



## BPO

# TP 1 AWAZ : un peu de kurde en musique

En Kurde, **awaz** signifie Mélodie. C'est le thème de ce premier Tp.

### Première partie : Mise en place de l'environnement de travail

- Créez le répertoire de nom **java** destiné à contenir tous les programmes que vous allez écrire pour **BPO**. Dans ce répertoire, créez un répertoire nommé **testAwaz**. Recopiez le fichier **MainAwaz.java** fourni sur Arche dans votre répertoire **java/testAwaz**. Vous obtenez l'arborescence de fichiers ci-contre.
- Ouvrez un terminal, placez-vous dans le répertoire **java**. Lancez une première compilation avec la commande :



```
javac -encoding "iso-8859-1" testAwaz/MainAwaz.java
```

L'option **encoding** permet d'accepter les accents (en fonction de votre ordinateur et de votre environnement de développement, cette option peut être inutile). Attention au copier/coller : certains caractères ne supportent pas le voyage entre cet énoncé en pdf et votre fenêtre "terminal"...

La compilation se passe mal, une lecture attentive du message d'erreur vous apprend que le compilateur ne trouve pas la classe **Awaz**. Pas de panique, c'est normal.

- Soyez curieux et ouvrez le fichier **MainAwaz.java** avec votre éditeur de textes favori (pas d'IDE comme Eclipse ou IntelliJ, on en utilisera plus tard). La classe **MainAwaz** ne contient qu'une fonction **main** qui crée une instance de **Awaz** (une mélodie). Cette classe fait partie des nombreuses ressources nécessaires à la réalisation de ce Tp.
- Deux ressources sont nécessaires : une archive Java (**awaz.jar**) contenant des classes toutes faites et la documentation associée (**javadoc.zip**). L'utilisation de ces ressources diffère en fonction de la machine sur laquelle vous travaillez.

<b>Vous travaillez sur votre propre ordinateur</b>	Les deux ressources nécessaires sont sur Arche. Il faut copier ces 2 ressources sur votre ordinateur, dans le répertoire <b>ressourcesBPO/awaz</b> , placé au même niveau que le répertoire <b>java</b> .  La ressource <b>javadoc.zip</b> doit être décompressée.
<b>Vous travaillez sur une machine de la FST (salle atelis 3ème niveau du bâtiment)</b>	Toutes les ressources nécessaires sont dans <b>/opt/depot/BPO</b> . Il n'est pas question de recopier ces ressources dans votre propre répertoire ; vous allez simplement apprendre à les utiliser, lors de la compilation et de l'exécution.

- La commande **javac** permet de compiler un fichier. Pour l'utiliser, il faut toujours se placer au-dessus du répertoire qui contient le fichier (ici, se placer dans le répertoire **java**, donc au-dessus du répertoire **testAwaz** qui contient le fichier **Awaz.java**). Dans la commande ci-dessous, l'option **classpath** permet de préciser l'emplacement des différentes ressources (les ressources communes sont dans le dépôt commun et vos ressources personnelles dans le répertoire courant noté par un point **.**).

Perso	<code>javac -classpath ../ressourcesBPO/awaz/awaz.jar:. -encoding "iso-8859-1" testAwaz/MainAwaz.java</code>
FST	<code>javac -classpath /opt/depot/BPO/awaz.jar:. -encoding "iso-8859-1" testAwaz/MainAwaz.java</code>

Si aucune erreur ne s'affiche, la compilation crée le fichier **testAwaz/MainAwaz.class** contenant la description compilée de la classe **testAwaz.MainAwaz**.

- La commande **java** permet d'exécuter les instructions contenues dans la fonction **main** d'une classe dont on fournit le nom complètement qualifié. L'option **classpath** est indispensable également (mais pas l'option **encoding**).

Perso	<code>java -classpath ../ressourcesBPO/awaz/awaz.jar:. testAwaz.MainAwaz</code>
FST	<code>java -classpath /opt/depot/BPO/awaz.jar:. testAwaz.MainAwaz</code>

## Deuxième partie : Création de mélodies

Les questions de cette partie doivent être résolues dans l'ordre. Elles consistent toutes à compléter la fonction **main** de **MainAwaz.java**. Pour répondre à ces questions, vous aurez besoin de consulter la documentation des ressources qui se trouve dans des fichiers HTML. Pour visualiser cette documentation dans un navigateur, il suffit de double-cliquer sur le fichier **index.html**.

En annexe de cet énoncé, vous trouverez des rudiments de solfège, qui peuvent vous aider à comprendre certaines notions musicales si besoin.

- Ajoutez quelques notes dans la mélodie (fonction **add** de **Awaz**).
- Affichez la mélodie (fonction **toString**).
- Affichez le nombre de notes de la mélodie.
- Transposez la mélodie de un ton et affichez la mélodie ainsi transposée.
- Effacez la mélodie avant d'en refaire une nouvelle avec quelques notes.
- Parcourez toutes les notes de la mélodie, pour les afficher sur la même ligne, séparées par un espace.
- Enregistrez la mélodie dans un fichier dont le nom est lu sur l'entrée standard.
- Ajoutez des notes à la mélodie.
- Affichez la mélodie en format ABC (<http://abcnotation.com/examples>).
- Affichez et jouez la partition de la mélodie (utilisation de la classe **AwazImage**). Après utilisation de cette nouvelle ressource, l'exécution de votre programme ne s'arrête pas, il faut tuer le processus par "ctrl-C" dans la fenêtre terminal.

## Troisième partie : Une application interactive

Dans le même répertoire **testAwaz**, écrivez une nouvelle classe **Menu** sur le modèle de la classe **MainAwaz**. La fonction **main** affiche sur la sortie standard, un menu permettant à l'utilisateur de réaliser toutes les opérations possibles sur une mélodie (ajout de notes, transposition, affichage, sauvegarde, *etc.*). Les données sont lues sur l'entrée standard.

# ANNEXE

## Un peu de culture solfégique

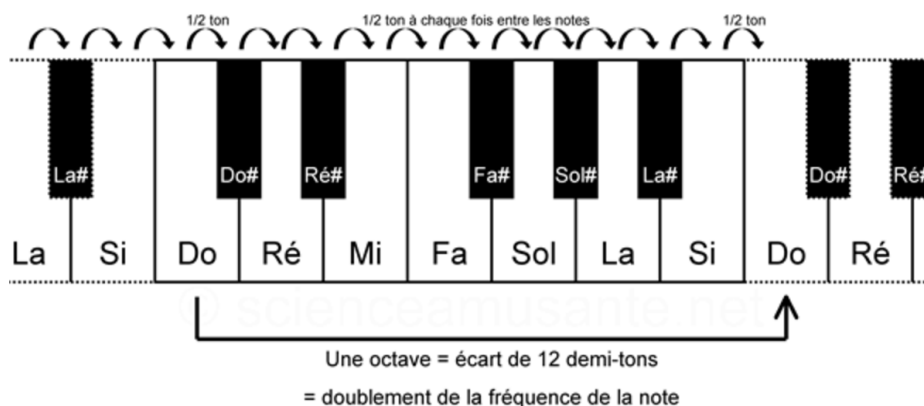
<http://www.theoriedelamusique.com>

(Théorie de la musique, Danhauser, 1996, Editions Henry Lemoine)

Au VI<sup>e</sup> siècle, Boèce utilisait les lettres de l'alphabet pour désigner les notes de musique : A, B, C, D, E, F et G. Au XI<sup>e</sup> siècle, Guido d'Arezzo a imaginé des noms de notes en prenant les premières syllabes de l'hymne à Saint Jean-Baptiste : Ut, Ré, Mi, Fa, Sol, La. La note Si, dont le nom est formé des deux initiales du dernier vers de l'hymne a été ajoutée à la fin du XVI<sup>e</sup> siècle. A la même époque, la première syllabe Ut a été remplacée par Do, plus facile à dire.

Chez nos voisins anglo-saxons/allemands, subsiste la notation alphabétique (A pour La, B pour Si, *etc.*).

Sur un clavier, on trouve plus que sept notes car on peut ajouter des altérations (bémol b et dièse #) à chaque note. Ainsi, au total, sur une octave, on trouve 12 notes : les 7 notes de base plus des notes altérées. Entre chacune de ces 12 notes, il y a un  $\frac{1}{2}$  ton. À noter que Do dièse et Ré bémol se jouent sur la même touche, comme Ré dièse et Mi bémol.



Transposer une mélodie revient à opérer une translation sur chaque note : la note Do transposée de trois tons devient la note Fa #.

Sur une partition, les notes représentent des durées et des hauteurs de son : selon leurs différentes formes, les notes expriment des durées différentes ; selon leurs positions sur la portée, les notes expriment des hauteurs de son différentes. Il existe sept formes de notes : la ronde, la blanche, la noire, la croche, la double croche, la triple croche et la quadruple croche. Une ronde vaut deux blanches, une blanche vaut deux noires, une noire vaut deux croches, *etc.*