

## PMR2300 - Computação para Automação

### Exercício Programa 3 - 2014

#### Interface gráfica para o sistema de simulação de filas

Prof. Dr. Fabio Gagliardi Cozman  
Prof. Dr. Newton Maruyama

## 1 Introdução

Neste exercício programa será desenvolvido uma interface gráfica para o sistema de simulação de filas desenvolvido no EP2.

## 2 Para você fazer

Requisitos da interface:

- Utilização da biblioteca AWT.
- No estado inicial deve existir como dar um *Exit* do programa.
- Deve existir como disparar a simulação.
- O usuário pode escolher entre as seguintes condições de simulação:
  1. Múltiplos Servidores, Múltiplas Filas, Fase Única (*Multiserver, single-line, single-phase*): Simulação com cinco caixas, cinco filas, fase única.
  2. Múltiplos Servidores, Fila Única, Fase Única (*Multiserver, single line, fase única*): Simulação com cinco caixas e fila única.
- Os seguintes parâmetros devem ser escolhidos através da interface:
  1. taxa de chegada de clientes  $\lambda$ ,
  2. taxa de atendimento  $\mu$  e
  3. tempo total de simulação.Sugere-se a utilização da classe `TextField`.
- Ao finalizar a simulação o programa gera um relatório contendo:
  1. Estatísticas: Número médio de clientes na fila, Tempo médio que um cliente permanece esperando na fila e seu respectivo desvio padrão, Maior tempo de permanência de um cliente na fila.

2. Gráfico mostrando a evolução no tempo do número de clientes em cada fila.

Sugere-se a utilização da classe `TextArea`.

- Ao final de cada simulação o sistema retorna ao estado inicial.

### 3 Como criar o gráfico

Existem várias bibliotecas que podem ser utilizadas para a criação de gráficos, porém aqui recomenda-se a utilização da biblioteca `JFreeChart` <sup>1</sup>

São necessários dois pacotes:

1. `JFreeChart`: <http://sourceforge.net/projects/jfreechart/files/1.%20JFreeChart/1.0.17/>
2. `JCommon`: <http://sourceforge.net/projects/jfreechart/files/3.%20JCommon/1.0.22/>

A variável de sistema `classpath` deve apontar para os dois arquivos `*.jar`. Por exemplo,

```
classpath=VariosDiretorios\jcommon-1.0.22\jcommon-1.0.22.jar;  
VariosDiretorios\jfreechart-1.0.17\jfreechart-1.0.17-demo.jar
```

A listagem 1 apresenta o programa `TesteXYStepChart.java` que ilustra como utilizar a biblioteca `JFreeChart` para a geração do gráfico contendo a evolução do número de pessoas na fila ao longo da simulação.

Listing 1: Programa `TesteXYStepChart.java`.

---

```
import java.awt.BasicStroke;  
import java.awt.Color;  
  
import org.jfree.chart.ChartFactory;  
import org.jfree.chart.ChartFrame;  
import org.jfree.chart.JFreeChart;  
import org.jfree.chart.plot.PlotOrientation;  
import org.jfree.chart.plot.XYPlot;  
import org.jfree.data.xy.XYDataset;  
import org.jfree.data.xy.XYSeries;  
import org.jfree.data.xy.XYSeriesCollection;  
import org.jfree.date.DateUtilities;  
import org.jfree.ui.RefineryUtilities;  
import org.jfree.chart.axis.CategoryAxis;  
import org.jfree.chart.axis.NumberAxis;  
  
public class TesteXYStepChart {
```

---

<sup>1</sup><http://www.jfree.org/jfreechart/>

```

private ChartFrame frame = null;

private void displayChart() {
    if (this.frame == null) {
        final String title = "Filas";
        final String xAxisLabel = "Tempo_de_Simulacao";
        final String yAxisLabel = "Numero_de_Pessoas_na_Fila";

        final XYDataset data = createStepXYDataset();

        final JFreeChart chart = ChartFactory.createXYStepChart(
            title,
            xAxisLabel, yAxisLabel,
            data,
            PlotOrientation.VERTICAL,
            true, // legend
            true, // tooltips
            false // urls
        );

        chart.setBackgroundPaint(new Color(216, 216, 216));

        final XYPlot plot = chart.getXYPlot();
        chart.getXYPlot().setDomainAxis(new NumberAxis());

        this.frame = new ChartFrame("Evolucao_da_Fila_no_Tempo", chart);
        this.frame.pack();
        RefineryUtilities.positionFrameRandomly(this.frame);
        this.frame.setVisible(true);
    }
    else {
        this.frame.setVisible(true);
        this.frame.requestFocus();
    }
}

public static XYDataset createStepXYDataset() {
    final XYSeries s1 = new XYSeries("Fila_1");
    s1.add(0.0,0);
    s1.add(10.0,1);
    s1.add(12.0,4);
    s1.add(13.0,4);

    final XYSeries s2 = new XYSeries("Fila_2");
    s2.add(0.0,0);
}

```

```

s2.add(2.0,1);
s2.add(3.0,4);
s2.add(12.0,3);
s2.add(13.0,3);

final XYSeriesCollection dataset = new XYSeriesCollection();
dataset.addSeries(s1);
dataset.addSeries(s2);
return dataset;
}

public static void main(final String[] args) {

final TesteXYStepChart demo = new TesteXYStepChart();
demo.displayChart();

}
}

```

Este programa fornece como saída o gráfico ilustrado na Figura 1.

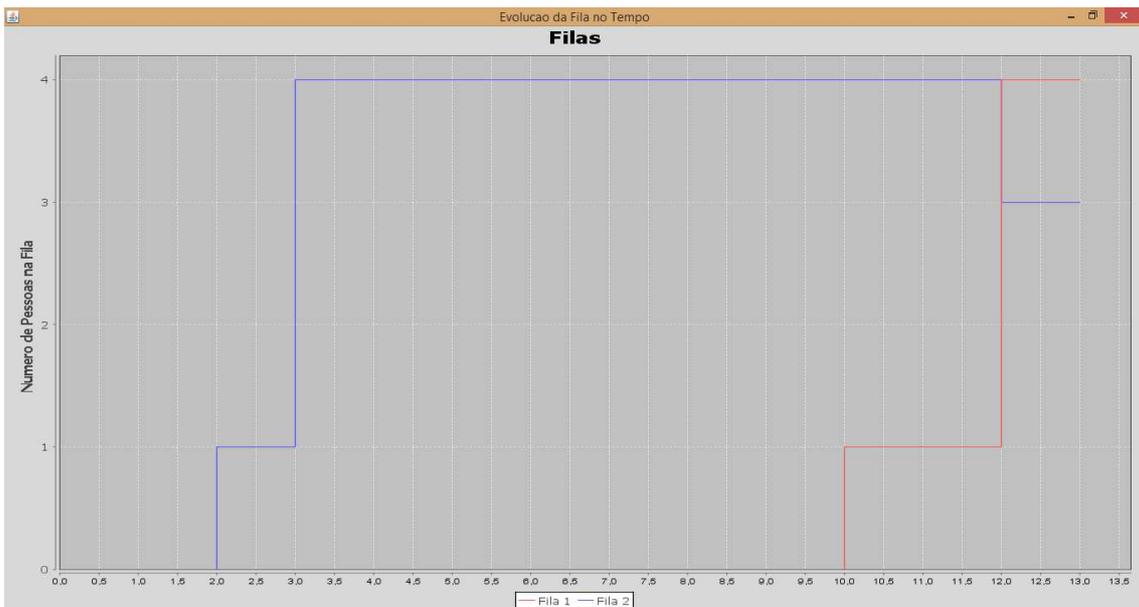


Figura 1: Resultado do programa TesteXYStepChart.

## 4 Coisas importantes

- **DATA FINAL DE ENTREGA: Quarta-Feira 25/06/2014 (No horário de monitoria: 12:00h-13:00h).**

- **O exercício deve ser feito individualmente.**
- **Entregue ao monitor da disciplina**
  - **Documentação impressa descrevendo o seu projeto em uma página A4,**
  - **Código fonte, programa executável (bytecode) e documentação em um arquivo compactado.**