

Implementação de threads

- Frequentemente, você precisa implementar seus próprios threads
- Isso é feito dentro do método run() do thread que especifica o comportamento do thread
- A class Thread implementa um thread que não faz nada (run() está vazio)
- Discutimos duas formas de especificar o que fazer no método run()

Herdando da classe Thread

- Basta herdar de Thread e fazer override de run()
- Exemplo:

```
public class SimpleThread extends Thread {  
    public SimpleThread(String str) {  
        super(str);  
    }  
    public void run() {  
        for (int i = 0; i < 10; i++) {  
            System.out.println(i + " " + getName());  
            try {  
                sleep((long)(Math.random() * 1000));  
            } catch (InterruptedException e) {}  
        }  
        System.out.println("DONE! " + getName());  
    }  
}
```

- O construtor é interessante pois dá um nome ao thread
 - O nome é obtido com getName()
- Vamos agora rodar um programa com 2 threads (na realidade, 3 threads!)

```
public class TwoThreadsTest {  
    public static void main (String[] args) {  
        new SimpleThread("Jamaica").start();  
        new SimpleThread("Fiji").start();  
    }  
}
```

- start() é usado para iniciar o thread
 - start() avaba chamando run() do thread
- Execute o programa; a saída será algo como isso:

```
0 Jamaica  
0 Fiji  
1 Fiji  
1 Jamaica  
2 Jamaica  
2 Fiji  
3 Fiji  
3 Jamaica
```

```
4 Jamaica
4 Fiji
5 Jamaica
5 Fiji
6 Fiji
6 Jamaica
7 Jamaica
7 Fiji
8 Fiji
9 Fiji
8 Jamaica
DONE! Fiji
9 Jamaica
DONE! Jamaica
```

- A saída de cada thread se mistura com a outra já que os programas são concorrentes
 - Se houver uma única CPU, os threads não rodam ao mesmíssimo tempo
 - Falaremos de escalonamento adiante
- Quando o loop termina, o thread morre
- Quando os dois threads morrem (o original já está morto), o programa termina

Implementando um thread através da interface Runnable

- O applet acima exibe a hora atual e a atualiza a cada segundo
- O clock continua sendo atualizado mesmo quando você usa o browser de várias formas
 - Porque o relógio roda no seu próprio thread
- Em vez de herdar de Thread, Clock implementa a interface Runnable e implementa o método run
- Clock cria um thread dentro de si como alvo para o Thread
- Quando é criado assim, um thread obtém seu método run do objeto alvo
 - Veja o código em destaque abaixo

```
import java.awt.*;
import java.util.*;
import java.applet.*;
import java.text.*;

public class Clock extends java.applet.Applet implements Runnable {
    private volatile Thread clockThread = null;
    DateFormat formatter;        // Formats the date displayed
    String lastdate;              // String to hold date displayed
    Date currentDate;            // Used to get date to display
    Color numberColor;           // Color of numbers
    Font clockFaceFont;
    Locale locale;

    public void init() {
        setBackground(Color.white);
        numberColor = Color.red;
```

```

        locale = Locale.getDefault();
        formatter =
            DateFormat.getDateInstance(DateFormat.FULL,
            DateFormat.MEDIUM, locale);
        currentDate = new Date();
        lastdate = formatter.format(currentDate);
        clockFaceFont = new Font("Sans-Serif",
            Font.PLAIN, 14);

        resize(275,25);
    }

    public void start() {
        if (clockThread == null) {
            clockThread = new Thread(this, "Clock");
            clockThread.start();
        }
    }

    public void run() {
        Thread myThread = Thread.currentThread();
        while (clockThread == myThread) {
            repaint();
            try {
                Thread.sleep(1000);
            } catch (InterruptedException e){ }
        }
    }

    public void paint(Graphics g) {
        String today;
        currentDate = new Date();
        formatter =
            DateFormat.getDateInstance(DateFormat.FULL,
            DateFormat.MEDIUM, locale);
        today = formatter.format(currentDate);
        g.setFont(clockFaceFont);

        // Erase and redraw
        g.setColor(getBackground());
        g.drawString(lastdate, 0, 12);

        g.setColor(numberColor);
        g.drawString(today, 0, 12);
        lastdate = today;
        currentDate=null;

    }

    public void stop() {
        clockThread = null;
    }
}

```

- O applet executa o método run() até que o browser o peça para parar
 - Alguns browsers criam um thread separado para um applet de qualquer forma mas não é bom depender disso
- Falaremos mais sobre este programa ([Clock.java](#)) adiante

Qual método usar para implementar um thread?

- Se você quiser usar herança para outra coisa, não herde de Thread
- De forma geral, é comum implementar a interface Runnable

programa