Afficher un graphe jfreechart en 5 minutes

Par Thierry Leriche-Dessirier

Date de publication : 13 janvier 2012

Dernière mise à jour : 25 janvier 2012

Ce mini article montre (par l'exemple) comment ajouter un graphique dans une fenêtre Swing, à l'aide de jfreechart, en quelques minutes. **Commentez**

I - Introduction....3
I-A - À propos....3
I-B - Avant de commencer....3
I-C - Mise-à-jour....3
II - Découverte du projet d'exemple....3
II-A - Télécharger, installer et importer le projet d'exemple....3
II-B - Ce que fait déjà le projet d'exemple....3
III - Action....4
III-A - Préparer le terrain....4
III-B - Un passage rapide par Maven et Eclipse....5
III-C - Camembert....8
III-D - Barres....9
IV - Conclusions....11
V - Remerciements....12
VI - Annexes....12
VI-A - Liens....12
VI-B - Les fichiers importants en entier....12

I - Introduction

Dans ce document, nous allons voir comment créer des graphique à l'aide de la librairie **jfreechart** et les afficher dans une fenêtre Swing.

Jfreechart est une librairie gratuite, écrite en Java, qui simplifie la génération et l'affichage de graphe de qualité professionnelle.

I-A - À propos

Découvrir une technologie n'est pas chose facile. En aborder plusieurs d'un coup l'est encore moins. Partant de ce constat, cet article a été écrit pour aller à l'essentiel. Les points importants sont présentés dans le corps de l'article et les éléments secondaires sont expliqués en annexes.

I-B - Avant de commencer

Pour écrire ce tutoriel, j'ai utilisé les éléments suivants :

- Java JDK 1.6.0_24-b07 ;
- Eclipse Indigo 3.7 JEE 64b ;
- Maven 3.0.3.

Ce tutoriel est la suite logique de l'article "**Afficher un tableau avec un Table Model en 5 minutes**" dont je recommande la lecture avant d'aller plus loin.

I-C - Mise-à-jour

25/01/2012 : Ajout d'un lien vers la série d'articles "en 5 minutes".

II - Découverte du projet d'exemple

II-A - Télécharger, installer et importer le projet d'exemple

Pour commencer, je vous propose de télécharger **le fichier Zip "notes2.zip"**, contenant un projet Java-Maven d'exemple.

Compilez le projet d'exemple et importez-le dans Eclipse (comme expliqué dans le tutoriel **"Importer un projet Maven dans Eclipse en 5 minutes"**) ou dans l'IDE de votre choix.

Pour suivre ce tutoriel, vous pouvez vous contenter de lire les codes proposés ci-dessous (codes complets en annexes) et de faire confiance aux captures d'écran.

II-B - Ce que fait déjà le projet d'exemple

Le projet d'exemple est composé d'un ensemble d'éléments déjà présentés dans le tutoriel "**Afficher un tableau avec un Table Model en 5 minutes**". Il permet d'affichier les notes des élèves au dernier examen. Les données sont chargées à l'aide d'un "table model".

Le projet et son tableau

III - Action

III-A - Préparer le terrain

Durée estimée : 1 minute.

Avant d'afficher un graphique dans notre IHM, nous allons ajouter un bouton. Un clic sur ce bouton devra ouvrir une fenêtre avec un graphe jfreechart représentant la proportion de fille/garçon parmi les élèves.

Pour commencer, nous allons créer une "action" simple sous la forme d'une classe interne.

Action

```
public class NotesJFrame extends JFrame {

 ...

 private class ShowRatioHommeFemmeAction extends AbstractAction {

  private ShowRatioHommeFemmeAction() {
   super("Ratio h/f");
  }

  @Override
  public void actionPerformed(ActionEvent arg0) {
   System.out.println("coucou");
  }
 }
```

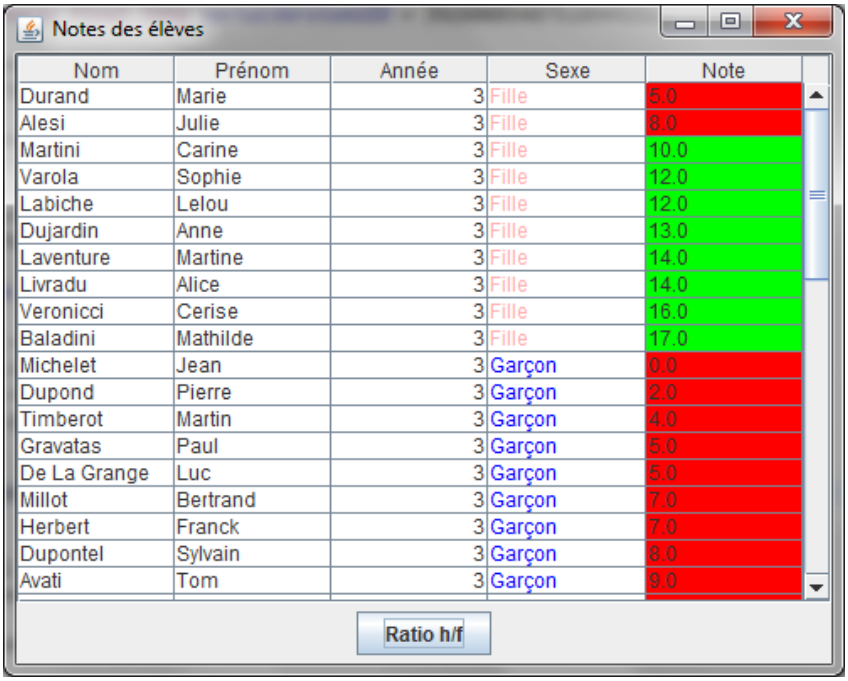
On ajoute ensuite un bouton dans la fenêtre principale. Un clic sur le bouton lancera l'action qu'on vient de créer.

Bouton lié à l'action

```
public class NotesJFrame extends JFrame {

 ...

 public NotesJFrame() {
```

Bouton lié à l'action

```
...

JPanel boutons = new JPanel();
boutons.add(new JButton(new ShowRatioHommeFemmeAction()));

getContentPane().add(boutons, SOUTH);
```



Un bouton en dessous du tableau

Quand on clique sur le bouton, ça écrit "coucou" dans la console.

Maintenant, à la place d'écrire dans la console, nous allons ouvrir une fenêtre.

Fenêtre

```
public class NotesJFrame extends JFrame {

 private JDialog ratioHommeFemmeJdialog;
 ...

 private class ShowRatioHommeFemmeAction extends AbstractAction {
  ...
  public void actionPerformed(ActionEvent e) {

   ratioHommeFemmeJdialog = new JDialog();
   ratioHommeFemmeJdialog.setTitle("Ratio H/F");

   ratioHommeFemmeJdialog.pack();
   ratioHommeFemmeJdialog.setVisible(true);
```

Pour le moment, la fenêtre qui s'ouvre est toute petite. Elle prendra sa taille définitive lorsqu'on ajoutera le graphique jfreechart.

III-B - Un passage rapide par Maven et Eclipse

Durée estimée : 1 minute.

Avant de pouvoir créer un graphique jfreechart, encore faut-il ajouter une dépendance vers cette librairie dans le fichier pom.xml.

pom.xml (code simplifié)

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
 <modelVersion>4.0.0</modelVersion>

 ...

 <dependencies>
  <!-- Junit -->
  <dependency>
   <groupId>junit</groupId>
   <artifactId>junit</artifactId>
   <scope>test</scope>
   <version>4.8.2</version>
  </dependency>

  <!-- jfreechart -->
  <dependency>
   <groupId>org.jfree</groupId>
   <artifactId>jfreechart</artifactId>
   <version>1.0.14</version>
  </dependency>
 </dependencies>

 ...
```

Ouvrez ensuite un terminal (commande *"cmd"* sous Windows) et placez-vous à la racine du projet (*c:\dvp\notes* pour cet exemple). Puis lancez une installation Maven-Eclipse pour prendre en compte l'ajout de la dépendance vers jfreechart.

Installation Maven

```
mvn clean install eclipse:eclipse
```

En fonction des éléments déjà présents sur le disque, le résultat Maven devrait ressembler à la trace suivante.

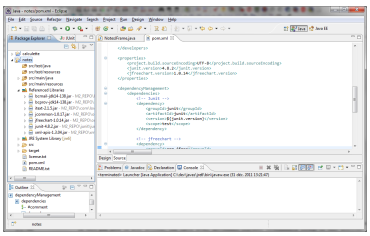
Installation Maven

```
C:\dvp\notes>mvn clean install eclipse:eclipse

[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building Notes 1.0-SNAPSHOT
[INFO] ------------------------------------------------------------------------
[INFO]
[INFO] --- maven-clean-plugin:2.4.1:clean (default-clean) @ notes ---
[INFO] Deleting C:\dvp\notes\target
[INFO]
[INFO] --- maven-resources-plugin:2.4.3:resources (default-resources) @ notes ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:2.3.2:compile (default-compile) @ notes ---
[INFO] Compiling 9 source files to C:\dvp\notes\target\classes
[INFO]
[INFO] --- maven-resources-plugin:2.4.3:testResources (default-testResources) @ notes ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:2.3.2:testCompile (default-testCompile) @ notes ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-surefire-plugin:2.7.2:test (default-test) @ notes ---
[INFO] Surefire report directory: C:\dvp\notes\target\surefire-reports
```

Installation Maven

```
-------------------------------------------------------
 T E S T S
-------------------------------------------------------
There are no tests to run.

Results :

Tests run: 0, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- maven-jar-plugin:2.3.1:jar (default-jar) @ notes ---
[INFO] Building jar: C:\dvp\notes\target\notes-1.0-SNAPSHOT.jar
[INFO]
[INFO] --- maven-install-plugin:2.3.1:install (default-install) @ notes ---
[INFO] Installing C:\dvp\notes\target\notes-1.0-SNAPSHOT.jar to C:\dev\mavens\repository\com\thi
\notes\1.0-SNAPSHOT\notes-1.0-
[INFO] Installing C:\dvp\notes\pom.xml to C:\dev\mavens\repository\com\thi\notes\1.0-SNAPSHOT
\notes-1.0-SNAPSHOT.pom
[INFO]
[INFO]  maven-eclipse-plugin:2.8:eclipse (default-cli) @ notes
[INFO]
[INFO]  maven-eclipse-plugin:2.8:eclipse (default-cli) @ notes
[INFO]
[INFO] --- maven-eclipse-plugin:2.8:eclipse (default-cli) @ notes ---
[INFO] Using Eclipse Workspace: null
[INFO] Adding default classpath container: org.eclipse.jdt.launching.JRE_CONTAINER
[INFO] Not writing settings - defaults suffice
[INFO] File C:\dvp\notes\.project already exists.
       Additional settings will be preserved, run mvn eclipse:clean if you want old settings to
 be removed.
[INFO] Wrote Eclipse project for "notes" to C:\dvp\notes.
[INFO]
       Sources for some artifacts are not available.
       Please run the same goal with the -DdownloadSources=true parameter in order to check
 remote repositories for sources.
       List of artifacts without a source archive:
         o xml-apis:xml-apis:1.3.04

       Javadoc for some artifacts is not available.
       Please run the same goal with the -DdownloadJavadocs=true parameter in order to check
 remote repositories for javadoc.
       List of artifacts without a javadoc archive:
         o xml-apis:xml-apis:1.3.04

[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 10.278s
[INFO] Finished at: Sat Dec 31 13:31:23 CET 2011
[INFO] Final Memory: 13M/61M
[INFO] ------------------------------------------------------------------------
C:\dvp\notes>
```

Il faut aussi rafraîchir le workspace d'Eclipse (touche F5). On constate que plusieurs librairies sont apparues.



De nouvelles dépendances dans Eclipse

On constate également qu'il n'y a pas que junit et jfreechart qui sont apparues dans Eclipse. Les autres librairies ont simplement été tirées par Maven car jfreechart en a besoin.

III-C - Camembert

Durée estimée : 1 minute.

Maintenant que le projet possède une dépendance vers la librairie jfreechart, on peut créer un premier graphique. Comme on veut représenter la proportion fille/garçon, on va naturellement utiliser un graphe en camembert. Dans un premier temps, il faut calculer les valeurs à afficher.

Le calcul du nombre de fille/garçon n'est pas important dans le cadre de ce tutoriel (même un peu hors sujet) mais j'en ai besoin pour avoir quelque chose à afficher.

Préparation des données

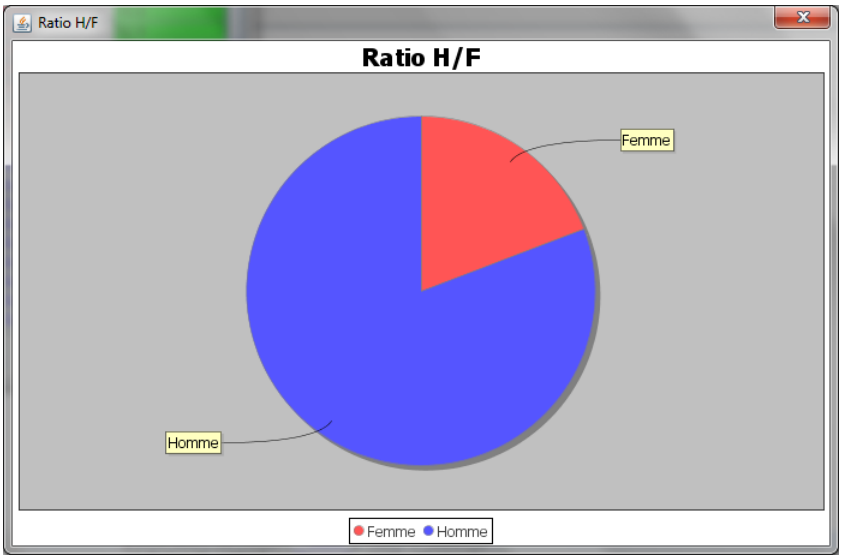
```
public class NotesJFrame extends JFrame {
 ...

 private class ShowRatioHommeFemmeAction extends AbstractAction {
  ...

  public void actionPerformed(ActionEvent e) {
   ratioHommeFemmeJdialog = new JDialog();
   ratioHommeFemmeJdialog.setTitle("Ratio H/F");

   // Calcul du ratio
   final List<NoteEleve> notes = modele.getNotes();

   int nombreFemmes = 0;
   int nombreHommes = 0;

   for (NoteEleve noteEleve : notes) {
    if (noteEleve.getEleve().getSexe() == Sexe.FEMME) {
     nombreFemmes++;
    } else {
     nombreHommes++;
    }
   }

   ...
```

On peut enfin générer et afficher le graphique.

Camembert

```
   ...
   @Override
   public void actionPerformed(ActionEvent arg0) {
    ...

    final DefaultPieDataset pieDataset = new DefaultPieDataset();

    pieDataset.setValue("Femme", nombreFemmes);
    pieDataset.setValue("Homme", nombreHommes);

    final JFreeChart pieChart = ChartFactory.createPieChart("Ratio H/F",
  pieDataset, true, false, false);
    final ChartPanel cPanel = new ChartPanel(pieChart);

    ratioHommeFemmeJdialog.getContentPane().add(cPanel, CENTER);

    ratioHommeFemmeJdialog.pack();
    ratioHommeFemmeJdialog.setVisible(true);
   }
```

Et voilà...



Camembert

III-D - Barres

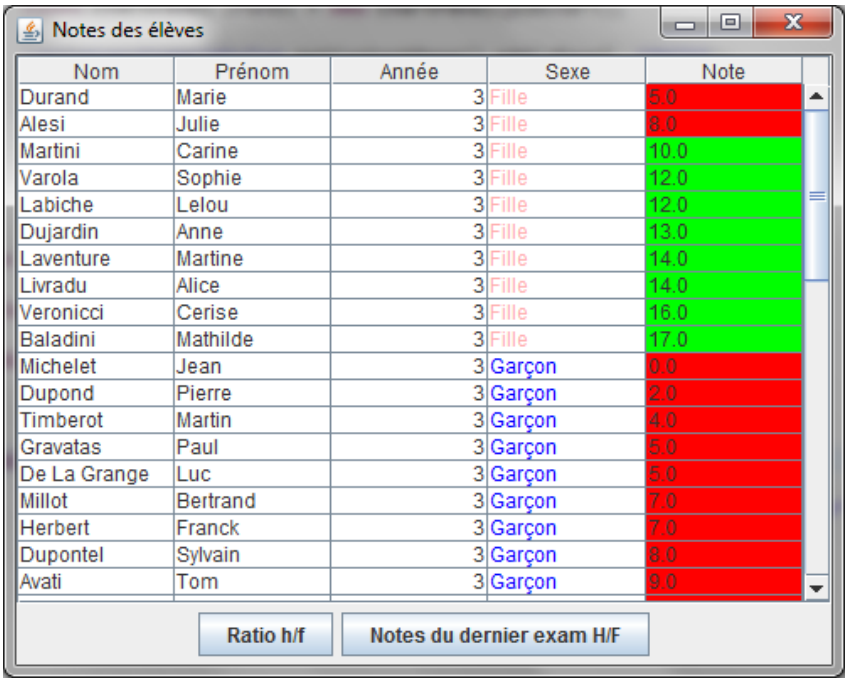
Durée estimée : 2 minutes.

On va maintenant afficher un second graphique, en barres, représentant les notes obtenues par les élèves, en fonction des sexes.

Comme pour le premier graphique, on crée d'abord une action et un bouton.

Action et bouton

```
public class NotesJFrame extends JFrame {
 ...
 private JDialog ratioHommeFemmeJdialog;
 private JDialog notesDernierExamHommeFemmeJdialog;

 public NotesJFrame() {
  ...

  JPanel boutons = new JPanel();
  boutons.add(new JButton(new ShowRatioHommeFemmeAction()));
  boutons.add(new JButton(new ShowNotesHommeFemmeAction()));

  getContentPane().add(boutons, SOUTH);
  ...

 private class ShowNotesHommeFemmeAction extends AbstractAction {

  public ShowNotesHommeFemmeAction() {
   super("Notes du dernier exam H/F");
  }

  public void actionPerformed(ActionEvent e) {
   notesDernierExamHommeFemmeJdialog = new JDialog();
   notesDernierExamHommeFemmeJdialog.setTitle("Notes du dernier exam H/F");

   notesDernierExamHommeFemmeJdialog.pack();
   notesDernierExamHommeFemmeJdialog.setVisible(true);
  }
 }
 ...
```



Second bouton

On doit ensuite calculer les jeux de données qui nous intéressent.

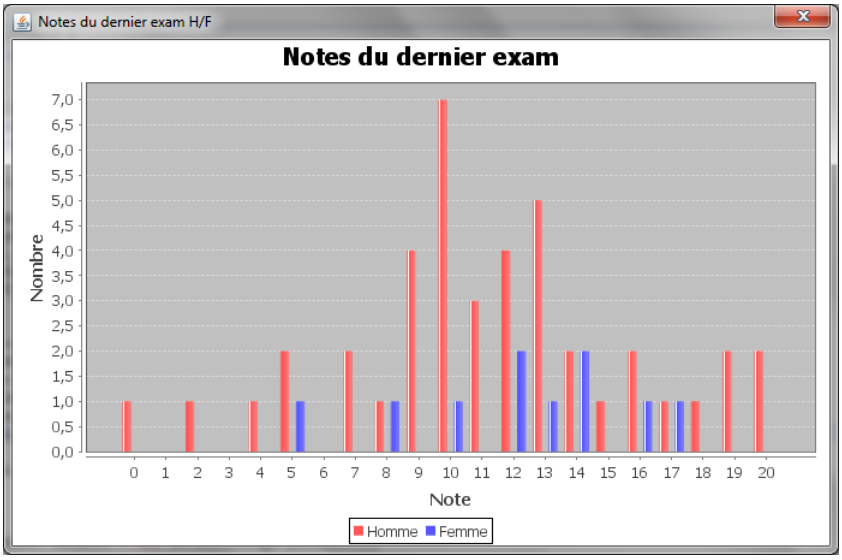
Calcul des séries

```
  ...

 public void actionPerformed(ActionEvent e) {
  notesDernierExamHommeFemmeJdialog = new JDialog();
  notesDernierExamHommeFemmeJdialog.setTitle("Notes du dernier exam H/F");

  // Calcul du nombre de copie sur chaque note

  final Map<Integer, Integer> repartitionHomme = new HashMap<Integer, Integer>();
  final Map<Integer, Integer> repartitionFemme = new HashMap<Integer, Integer>();
  for (int i = 0; i <= 20; i++) {
   repartitionHomme.put(i, 0);
   repartitionFemme.put(i, 0);
  }

  final List<NoteEleve> notes = modele.getNotes();
  for (NoteEleve noteEleve : notes) {
   Double note = noteEleve.getNote();
   if (noteEleve.getEleve().getSexe() == Sexe.FEMME) {
    incrementNb(note, repartitionFemme);
   } else {
    incrementNb(note, repartitionHomme);
   }
  }

  notesDernierExamHommeFemmeJdialog.pack();
  notesDernierExamHommeFemmeJdialog.setVisible(true);
 }
}

private void incrementNb(Double note, Map<Integer, Integer> repartition) {

 int ceil = (int) Math.ceil(note);
 Integer nb = repartition.get(ceil);
 nb++;
 repartition.put(ceil, nb);
}
```

Ce calcul est un peu complexe et n'est pas important pour suivre ce tutoriel. Il permet uniquement d'avoir des séries à utiliser pour créer le graphique en barres.

Enfin, on peut générer le graphe.

Barres
```
...

public void actionPerformed(ActionEvent e) {
 ...

 final DefaultCategoryDataset dataset = new DefaultCategoryDataset();

 for (int i = 0; i <= 20; i++) {
  dataset.addValue(repartitionHomme.get(i), "Homme", new Integer(i));
 }
 for (int i = 0; i <= 20; i++) {
  dataset.addValue(repartitionFemme.get(i), "Femme", new Integer(i));
 }

 final JFreeChart barChart = ChartFactory.createBarChart("Notes du dernier
exam", "Note", "Nombre",
     dataset, PlotOrientation.VERTICAL, true, true, false);

 final ChartPanel cPanel = new ChartPanel(barChart);

 notesDernierExamHommeFemmeJdialog.getContentPane().add(cPanel, CENTER);

 notesDernierExamHommeFemmeJdialog.pack();
 notesDernierExamHommeFemmeJdialog.setVisible(true);
}
...
```



Barres

IV - Conclusions

Globalement, c'est donc assez simple d'utiliser jfreechart pour créer des graphiques. La librairie permet de générer de nombreux autres types de graphes (cf. **captures sur le site de jfreechart**). En outre, de nombreuses options sont disponibles. Je vous conseille d'ailleurs de jouer avec les options (transformez par exemple les *true* en *false*) utilisées dans ce tutoriel pour en découvrir les impacts.

Le code final de ce tutoriel est disponible dans le fichier ZIP **notes3.zip.**

Dans ce tutoriel, je n'ai présenté que deux types de graphiques. En partant du même jeu de test, on peut imaginer bien d'autres graphes et représentations diverses. J'invite donc les lecteurs à m'envoyer leurs propositions, que j'intégrerai à l'article.

Retrouvez les tutoriels de la série "en 5 minutes" sur developpez.com à l'adresse **http://thierry-leriche-dessirier.developpez.com/#page_articles**

V - Remerciements

Je tiens à remercier, en tant qu'auteur de tutoriel rapide, toutes les personnes qui m'ont aidé et soutenu. Je pense tout d'abord à mes collègues qui subissent mes questions au quotidien mais aussi à mes contacts et amis du web, dans le domaine de l'informatique ou non, qui m'ont fait part de leurs remarques et critiques. Bien entendu, je n'oublie pas l'équipe de developpez.com qui m'a guidé dans la rédaction de cet article et m'a aidé à le corriger et le faire évoluer, principalement sur le forum.

Plus particulièrement j'adresse mes remerciements, par ordre alphabétique, à *jacques_jean*, *Gueritarish* et ***keulkeul***

VI - Annexes

VI-A - Liens

Le site web de jfreechart est le suivant
http://www.jfree.org/jfreechart

Pour suivre cet article, je conseille la lecture des tutoriels
"Importer un projet Maven dans Eclipse en 5 minutes" disponible à l'adresse
http://thierry-leriche-dessirier.developpez.com/tutoriels/java/importer-projet-maven-dans-eclipse-5-min
et "Afficher un tableau avec un Table Model en 5 minutes" disponible à l'adresse
http://thierry-leriche-dessirier.developpez.com/tutoriels/java/afficher-tableau-avec-tablemodel-5-min

VI-B - Les fichiers importants en entier

pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/
maven-4.0.0.xsd">
 <modelVersion>4.0.0</modelVersion>

 <groupId>com.thi</groupId>
 <artifactId>notes</artifactId>
 <version>1.0-SNAPSHOT</version>
 <packaging>jar</packaging>

 <name>Notes</name>
```

pom.xml

```
  <description>Programme d'exemple.</description>
  <url>http://www.thierryler.com</url>

  <licenses>
   <license>
    <name>Copyright ©1995-2011 thierryler.com et Copyright ©2011 Developpez.com</name>
    <comments>Les sources présentés sur cette page sont libres de droits, et vous pouvez ... </
 comments>
   </license>
  </licenses>

  <developers>
   <!-- Thierry -->
   <developer>
    <name>Thierry Leriche-Dessirier</name>
    <roles>
     <role>Developper</role>
    </roles>
    <organization>ICAUDA</organization>
   </developer>

  </developers>

  <properties>
   <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
   <junit.version>4.8.2</junit.version>
   <jfreechart.version>1.0.14</jfreechart.version>
  </properties>

  <dependencyManagement>
   <dependencies>
    <!-- Junit -->
    <dependency>
     <groupId>junit</groupId>
     <artifactId>junit</artifactId>
     <version>${junit.version}</version>
     <scope>test</scope>
    </dependency>

    <!-- jfreechart -->
    <dependency>
     <groupId>org.jfree</groupId>
     <artifactId>jfreechart</artifactId>
     <version>${jfreechart.version}</version>
    </dependency>
   </dependencies>
  </dependencyManagement>

  <dependencies>
   <!-- Junit -->
   <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <scope>test</scope>
   </dependency>

   <!-- jfreechart -->
   <dependency>
    <groupId>org.jfree</groupId>
    <artifactId>jfreechart</artifactId>
   </dependency>
  </dependencies>

 </project>
```

NotesJFrame.java

```
 package com.thi.notes.ihm;

 import static java.awt.BorderLayout.CENTER;
 import static java.awt.BorderLayout.SOUTH;
```

NotesJFrame.java

```
import static org.jfree.chart.ChartFactory.createPieChart;
import static org.jfree.chart.ChartFactory.createBarChart;
import static org.jfree.chart.plot.PlotOrientation.VERTICAL;

import java.awt.Dimension;
import java.awt.event.ActionEvent;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

import javax.swing.AbstractAction;
import javax.swing.JButton;
import javax.swing.JDialog;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTable;

import org.jfree.chart.ChartFactory;
import org.jfree.chart.ChartPanel;
import org.jfree.chart.JFreeChart;
import org.jfree.data.category.DefaultCategoryDataset;
import org.jfree.data.general.DefaultPieDataset;

import com.thi.notes.domain.NoteEleve;
import com.thi.notes.domain.Sexe;

public class NotesJFrame extends JFrame {

 /**
  * serialVersionUID
  */
 private static final long serialVersionUID = 3928008548751894521L;

 private NotesModele modele;
 private JTable table;

 private JDialog ratioHommeFemmeJdialog;
 private JDialog notesDernierExamHommeFemmeJdialog;

 public NotesJFrame() {
  super();
  setTitle("Notes des élèves");
  setPreferredSize(new Dimension(500, 400));
  setDefaultCloseOperation(EXIT_ON_CLOSE);

  modele = new NotesModele();

  table = new JTable(modele);
  table.setAutoCreateRowSorter(true);
  table.setDefaultRenderer(Sexe.class, new SexeCellRenderer());
  table.getColumnModel().getColumn(4).setCellRenderer(new NoteCellRenderer());

  getContentPane().add(new JScrollPane(table), CENTER);

  JPanel boutons = new JPanel();
  boutons.add(new JButton(new ShowRatioHommeFemmeAction()));
  boutons.add(new JButton(new ShowNotesHommeFemmeAction()));

  getContentPane().add(boutons, SOUTH);

  pack();
 }

 private class ShowRatioHommeFemmeAction extends AbstractAction {

  private ShowRatioHommeFemmeAction() {
   super("Ratio h/f");
  }
```

NotesJFrame.java

```
    @Override
    public void actionPerformed(ActionEvent arg0) {
     ratioHommeFemmeJdialog = new JDialog();
     ratioHommeFemmeJdialog.setTitle("Ratio H/F");

     // Calcul du ratio
     final List<NoteEleve> notes = modele.getNotes();

     int nombreFemmes = 0;
     int nombreHommes = 0;

     for (NoteEleve noteEleve : notes) {
      if (noteEleve.getEleve().getSexe() == Sexe.FEMME) {
       nombreFemmes++;
      } else {
       nombreHommes++;
      }
     }

     final DefaultPieDataset pieDataset = new DefaultPieDataset();

     pieDataset.setValue("Femme", nombreFemmes);
     pieDataset.setValue("Homme", nombreHommes);

     final JFreeChart pieChart = createPieChart("Ratio H/F", pieDataset, true, false, false);
     final ChartPanel cPanel = new ChartPanel(pieChart);

     ratioHommeFemmeJdialog.getContentPane().add(cPanel, CENTER);

     ratioHommeFemmeJdialog.pack();
     ratioHommeFemmeJdialog.setVisible(true);
    }

   }

   private class ShowNotesHommeFemmeAction extends AbstractAction {

    private static final long serialVersionUID = -2594764540109647705L;

    public ShowNotesHommeFemmeAction() {
     super("Notes du dernier exam H/F");
    }

    public void actionPerformed(ActionEvent e) {
     notesDernierExamHommeFemmeJdialog = new JDialog();
     notesDernierExamHommeFemmeJdialog.setTitle("Notes du dernier exam H/F");

     // Calcul du nombre de copie sur chaque note

     final Map<Integer, Integer> repartitionHomme = new HashMap<Integer, Integer>();
     final Map<Integer, Integer> repartitionFemme = new HashMap<Integer, Integer>();
     for (int i = 0; i <= 20; i++) {
      repartitionHomme.put(i, 0);
      repartitionFemme.put(i, 0);
     }

     final List<NoteEleve> notes = modele.getNotes();
     for (NoteEleve noteEleve : notes) {
      Double note = noteEleve.getNote();
      if (noteEleve.getEleve().getSexe() == Sexe.FEMME) {
       incrementNb(note, repartitionFemme);
      } else {
       incrementNb(note, repartitionHomme);
      }
     }

     final DefaultCategoryDataset dataset = new DefaultCategoryDataset();

     for (int i = 0; i <= 20; i++) {
      dataset.addValue(repartitionHomme.get(i), "Homme", new Integer(i));
```

NotesJFrame.java

```
    }

    for (int i = 0; i <= 20; i++) {
     dataset.addValue(repartitionFemme.get(i), "Femme", new Integer(i));
    }

    final JFreeChart barChart = createBarChart("Notes du dernier exam", "Note", "Nombre", dataset,
  VERTICAL, true, true, false);

    final ChartPanel cPanel = new ChartPanel(barChart);

    notesDernierExamHommeFemmeJdialog.getContentPane().add(cPanel, CENTER);

    notesDernierExamHommeFemmeJdialog.pack();
    notesDernierExamHommeFemmeJdialog.setVisible(true);
   }
  }

  private void incrementNb(Double note, Map<Integer, Integer> repartition) {

   int ceil = (int) Math.ceil(note);
   Integer nb = repartition.get(ceil);
   nb++;
   repartition.put(ceil, nb);
  }

 }
```