener{
    public void findStart(){
        // do the work here
    }

    public void actionPerformed(ActionEvent e){
        Thread advanceThread = new Thread(){
            public void run(){
                findStart();
            }
        };
        advanceThread.start();
    }
}