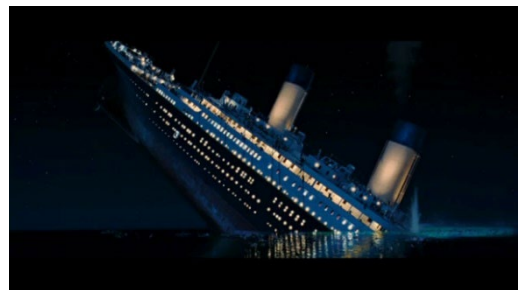


TP1 –Rappels de Java et Librairie JFreeChart

R3.04 QUALITE DE DEVELOPPEMENT

L'objectif de ce TP est à la fois de reprendre en main la programmation en Java, ainsi que de découvrir une bibliothèque graphique, *JFreeChart*, qui permet de dessiner facilement des diagrammes de statistiques élégants.

Le sujet du TP est l'affichage d'informations sur les passagers du Titanic sous forme de graphiques. Les données sont fournies sur Moodle via un fichier texte "*titanic.dbf*", que vous importerez dans votre programme. Téléchargez-le et ouvrez-le pour voir sa structure : les 5 premières lignes sont des commentaires vous permettant de comprendre le codage, ensuite il s'agit d'une série de quintuplés (Id, classe, âge, sexe, survivant) correspondant à la liste des personnes présentes sur le bateau.



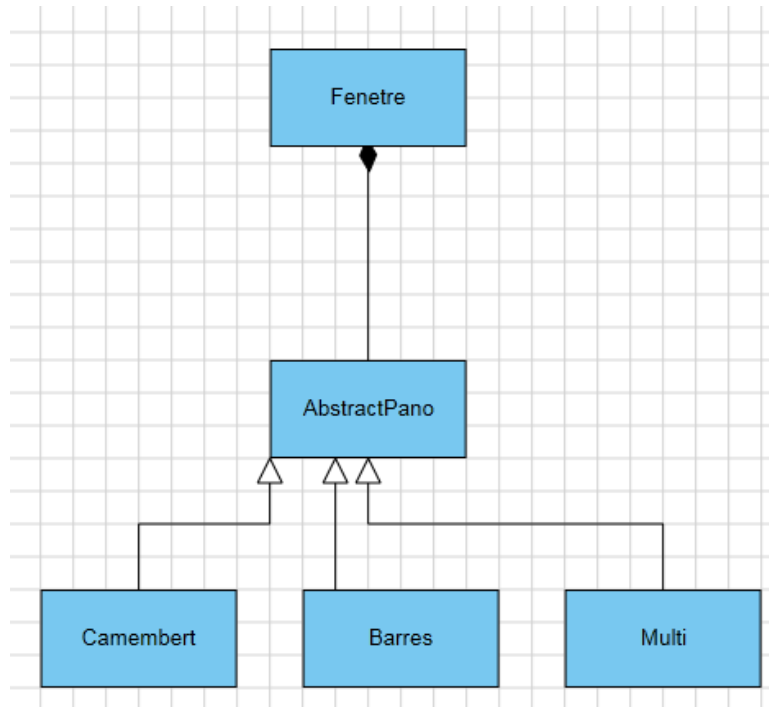
Pour tout le TP, vous êtes libres de rajouter les fonctions nécessaires à la bonne réalisation de votre programme (par ex accesseurs/mutateurs, fonctions d'affichage...).

I. LE MODELE : LES PASSAGERS DU TITANIC

- Ecrivez une classe *Passager* qui contient les 5 attributs correspondant aux 5 informations contenues dans le fichier, à savoir Identifiant (String), classe, âge, sexe, survivant (entiers).
- Ecrivez une classe *Data* qui contient comme attribut une liste de Passagers. Pour cette classe, codez une méthode de chargement qui lit le fichier "*titanic.dbf*", crée les instances des passagers et les stokes dans la liste. Pensez à utiliser la classe Scanner (<https://docs.oracle.com/javase/8/docs/api/java/util/Scanner.html>)
- Testez votre fonction dans la classe principale du projet.



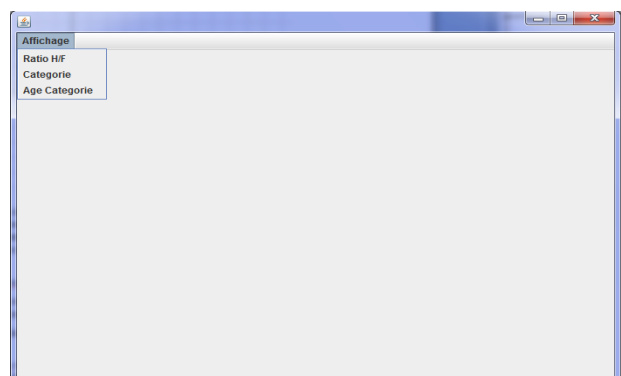
II. LA VUE : FENETRE PRINCIPALE



On vous demande pour le moment de construire une simple fenêtre avec un menu à trois items qui affichera trois types de diagrammes différents (cf. screenshot ci-dessous). Ces diagrammes seront intégrés dans des panneaux. Comme vous l'avez vu l'année dernière, puisque les diagrammes seront modifiés à la volée à chaque fois que l'utilisateur cliquera sur un item, l'utilisation d'un schéma d'héritage est recommandé. Cela donne donc les classes suivantes :

- Une classe abstraite *AbstractPano*, qui hérite de *JPanel*.
- Une classe *Fenêtre*, qui correspond à la fenêtre principale, qui contiendra notamment un *AbstractPano*, la gestion du menu...
- Les trois classes correspondant aux différents diagrammes, héritant toutes de la classe abstraite.

- Codez les classes *Fenetre* et *AbstractPano*. Pensez à raccorder le diagramme de la vue (ci-dessus) avec celui du modèle (autrement dit, la vue doit pouvoir accéder aux données). Testez votre programme.

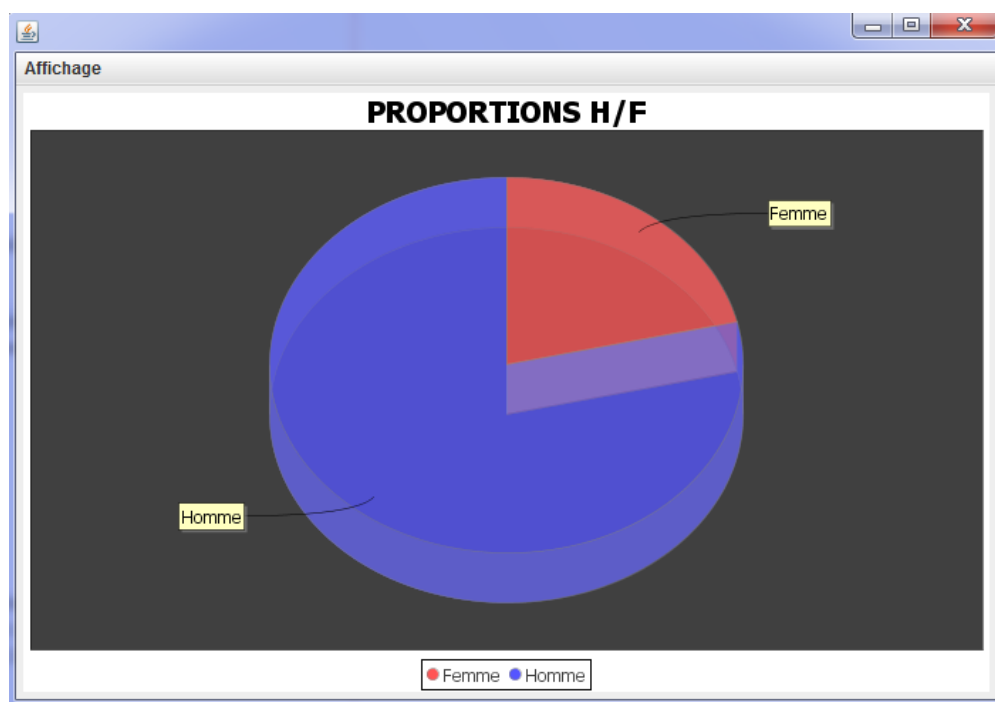


III. LA VUE : DIAGRAMME CAMEMBERT

Le fichier `.jar` associé à la bibliothèque `JFreeChart`, ainsi que `JCommon` se trouvent sous Moodle, vous pourrez les télécharger et les ajouter à votre projet (clic droit sur `Libraries` de votre projet puis `add Jar`).

Vous allez ici coder le premier diagramme, il s'agit d'un diagramme camembert qui affiche le ratio Hommes/Femmes présents sur le bateau. En anglais, un tel diagramme s'appelle un "`Pie Chart`".

Il pourra ressembler à cela (vous n'êtes pas obligés de choisir le mode d'affichage 3D).



Le lien ci-dessous vous aidera à comprendre ce qui suit

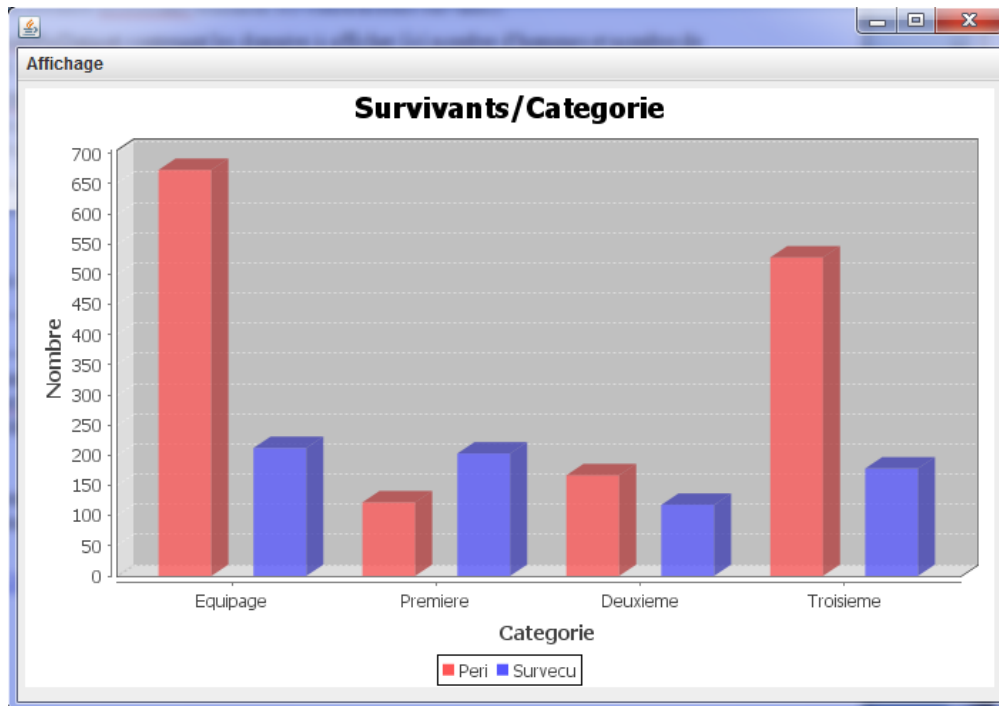
<http://www.java2s.com/Code/Java/Chart/JFreeChartPieChart3DDemo1.htm>

Pour le réaliser, la librairie `JFreeChart` demande les constructions suivantes :

- Un `DefaultPieDataset` contenant les données à afficher (ici nombre d'hommes et nombre de femmes). En général, on l'initialise avec la méthode `setValue`.
 - Puis un `JFreeChart`, qui contiendra à proprement parler le camembert. Celui-ci sera créé via la méthode statique `createPieChart3D` de la classe `ChartFactory`. Vous pourrez modifier les éléments graphiques de ce camembert en récupérant sa vue (méthode `getPlot`).
 - Enfin, on crée un panneau (`ChartPanel`) contenant le chart construit ci-dessus, et on l'ajoute au panneau Camembert.
- Codez la classe `Camembert` et testez cet affichage. Vous penserez à mettre à jour la fenêtre avec le nouveau diagramme et rafraichir son contenu.

IV. LA VUE : DIAGRAMME BARRES

Le second diagramme correspond aux statistiques suivantes, sous forme de diagramme en barres (*CategoryDataset*): pour chacune des 4 catégories (équipage, 1^{ère} classe, 2^{ème}, 3^{ème}), affichez le nombre de survivants et de morts. Voici un Screenshot d'exemple :



Le fonctionnement du diagramme bâtons est similaire à celui du camembert, à la différence que vous aurez besoin d'un *CategoryDataSet* pour stocker les données à afficher. Suivez le lien ci-dessous pour vous aider :

<http://www.java2s.com/Code/Java/Chart/JFreeChartVerticalBarChart3DDemo1.htm>

- Codez la classe *Barres* qui contient ce diagramme. Testez-le.
- Gérez les événements de la fenêtre pour passer de la vue camembert à la vue bâtons (et réciproquement) en cliquant sur les items du menu.

V. LA VUE : DIAGRAMME MULTI

- Codez le dernier panneau (Multibarres) qui contient 4 diagrammes bâtons horizontaux, avec les infos suivantes :
 - nombre de passagers par classe
 - nombre de survivants par classe
 - nombre de passager par âge (adulte ou enfant)
 - nombre de survivants par âge

Ci-dessous un Screenshot d'exemple, et une URL pour vous aider :

<http://www.java2s.com/Code/Java/Chart/JFreeChartStackedBarChartDemo2.htm>

