

# Travaux pratiques d'informatique N° 1

Le but de cette séance est de vous présenter deux exemples élémentaires de programmes Java, afin de vous permettre de faire connaissance avec l'**environnement de programmation intégré (IDE) Eclipse**.

**1.** Cet exercice vous demande de créer un projet dans l'environnement de programmation Eclipse, correspondant à une **application (autonome)** simple qui sera exécutée en **mode console**.

Marche à suivre :

## 1.0 Monter localement le dossier personnel distant My Documents

*Remarques :*

- le système d'exploitation Windows permet de se connecter à un dossier réseau partagé et d'assigner une lettre à la connexion afin d'accéder à ce dossier depuis le poste de travail ;

- cette opération est appelée couramment montage (ou mapping) d'un dossier distant en tant que disque dur local (ou partition locale) ;

- pour accéder au contenu d'un dossier distant monté localement, il suffit d'indiquer un chemin d'accès absolu commençant par la lettre assignée pendant la phase de montage (mapping).

- ouvrir une fenêtre Windows Explorer (par exemple, **Start → Computer**) ;
- (dans le menu **Tools...** ,) sélectionner **Map Network Drive...** ;
- remplir convenablement les zones de texte de la nouvelle fenêtre affichée, à savoir :
  - **Z:** en tant que **Drive** ;
  - **\\files~~n~~\data\username\My Documents** en tant que **Folder** ;

*Remarque : dans le chemin d'accès indiqué ci-dessus, il faut remplacer le caractère **n** à la fin du mot **files~~n~~** par le dernier chiffre de votre numéro SCIPER et le mot **username** par votre nom d'utilisateur.*

- vérifier que la case **Reconnect at logon** est bien cochée ;
- cliquer sur le bouton **Finish** (et taper le mot de passe associé à votre nom d'utilisateur, si nécessaire).

## 1.1 Préparer les sources

- sur le "disque" **Z**, créer un nouveau dossier **Informatique** contenant deux nouveaux sous-dossiers : **SourcesJava** (dans lequel seront téléchargés divers fichiers sources Java depuis le bureau virtuel du CMS) et **CMS\_Workspace** (dans lequel seront créés les futurs projets Eclipse) ;
- dans le dossier **SourcesJava**, copier (télécharger) "localement" le fichier **CP\_TP1Exo1.java** depuis le bureau virtuel du CMS ;
- ouvrir le fichier mentionné ci-dessus à l'aide d'un éditeur de texte simple (par exemple, Notepad) et identifier le nom de la classe Java qu'il définit.

*Remarque : le fichier contenant le code source d'une classe publique Java doit porter le même nom que la classe publique et avoir l'extension .java.*

## 1.2 Préparer l'environnement de programmation Eclipse et créer un nouveau projet

- ouvrir l'environnement de programmation Eclipse (en cliquant sur le bouton **Start** du bureau (**Desktop**) et en choisissant ensuite **All Programs → Eclipse → Eclipse**) ;
- dans la nouvelle fenêtre **Eclipse Launcher** affichée, indiquer d'abord le dossier **CMS\_Workspace** créé au point 1.1 en tant que dossier utilisé pour stocker les projets Eclipse (en écrivant son nom complet, c'est-à-dire le nom du dossier précédé par le chemin d'accès, dans la zone de texte **Workspace**), cocher ensuite la case **Use this as the default and do not ask again** afin d'imposer automatiquement ce choix par la suite et appuyer finalement sur le bouton **Launch** ;
- une fois l'environnement de programmation **Eclipse IDE** chargé, choisir le menu **File → New → Project...** ;
- dans la nouvelle fenêtre **New Project** affichée, sélectionner **Java Project** et appuyer sur le bouton **Next >** ;
- dans la zone de texte **Project Name**, introduire le nom du nouveau projet, à savoir **PrTP1Exo1** ;
- dans la région **JRE**, vérifier que le bouton radio **Use default JRE (currently 'jre-10.0.2')** est bien enfoncé ;
- dans la région **Project layout**, enfoncer le bouton radio **Create separate folders for sources and class files** afin de placer les sources (fichiers **.java**) et les bytecodes (fichiers **.class**) dans deux dossiers différents : **src** et, respectivement, **bin** ;
- appuyer successivement sur les boutons **Next** et **Finish**.

*Remarque : une fois le nouveau projet Eclipse créé, la **perspective Java** présente la structure arborescente du projet dans la colonne gauche (**Package Explorer**) de la fenêtre principale (**Workbench**).*

### 1.3 Créer une nouvelle classe principale (Main Class) dans le projet

- dans le **Package Explorer**, sélectionner le projet **PrTP1Exo1** ;
- dans le menu **File**, choisir **New → Class** ;
- dans la nouvelle fenêtre **New Java Class** affichée, vérifier que dans la zone de texte **Source folder** est marqué **PrTP1Exo1/src**, indiquer dans la zone de texte **Package** le nom du package auquel appartient la nouvelle classe, à savoir **cms\_tp1**, donner dans la zone de texte **Name** le nom de la classe créée, à savoir **CP\_TP1Exo1**, vérifier que le bouton radio **public** est enfoncé pour les **Modifiers** et que le nom de classe de base **java.lang.Object** est fourni dans la zone de texte **Superclass** et cocher la case **public static void main(String[] args)** ;
- appuyer sur le bouton **Finish**.

*Remarques : dans la fenêtre principale (**Workbench**) de l'environnement Eclipse (vu sous la **perspective Java**), on observe maintenant que :*

- dans la colonne de gauche (**Package Explorer**), le dossier **src** englobe le sous-dossier du package **cms\_tp1** qui lui contient le fichier source **CP\_TP1Exo1.java** ;
- la région centrale est occupée par l'éditeur Java (**Java Editor**) qui affiche le contenu du fichier **CP\_TP1Exo1.java** ;
- dans la colonne de droite (**Outline**), on présente la structure (les champs et les méthodes) de la nouvelle classe **CP\_TP1Exo1**.

### 1.4 Modifier la classe principale (Main Class) et exécuter le projet de type application (autonome) Java

- à l'aide de l'éditeur Java (**Java Editor**), ajouter dans la méthode **main** de la classe principale (**Main Class**) **CP\_TP1Exo1** une instruction qui affiche à l'exécution le message "*Bonjour ! C'est ma première application Java !*" (en prenant comme exemple le contenu du fichier téléchargé **CP\_TP1Exo1.java** du dossier **JavaSources**);
- dans le menu **File**, choisir **Save** (raccourci clavier **Ctrl + S**) afin d'enregistrer les modifications apportées au fichier source **CP\_TP1Exo1.java** ;
- dans le menu **Run**, exécuter **Run As → Java Application** ;

*Remarque : suite à l'exécution de la classe principale (**Main Class**) du projet, le message "Bonjour ! C'est ma première application Java !" est affiché dans une zone (**View**) appelée **Console** située en bas de la fenêtre principale (**Workbench**) et qui peut être rendue visible en choisissant **Window → Show View → Console**.*

### 1.5 Amélioration intéressante mais facultative du projet Java Application

- modifier le code source **.java** de sorte que le projet puisse s'exécuter interactivement comme dans la fenêtre **Console** ci-dessous.

```
Bonjour ! C'est ma première application Java !
Introduisez votre nom, svp :
Blanc
Introduisez votre prénom, svp :
Bonnet
Introduisez votre année de naissance, svp :
1998
Bonjour Bonnet Blanc !
Vous avez 20 ans !
Au revoir !
```

**2. (Facultatif)** Cet exercice vous demande de créer un projet Eclipse correspondant à une **applet swing (JApplet)** simple exécutée par l'**AppletViewer** ou à partir d'un **fichier HTML**.

Marche à suivre :

#### 2.1 Préparer les sources

- copier "localement" (télécharger) dans le dossier **SourcesJava** les fichiers **MaPremiereApplet.java** et **MonPremierHTML.html** depuis le bureau virtuel du CMS.

#### 2.2 Créer un nouveau projet

- en procédant comme au point **1.2**, créer un nouveau projet appelé **PrTP1Exo2**.

*Remarque : une fois le nouveau projet Eclipse créé, sa structure arborescente est ajoutée dans la colonne de gauche (**Package Explorer**) de la fenêtre principale (**Workbench**).*

#### 2.3 Créer une nouvelle classe applet dans le projet

- dans le **Package Explorer**, sélectionner le projet **PrTP1Exo2** ;
- dans le menu **File**, choisir **New → Class** ;

- dans la nouvelle fenêtre **New Java Class** affichée, vérifier que dans la zone de texte **Source folder** est marqué **PrTP1Exo2/src**, indiquer dans la zone de texte **Package** le nom du package auquel appartient la nouvelle classe, à savoir **cms\_tp1**, donner dans la zone de texte **Name** le nom de la classe créée, à savoir **MaPremiereApplet**, vérifier que le bouton radio **public** est enfoncé pour les **Modifieurs**, fournir dans la zone de texte **Superclass** le nom qualifié de la classe de base, à savoir **javax.swing.JApplet**, et décocher, si nécessaire, la case **public static void main(String[] args)** ;
- appuyer sur le bouton **Finish**.

*Remarque : en étudiant le contenu du fichier source **MaPremiereApplet.java** à l'aide de l'éditeur Java (**Java Editor**), on peut observer que l'environnement de programmation Eclipse a prévu une instruction qui importe la classe prédéfinie **JApplet** du package **javax.swing**.*

## 2.4 Modifier la classe applet et exécuter le projet de type Applet Java

- à l'aide de l'éditeur Java (**Java Editor**), ajouter dans la classe **MaPremiereApplet** des instructions qui affichent à l'exécution le message "*Bonjour ! C'est ma première applet Java !*" (en prenant comme exemple le contenu du fichier téléchargé **MaPremiereApplet.java** du dossier **JavaSources**) ;
- dans le menu **File**, choisir **Save** (raccourci clavier **Ctrl + S**) afin d'enregistrer les modifications apportées au fichier source **MaPremiereApplet.java** ;
- dans le menu **Run**, exécuter **Run As → Java Applet** ;

*Remarque : suite à l'exécution du projet, le message "*Bonjour ! C'est ma première applet Java !*" est affiché dans une nouvelle fenêtre **AppletViewer** placée initialement dans le coin supérieur gauche de l'écran.*

## 2.5 Créer, ajouter et modifier un fichier HTML dans le projet de type Applet Java

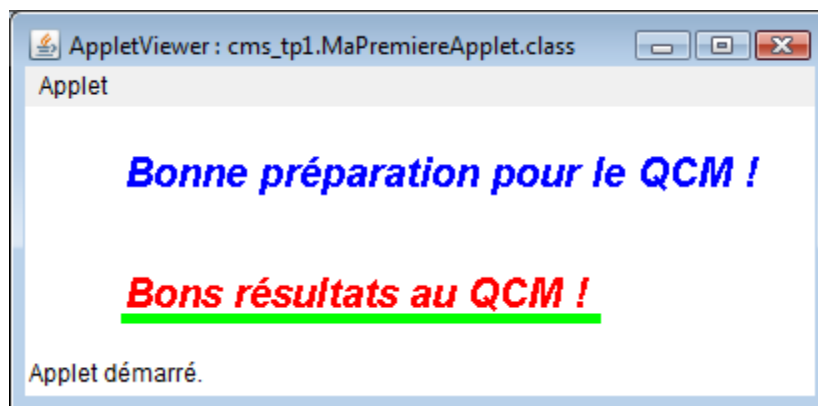
- dans le **View Package Explorer**, sélectionner le projet **PrTP1Exo2** ;
- dans le menu **File**, choisir **New → File** ;
- dans la nouvelle fenêtre **New File** affichée, sélectionner **PrTP1Exo2** comme dossier parent pour le nouveau fichier et indiquer dans la zone de texte **File Name** le nom du fichier créé, à savoir **MonPremierHTML.html** ;
- appuyer sur le bouton **Finish** ;
- dans le **Package Explorer**, cliquer avec le bouton droit de la souris sur le fichier **MonPremierHTML.html** et choisir dans le menu contextuel **Open With → HTML Editor** (ou **Web Page Editor** ou **Text Editor**) ;

- dans la fenêtre de l'éditeur choisi (**HTML Editor**) affichée dans la zone centrale de la fenêtre principale (**Workbench**), copier le contenu du fichier téléchargé **MonPremierHTML.html** et enregistrer ensuite cette modification (menu **File** → **Save** ou raccourci clavier **Ctrl + S**).

*Remarque : en dehors de l'environnement de programmation Eclipse, le fichier **MonPremierHTML.html** peut être maintenant ouvert avec un navigateur Web (browser) compatible Java (y compris **Internet Explorer**), ce qui produit un résultat similaire celui obtenu au point 2.4.*

## 2.6 Amélioration intéressante mais facultative du projet Java Applet

- modifier le code source **.java** de sorte que le résultat de l'exécution du projet par l'**AppletViewer** corresponde à la figure ci-dessous ;



*Remarques :*

- les deux messages doivent être écrits avec une police **Arial 20, grasse et italique** ;
- il faut utiliser la couleur **bleue** pour le premier message, la couleur **rouge** pour le deuxième message et la couleur **verte** pour le soulignement.

*Indications :*

- utiliser les classes **Font** et **Color** du package **java.awt** ;
- employer les méthodes **setFont( )**, **setColor( )**, **drawString( )** et **drawLine( )** de la classe **Graphics** du package **java.awt**.

En outre, pour avoir plus d'informations concernant la machine virtuelle Java, les différences entre les applications autonomes et les applets, ainsi que la syntaxe et les classes Java, vous pouvez consulter la documentation Java d'Oracle, en général, et les spécifications de l'API Java, en particulier.