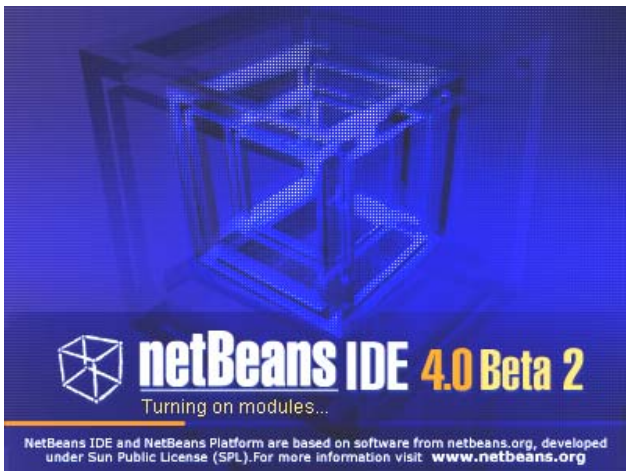


Créer une simple application Java avec netBeans

Par Ahcène BOUNCEUR

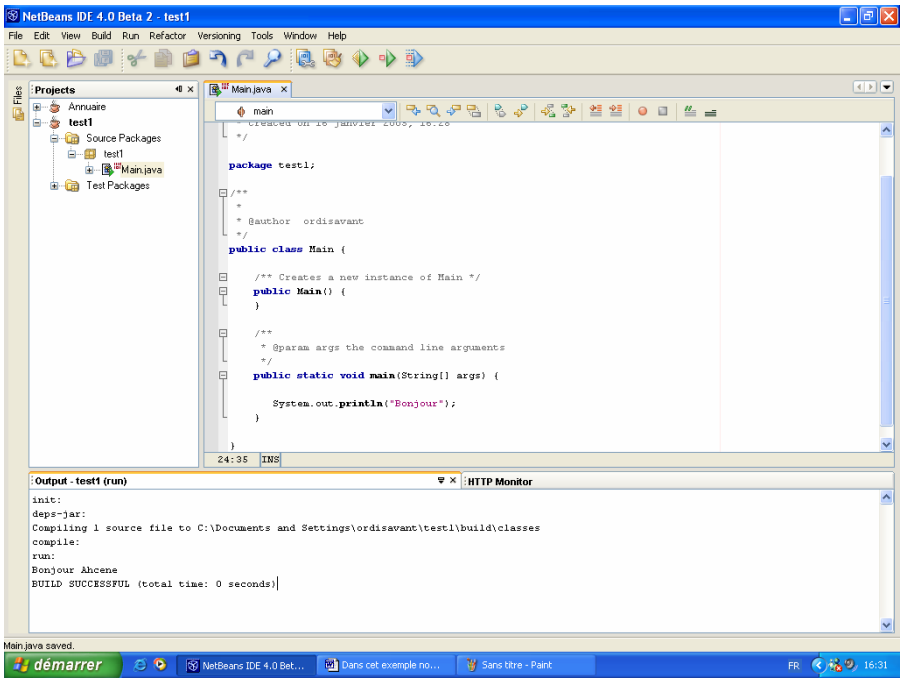
Janvier 2005

INTRODUCTION

Dans cet exemple nous allons vous montrer comment peut-on créer une simple application Java en utilisant l'IDE NetBeans. Cette application doit tout simplement afficher la chaîne de caractères "Bonjour".

Etape 1 : Création d'un nouveau projet

Lancez NetBeans.



D'abord nous devons créer un projet Java dans lequel se trouveront nos classes. Pour ce faire, allez dans le menu Fichier → Nouveau Projet (ou File → New Project), voir Figure 1.

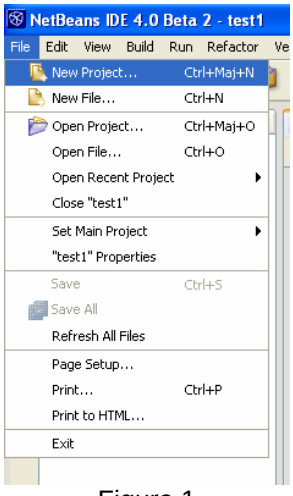


Figure 1

Puis une fenêtre s'affiche (voir Figure 2), dans cette fenêtre spécifiez le type du projet à créer. Vous choisissez donc dans categories (à gauche) le type standard, puis dans Projects (à droite) choisissez Java Application.

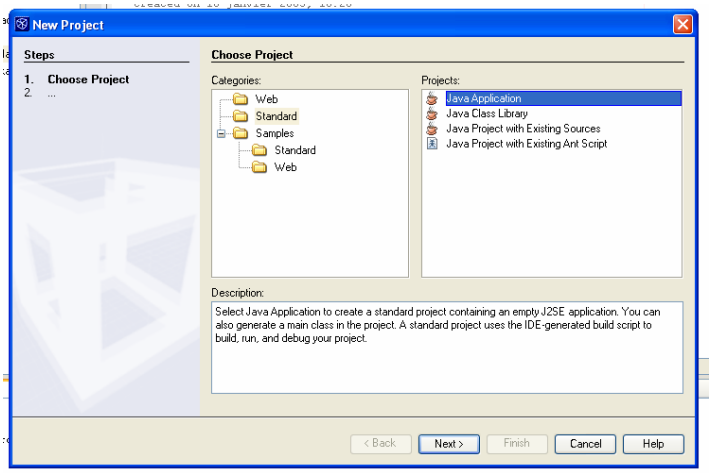


Figure 2

Puis cliquez sur le bouton Next, une autre fenêtre s'affiche (voir Figure 3), dans laquelle il faut entrer le nom du projet dans la partie Project Name, et faites entrer le nom "test_bonjour" et contrairement à ECLIPSE, dans netBeans la classe main peut être créée au même temps que le projet si la case Create Main Class est cochée. Cochez donc cette case, puis faites entrer dans le champ correspondant test_bonjour.classe_bonjour pour nommer la classe main classe_bonjour.

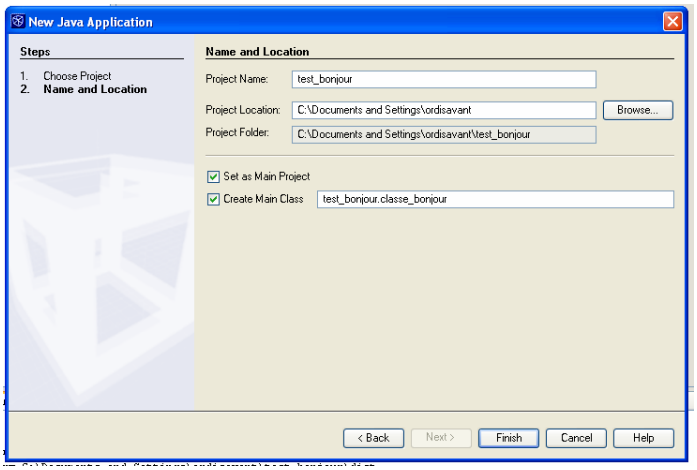
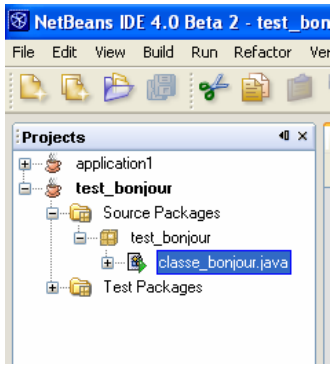
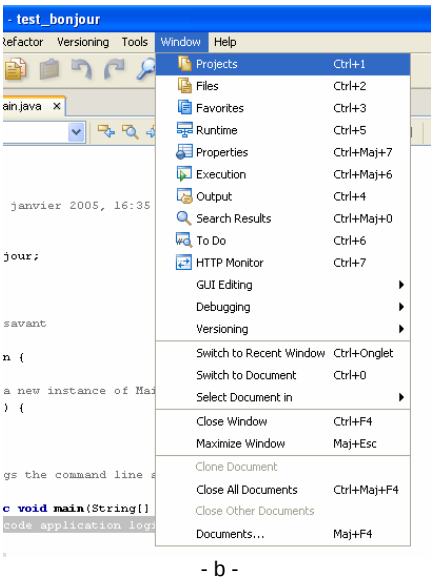


Figure 3

Puis cliquez sur le bouton Finish. Le projet est créé (voir Figure 4-a) ainsi que la classe main classe_bonjour. Si vous ne voyez pas les projets comme dans la Figure 4-a alors cliquez dans le menu sur Window puis sur Projects, voir Figure 4-b.



- a -



- b -

Figure 4

Question 1 : Que signifie un projet créé, au niveau du disque-dur ?
Exercice 1 : Comment peut on connaître le chemin où se trouve le projet ?

Étape 2 : Création de la classe principale

La classe principale avec netBeans peut être créée au même temps que le projet. Nous l'avons déjà créée dans l'étape 1. Si cette classe n'est pas déjà créée alors suivez les mêmes instructions pour créer un package (voir l'étape 5).

Un package est un répertoire créé dans votre projet, et la Figure 5 montre bien que la classe classe_bonjour se trouve dans un package nommé (test_bonjour) ; vérifiez dans votre disque si un tel répertoire a été créé.

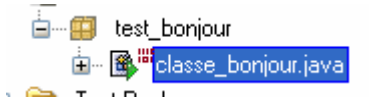


Figure 5

Question 2 : Un tel répertoire existe-il ou non ? Où est ce que se trouve donc le fichier source classe_bonjour.java dans votre disque ? Conclusion ?

Étape 3 : Écriture du contenu de la classe classe_bonjour

Le contenu du fichier classe_bonjour.jar est affiché dans une fenêtre texte (voir Figure 6).

```
/*
 * classe_bonjour.java
 *
 * Created on 16 janvier 2005, 18:14
 */
package test_bonjour;
/**
 *
 * @author  ordisavant
 */
public class classe_bonjour {
    /** Creates a new instance of classe_bonjour */
    public classe_bonjour() {
    }
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
    }
}
```

Figure 6

Nous allons donc modifier le contenu du code source classe_bonjour.java en ajoutant un code permettant d'afficher "Bonjour" à l'intérieur de la méthode main (voir Figure 7).

```
public static void main(String[] args) {
     System.out.println("Bonjour");
}
```

Figure 7

Pour compiler ce code, cliquez dans le menu sur Build puis sur Build main Project (voir Figure 8). Si votre projet n'est pas le projet principal (main project) alors cliquez sur le projet avec le bouton droit de la souris puis cliquez sur Set Main Project puis compilez comme c'est décrit précédemment. Et pour l'exécuter soit cliquez sur le bouton Run Main Project (voir Figure 9) soit en cliquant dans le menu sur Run puis sur Run Auther puis sur Test "test_bonjour" (voir Figure 10).

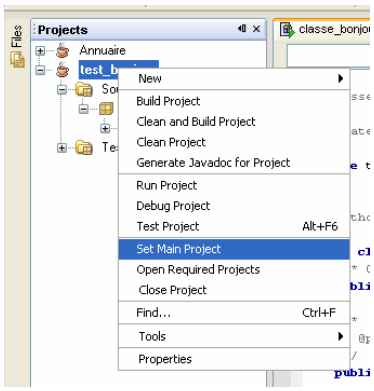


Figure 8

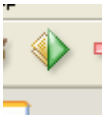


Figure 9

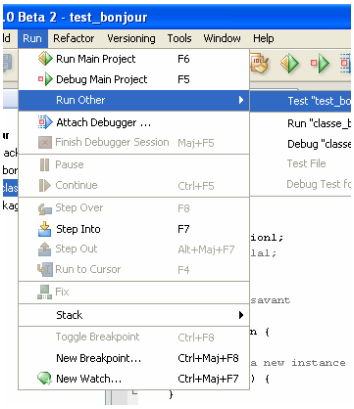


Figure 10

Question 3 : Observez le résultat de l'exécution puis donnez le nom de la fenêtre qui affiche le résultat et ainsi que le résultat.

Étape 4 : La classe classe_bonjour sous forme d'une applet

Exercice 2 : Comment créer une applet qui affiche Bonjour avec netBeans ?
Question 4 : Le résultat est il le même que celui d'une application ? Où est ce que le mot "Bonjour" est affiché ?
Exercice 3 : Créez une page-web qui utilise cette applet.

Étape 5 : Création et utilisation d'un package

Pour créer un package, cliquez sur le projet avec le bouton droit puis sur nouveau puis sur Java Package… (voir Figure 12).

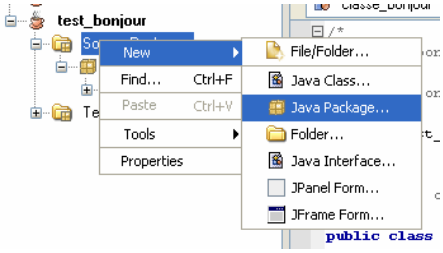


Figure 12

Une fenêtre s'affiche.

Question 5 : Faite entrer, dans cette fenêtre, comme nom le mot "package", est il possible ? Sinon, entrer le nom "package1"
Question 6 : Pourquoi le mot "package" ne peut pas être un nom de notre package ?

Le résultat est présenté dans la Figure 13.

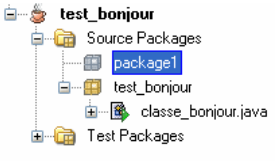


Figure 13

Créez une classe appelée "classe1" dans le package "package1".

Question 7 : Quelles sont les démarches à suivre pour créer cette classe ?

Créez dans la classe classe1 un attribut appelé "s" de type String et un constructeur qui initialise la variable s à la valeur "Salut" et une méthode appelée valeur_de_s qui renvoie la valeur de s.

Question 8 : Que faut-il écrire ?

Modifiez le code de la classe main en affichant à la place de "bonjour" la valeur de l'attribut s d'un objet de type classe1.

Question 9 : Que faut-il écrire ? Exécutez, et dire ce qui est affiché.

Vous avez sûrement ajouté la commande **`import package1.classe1`**, sinon erreur.

Étape 6 : CLASSPATH

1 – Appel d'une classe dans un package local

Question 10 : La commande **`import package1.classe1`** est elle toujours nécessaire ?
Question 11 : Enlevez cette commande (ou ligne) puis exécutez, c'est quoi le résultat ?

2 – Appel d'une classe dans un **JAR**

Créez un répertoire dans votre disque et appelez le mes_jars, puis télécharger le fichier classe_ext.jar sur l'adresse suivante : http://, et finalement placez ce fichier dans le répertoire mes_jars que vous venez de créer.

Question 12 : Que faut-il faire pour pouvoir utiliser les différentes classes de ce JAR ? Quelles sont alors les classes que contient ce JAR (sans ouvrir le fichier JAR) ?
Question 13 : Ce JAR contient une classe appelée classe_plus, quels sont les attributs de cette classe ? Ils sont de quels types ?
Question 14 : Quelles sont les différents constructeurs et les différentes méthodes de cette classe ? Les méthodes sont de quels types ?
Question 15 : La classe classe_plus contient uniquement un seul attribut de type String. Utilisez votre applet pour afficher la valeur de cet attribut. Quelle est donc la valeur de cet attribut ?

3 – Créer un JAR

Exercice 4 : Décrivez toutes les étapes à suivre pour créer un JAR.
Exercice 5 : Créez un JAR qui contiendra tout le projet que vous venez de créer. Puis créez un nouveau projet qu'il faut appeler projet_jar. Décrivez les démarches à suivre pour exécuter le code qui se trouve dans ce JAR.

Étape 8 : Le Javadoc

Il existe deux types de commentaires dans les codes Java. Le premier type représente les commentaires classiques qui sont sous la forme suivante :

```
/*
Ligne 1 de mon commentaire,
Ligne 2 de mon commentaire,
...
*/
```

Ce type de commentaire est utile pour le programmeur. Le deuxième type a le même principe que le premier mais il est utile pour la génération du Javadoc. Il s'écrit comme suit :

```
/**
Ligne 1 de mon commentaire,
Ligne 2 de mon commentaire,
...
*/
```

Chaque commentaire de type 2 doit définir la méthode le suivant.
Ajouter à votre programme tous les commentaires de type 2 définissant toutes les méthodes et attributs.

Question 16 : Comment générer un Javadoc ?

Étape 7 : Introduction à l'Orientée Objet

La valeur de l'attribut s de la classe classe_plus qui se trouve dans le fichier classe_ext.jar est initialisée dans le constructeur de celle-ci. Supposons que nous voulons afficher un autre mot, d'une autre manière, nous voulons modifier la valeur de s

Question 17 : Est ce cela est possible ? Si je vous dis que la réponse est oui, quel est le principe de l'Orienté Objet qui nous permet d'effectuer une telle modification ? Comment s est-il, donc, déclaré dans la classe classe_plus pour qu'on puisse effectuer cette opération ?

Exercice 6 : Écrivez le code de la classe classe_her qui nous permettra de modifier la valeur de s de la classe classe_plus à travers la méthode changer_s(String v).

Bon TP