

## TP N° 1

### Présentation du logiciel REALJ

Vous voici au pied du mur (de la machine pourrait-on dire). Nous allons vous apprendre les rudiments de la programmation. Sachez que vous allez devoir maîtriser un savoir-faire fait de myriades d'opérations simples, mais si nombreuses qu'il ne faut passer à côté d'aucune, sous peine de se sentir très vite dépassé.

Ces feuilles, conçues de façon complémentaire au cours, devront être lues, comprises et effectuées dans leur intégralité. A l'avenir, elles se composeront de trois parties distinctes :

- un rappel du cours, intitulé *Ça va mieux en le révisant ...*
- un ensemble d'exercices d'application du cours allant de difficulté croissante
- une dernière partie **Approfondissement**, qui, si elle va au-delà de la simple application du cours, mérite une attention toute particulière car elle sert à consolider les connaissances acquises.

Si par malchance vous ne réalisez pas l'intégralité de la feuille lors de votre séance de TP, vous devez absolument la finir avant la séance suivante. Vous trouverez toujours des salles en libre-service afin de travailler en dehors des TP.

Encore une fois, l'apprentissage de la programmation s'appuie sur tant de compétences différentes (réflexion, rigueur, mémoire, sagacité, persévérance... voire débrouillardise) que vous devez à proprement parler entraîner votre esprit à cette nouvelle gymnastique.

## 1 Présentation de REALJ

Commencez par démarrer une nouvelle session WINDOWS, quitte à fermer celle que vous pourriez trouver ouverte.

**Exercice 1)** Lancez le logiciel REALJ<sup>1</sup> à partir du bouton *Démarrer* de WINDOWS qui se trouve dans le coin inférieur gauche de votre écran.

Il apparaît la fenêtre du logiciel REALJ, ainsi qu'une toute petite fenêtre où vous pouvez lire le *truc du jour*. C'est instructif et vous pouvez en lire plusieurs en cliquant sur le bouton *Next tip...* mais il faut en laisser pour les séances suivantes ; aussi finissez par fermer cette fenêtre.

---

1. Spécialement pour ceux qui voudraient installer le logiciel REALJ sur leur ordinateur personnel sous WINDOWS afin de programmer en JAVA chez eux, une documentation détaillée est disponible sur la page INTERNET du cours.

Pour l'instant, la fenêtre propose 4 menus déroulants : File, Project, View et Help ainsi qu'une quinzaine d'icônes sur la barre d'outils. Nous allons les numéroter de gauche à droite, histoire de nous y retrouver. Parmi ces icônes, il y en a qui sont *grisées* : elles sont momentanément inopérantes. On remarque également, tout en bas, une *barre d'état*. Elle comporte peu d'informations pour l'instant.

**Exercice 2)** Balladez le curseur de la souris en effectuant des pauses afin de faire apparaître les petites bulles d'aide auprès de chaque icône.

Vous savez que les icônes 1 (la page blanche) et 2 (le classeur jaune) signifient respectivement New et Open. Mais avez-vous remarqué que dans l'angle inférieur gauche de la fenêtre REALJ, il y a une version longue (mais toujours en anglais !) de la signification des icônes ?

**Exercice 3)** Cliquez sur la première icône New. Une boîte de dialogue apparaît : sélectionnez RealJ Project puis cliquez deux fois sur le bouton OK. En partie gauche de la fenêtre, il apparaît une petite page blanche. C'est là qu'apparaîtront les noms des classes de votre projet. Nous allons donner un nom à ce projet : pour cela, dans le menu File, sélectionnez Save as ... et donnez par exemple le nom TP1.jpr au projet du jour que vous sauverez sur votre disquette.

Ici, jpr est le suffixe consacré et signifie *Java P*roject. Il faut savoir que sous le logiciel REALJ<sup>2</sup>, les fichiers JAVA sont organisés en *projets*. Un projet est juste un ensemble de fichiers qui porte un nom. Nous vous proposons de créer un projet par séance ainsi tous les fichiers d'une séance de programmation appartiendront au même projet. Au début de chaque séance, nous créerons donc un nouveau projet, comme nous venons de le faire avec le projet TP1.jpr.

Venons-en à notre premier programme :

**Exercice 4)** Cliquez à nouveau sur l'icône de la page blanche, mais cette fois-ci, vous choisirez de créer un nouveau Java Source File. Sentez-vous poindre l'angoisse de la page blanche ?

Tôt ou tard, pensez à donner un nom à ce fichier. Ce doit être le nom de la classe suivi du suffixe .java. En l'occurrence, appelons-le Essai1.java.

**En JAVA, le nom d'une classe commence impérativement par une majuscule.**

Pour baptiser ce fichier, utilisez une nouvelle fois Save as ... du menu File. On aurait pu faire la même sauvegarde en commençant par cliquer sur l'icône 3 (la disquette).

**Exercice 5)** Nous sommes à même de rédiger notre premier programme sur la grande page

---

2. Il ne faut pas confondre JAVA et REALJ. JAVA est un langage de programmation alors que REALJ est le logiciel que nous avons choisi pour que vous découvriez le langage JAVA dans un environnement simple et ergonomique.

blanche : tapez-y le programme `Essai1.java` du cours ... Ah oui ! il faut apprendre à taper les accolades (AltGr {, AltGr }), les crochets (AltGr [, AltGr ]) et les guillemets afin que la présentation soit lisible. Une touche de *tabulation* située en haut à gauche de votre clavier vous permet de décaler votre texte à droite.

Pensez à sauvegarder fréquemment votre programme : la première fois, il faut lui donner un nom mais une fois que le fichier est baptisé, il suffit de cliquer sur l'icône 3 (la disquette) pour enregistrer toutes modifications supplémentaires.

Vous avez peut-être remarqué que la liste des menus s'est agrandie et que des icônes jusqu'alors inactives ont repris des couleurs... Parmi elles, les icônes 4 (les ciseaux), 5 (la copie) et 6 (le pot de colle) assurent les opérations de *couper/copier/coller* que vous verrez très bientôt en bureautique.

Chaque fois que vous apprenez la signification d'une icône, ayez donc le réflexe de rechercher comment la même chose peut être faite à partir des menus.

Voici venu le moment de vérité : nous allons compiler puis exécuter ce programme. Compiler un programme `Xxx.java` permet de vérifier s'il est *syntactiquement* correct puis d'engendrer un fichier de *byte-code* du nom de `Xxx.class`.

Il ne restera plus qu'à lancer l'exécution de la classe `Xxx.class`.

**Exercice 6)** Compilez votre programme grâce à l'icône 10. Si le programme présente des erreurs, à vous de les corriger ! Sinon, vérifiez la présence du fichier `Essai1.class` dans votre répertoire. Cliquez ensuite sur l'icône 13 (la flèche jaune) pour exécuter la classe ... Zut! ça ne marche pas ! et c'est normal :

Il faut incorporer votre classe au projet : dans le menu `Project`, sélectionnez `add Java Sources Files`. Vous voyez le nom de votre classe apparaître dans la fenêtre du projet, à gauche.

Cliquez à nouveau sur l'icône 13 (toujours la flèche jaune `run`)... re zut ! ça ne marche toujours pas ! ... et c'est encore normal :

Un projet `REALJ` est un ensemble de fichiers `JAVA`. Avant de lancer une exécution, encore faut-il savoir quelle classe contient la fonction `main( . . . )` qui constituera le point de départ de l'exécution.

Bon, ici, il n'y en a qu'une... mais l'ordinateur ne le sait pas. Ainsi, vous devez toujours préciser à partir de quelle classe on lance l'exécution. Pour cela, il vous suffit d'appuyer avec le bouton de droite de la souris, sur le nom de la classe concernée (dans la fenêtre du projet) puis de sélectionner `set as main` dans le menu qui vient de surgir. Un petit carré rose signale alors

la classe sélectionnée comme point d'entrée de l'exécution.

Recommençons : cliquez sur la flèche jaune et ... admirez le résultat.  
Bravo!! Vous venez d'exécutez votre première classe JAVA.

## 2 Premiers programmes

**Exercice 7)** On a l'intention de réutiliser le programme de l'exercice précédent pour en fabriquer un autre. Sauvez ce programme sous le nom `Essai2.java`. Ajoutez-y deux lignes afin qu'il affiche :

```
Salut la fac !  
Je m'appelle ...  
Et toi ?
```

N'espérez pas que l'ordinateur vous réponde... mais compilez et exécutez cette nouvelle classe. Au besoin, aidez-vous du récapitulatif ci-dessous :

<i>saisir la classe</i>	<code>class Xxx {...}</code>
<i>sauver le fichier</i>	icône 3 (la disquette)
<i>compiler la classe</i>	icône 10
<i>incorporer la classe au projet</i>	menu <u>P</u> roject add Java Sources Files
<i>sélectionner la classe à exécuter</i>	set as main obtenu en cliquant avec le bouton de droite sur le nom de la classe voulue
<i>lancer l'exécution</i>	icône 13 (la flèche jaune)

Avant de passer à un troisième programme JAVA, passez en revue les possibilités offertes par l'ensemble des menus déroulants. Vous vous apercevrez que le menu File assure la gestion des fichiers tandis que Project est spécialisé dans celle des projets. Le menu Edit est une aide à l'édition. Vous pouvez y découvrir notamment des raccourcis clavier pour les opérations simples de *couper/copier/coller* ainsi que le très très utile `undo` qui permet d'annuler la dernière frappe. Le menu View permet seulement de faire apparaître ou disparaître les fenêtres citées ainsi que les barres d'outils et d'état.

**Exercice 8)** Dans un troisième *fichier-source*<sup>3</sup>, écrivez un programme qui affiche les deux phrases suivantes, disposées comme ci-dessous :

```
La représentation binaire de l'entier 173 est ...
```

---

3. On appelle fichier-source un fichier qui renferme un texte dans tel ou tel langage de programmation ; nos fichiers-sources sont les `Xxx.java`.

101110 est la représentation binaire de l'entier ...

On ne demande pas à l'ordinateur de calculer ces valeurs. On s'armera plutôt de papier, de crayon et surtout de sa tête pour trouver les résultats voulus afin de remplir les points de suspension.

**Exercice 9)** Nous vous proposons à présent d'écrire un nouveau programme qui affiche l'addition et la multiplication des deux nombres en binaire 1101 et 101. Pour l'addition, on devra faire afficher :

```
Addition :
      1 0 1
+   1 1 0 1
-----
    1 0 0 1 0
```

Préparez la multiplication sur papier afin de pouvoir l'afficher, tout comme l'addition. Le cas échéant, vérifiez vos calculs en convertissant ces nombres en décimal...

**Exercice 10)** Utilisez `set as main` pour sélectionner la classe de l'exercice 8. Lancez son exécution. Vous remarquerez alors que la classe qui s'exécute n'est pas forcément celle visible dans l'éditeur.

**Exercice 11)** Sachant qu'un octet est une réunion de huit *bits* (binary digits), quel est le plus grand entier dont on peut écrire la représentation binaire sur un octet ?

### 3 Avant de partir...

**Exercice 12)** Assurez-vous que vos programmes sont bien sauvegardés sur votre disquette. Vous avez sûrement déjà repéré dans le menu File comment quitter REALJ, faites-le.

A présent, fermez la session WINDOWS.