

TP1 : Environnement de développement NetBeans

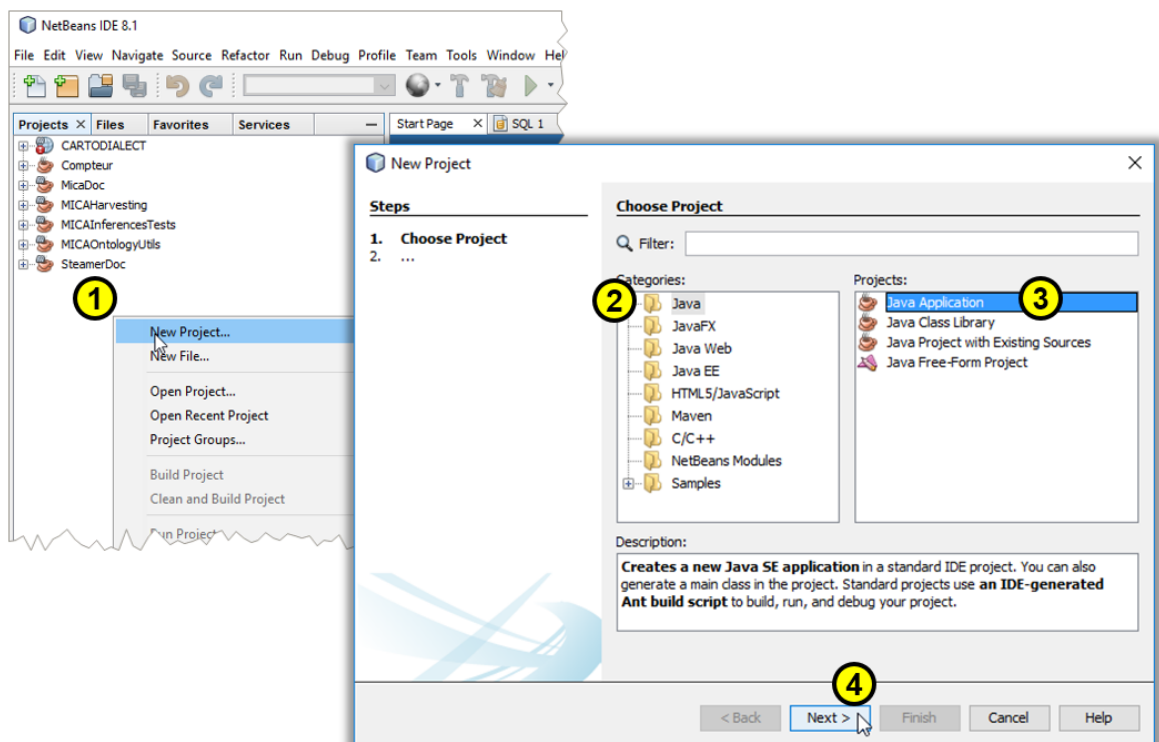
1. Introduction

NetBeans est facile à prendre en main et **gratuit**.

2. Création d'une application Java avec NetBeans

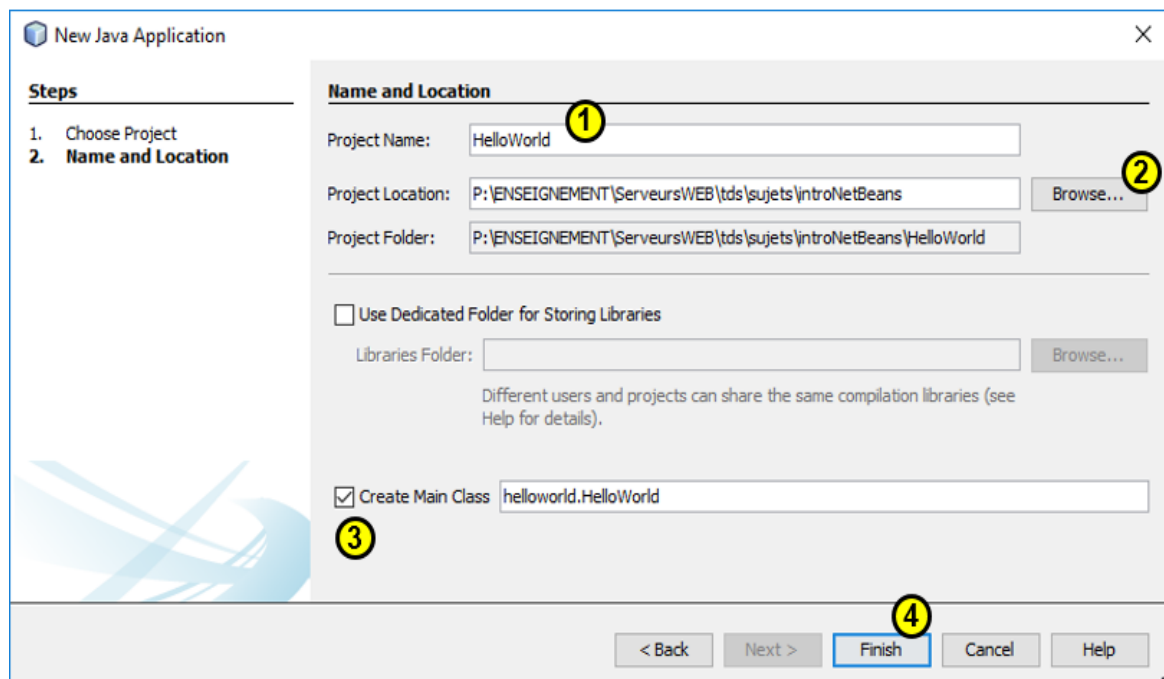
Pour créer un nouveau projet d'application JAVA :

1. choisissez **File > New Project** (Maj-Ctrl-N). Vous pouvez aussi faire directement un clic droit dans la vue **Projects**
2. dans la boîte de dialogue pour la création de nouveaux projets qui apparaît sélectionnez Java sous la rubrique Categories,
3. sous la rubrique Projects, sélectionnez Java Application
4. cliquez sur Next.



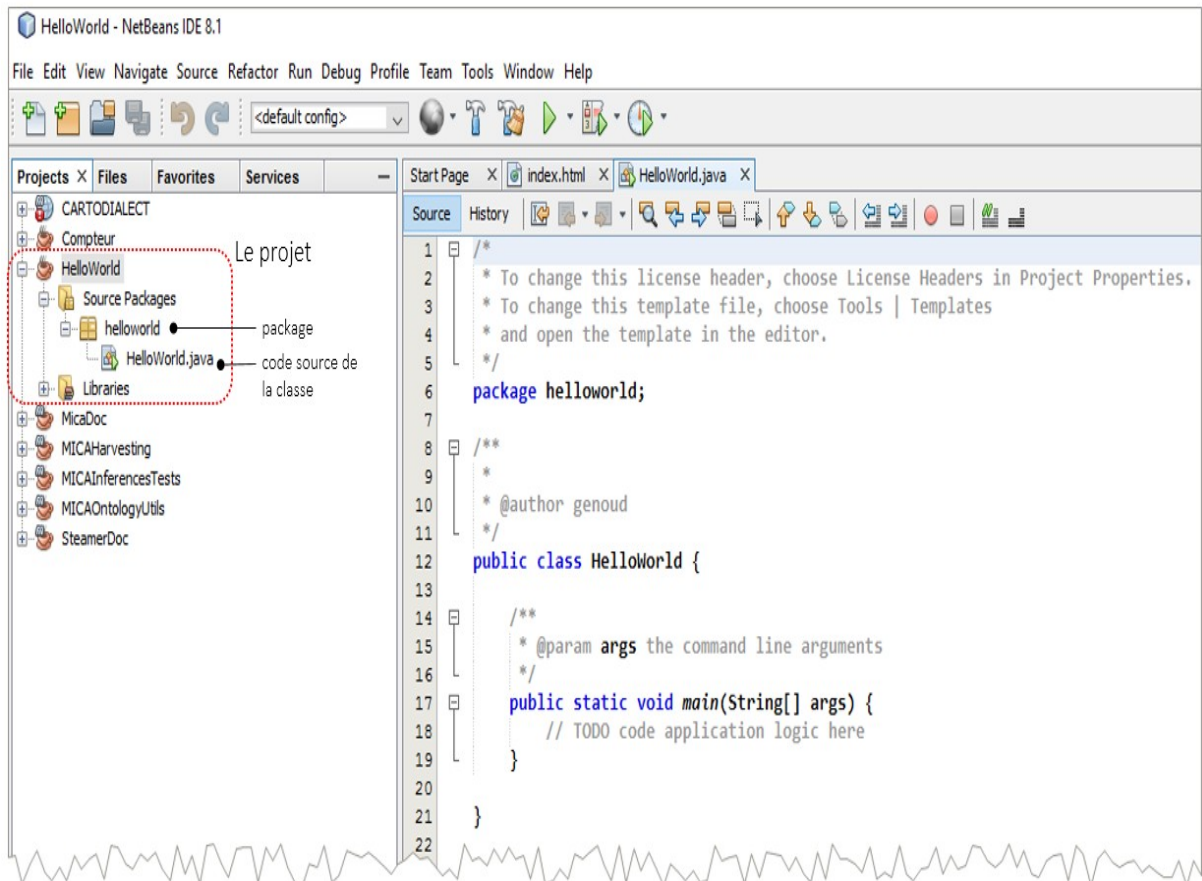
Une boîte de dialogue pour la création d'une nouvelle application Java (JSE) est alors affichée.

1. Sous **Project Name**, introduisez HelloWorld.
2. Avec **Project Location** choisissez un emplacement pour les fichiers de votre projet.
3. Demandez à Netbeans de créer automatique une classe contenant un programme principal (méthode **public static void main(String[] args)**). Par défaut Netbeans vous propose de créer une classe HelloWorld (de même nom que votre projet) dans un package helloworld (même nom que le project mais en minuscules).
4. Cliquez sur **Finish**.




NetBeans crée un dossier HelloWorld dans le répertoire que vous avez choisi pour votre projet (dans l'exemple ci-dessus P:\ENSEIGNEMENT\ServeursWEB\tds\sujets\introNetBeans\HelloWorld).

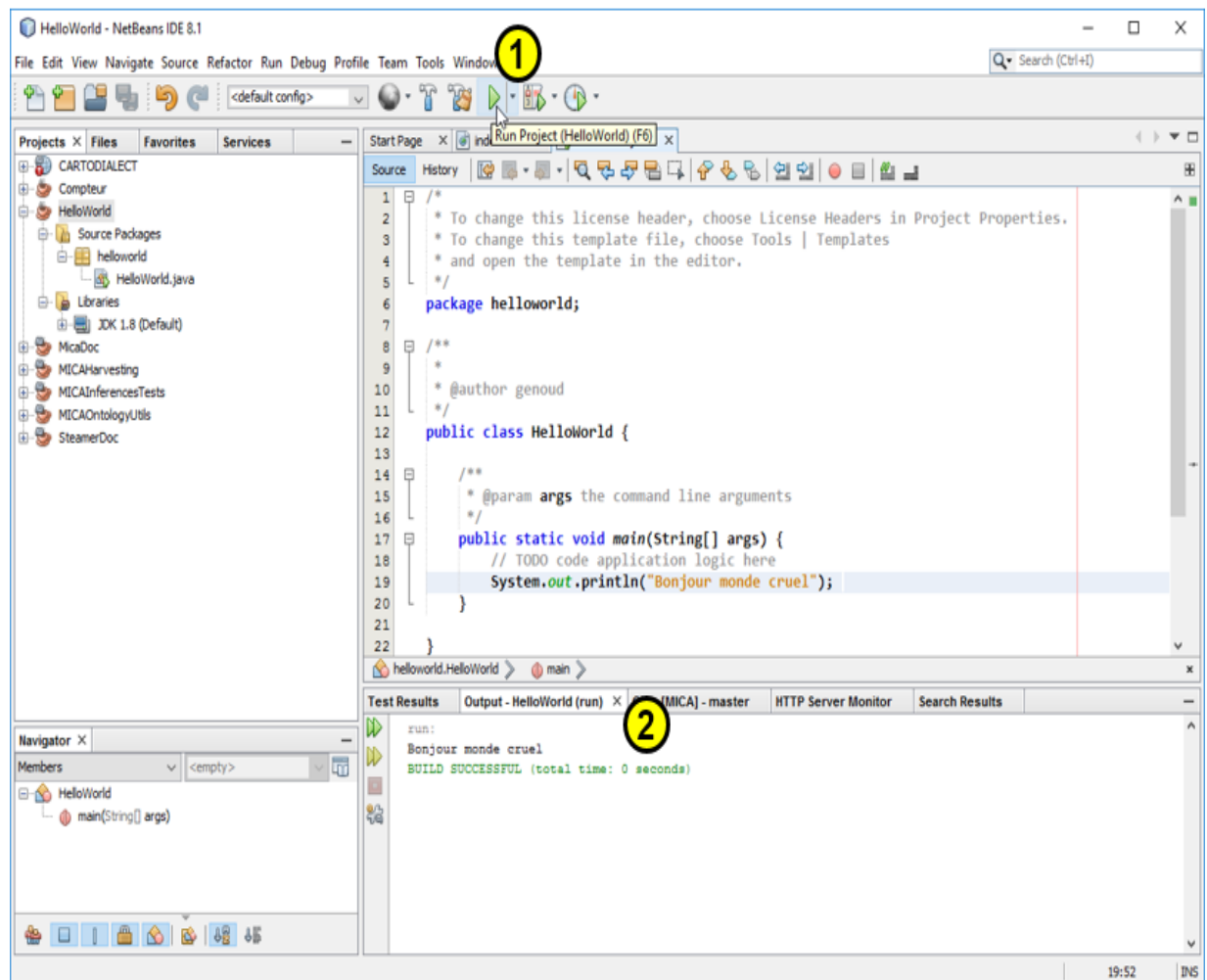
Le projet HelloWorld s'ouvre et la classe HelloWorld.java est affichée dans l'éditeur de sources.



- L'onglet **Projects** permet de visualiser la structure logique du projet (les différents packages Java qui le constituent), l'onglet **Files** permet de visualiser sa structure de fichiers.

3. Compiler et exécuter le projet

Une fois le fichier sauvegardé vous pouvez lancer la compilation et l'exécution de votre projet par **Menu Run-> Project (F6)** ou en cliquant sur le bouton  de la barre d'outils. Cela a permet de compiler puis exécuter votre application. Les sorties de ces commandes sont affichées sur la fenêtre **Output** en bas de la fenêtre de NetBeans.



Exemples à exécuter

Rappel

- Un programme Java est constitué de classes (tout programme Java se fonde sur l'utilisation des **classes**)
- Les classes ont un nom qui commence par une majuscule
- Une seule classe contient une méthode main : c'est la fonction qui sert de programme principal (point de départ de l'exécution)
- Un fichier source contenant une application java doit nécessairement avoir comme nom le nom d'une classe existant dans le fichier.

Cette classe est la seule classe publique et ayant le même nom que le fichier. Et elle contient la méthode **main()**

- On peut ranger les classes selon des ensembles : les **paquetages**.
- Il y a des règles de visibilité (public, protected, private...) entre les classes et entre les attributs et les méthodes qu'elles contiennent. Ces règles dépendent de l'appartenance ou non à un même paquetage.

Exemple 1

```
package aaaa;

public class Aaaaa {

    public static void main(String[] args) {

        System.out.println("BONJOUR");

    }

}
```

Exemple 2

```
public static void main(String [] args) {

    System.out.println("BONJOUR");

    for (int i = 0; i < args.length; i++) {

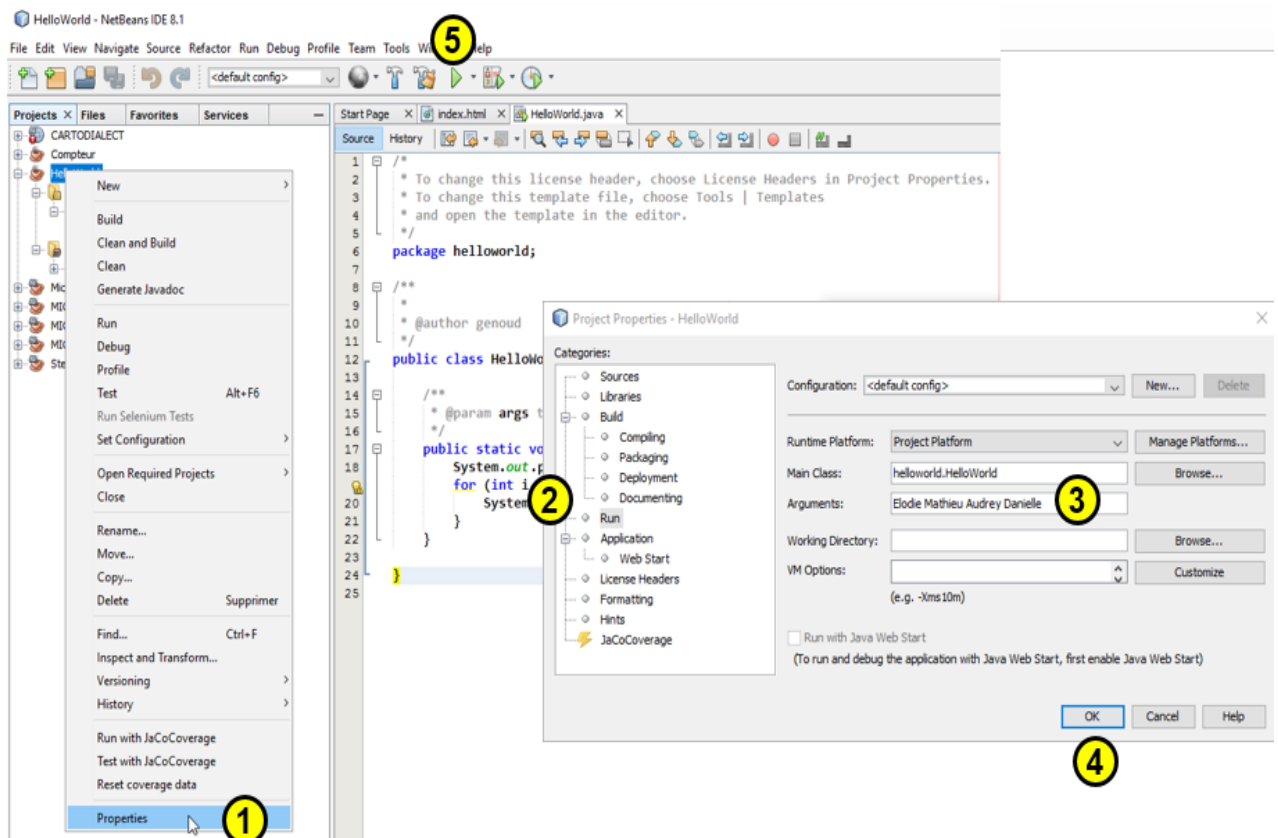
        System.out.println("Bonjour " + args[i]);

    }

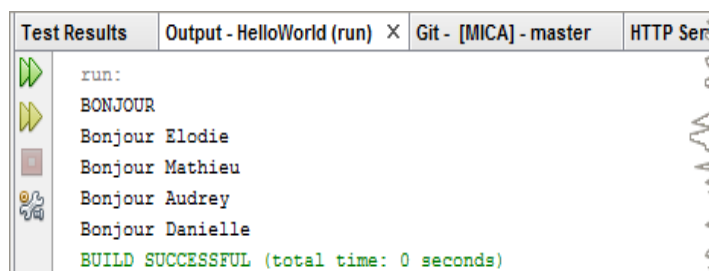
}
```

Pour lancer ce programme depuis NetBeans en lui communiquant une suite d'arguments il vous faut configurer les propriétés de votre projet.

1. accédez au panneau de configuration de projet en effectuant un clic droit sur le projet et en choisissant l'item **Properties** dans le menu contextuel.
2. une fenêtre d'édition des propriétés du projet HelloWorld apparaît.
Sélectionnez la catégorie **Run**.
3. dans le champ de saisie Arguments introduisez une suite de noms séparé par des espaces
4. validez par **OK**,
5. relancez l'exécution du projet (Run -> Run Main Project ou F6).



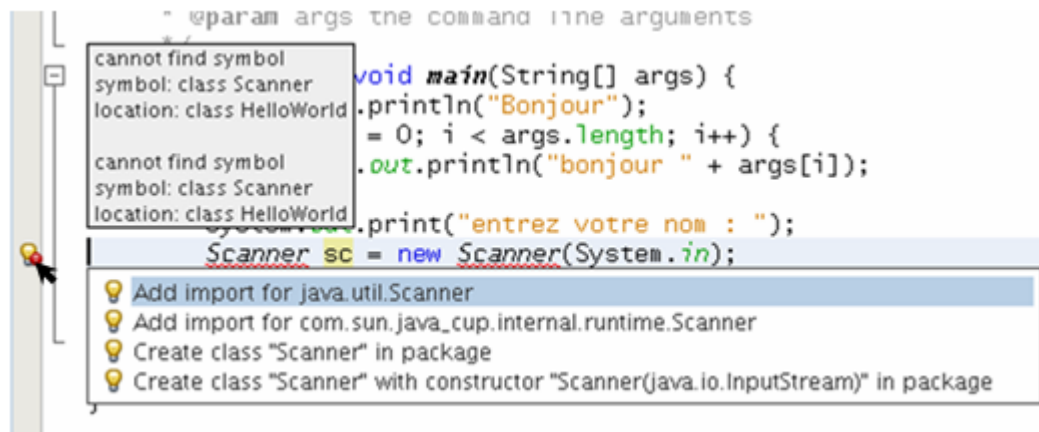
Dans la fenêtre de sortie vous devez obtenir l'affichage suivant :



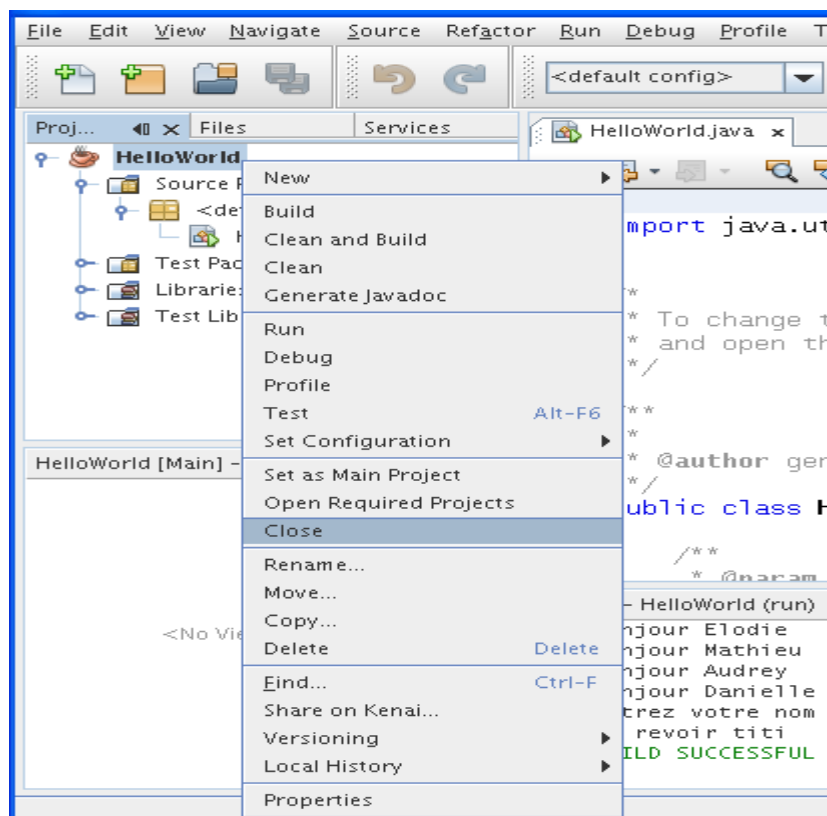
Complétez ensuite le main de votre classe HelloWorld avec le code suivant :

```
public static void main(String[] args) {
    System.out.println("BONJOUR");
    for (int i = 0; i < args.length; i++) {
        System.out.println("Bonjour " + args[i]);
    }
    System.out.print("entrez votre nom : ");
    Scanner sc = new Scanner(System.in);
    String nom = sc.next();
    System.out.println("Au revoir " + nom);
}
```

La classe Scanner est une classe du package **java.util**, pour pouvoir l'utiliser dans votre code il est nécessaire d'effectuer un import au début du programme. Pour cela, vous pouvez cliquer avec la souris sur l'icone qui apparait en face de la ligne où cette classe est utilisée. NetBeans vous propose alors des suggestions pour "fixer" cette erreur. Sélectionnez l'option **Add import for java.util.Scanner**.



Lancez l'exécution de votre application (**Run -> Run Main Project** ou **F6**).

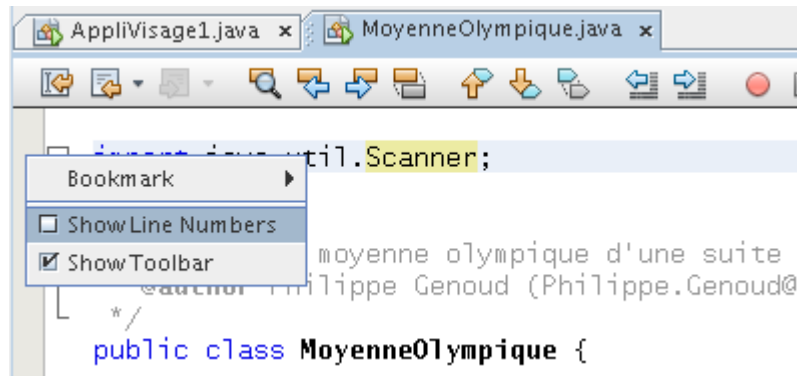


4. Interrompre l'exécution d'un programme

A l'aide de **Ctrl+C**

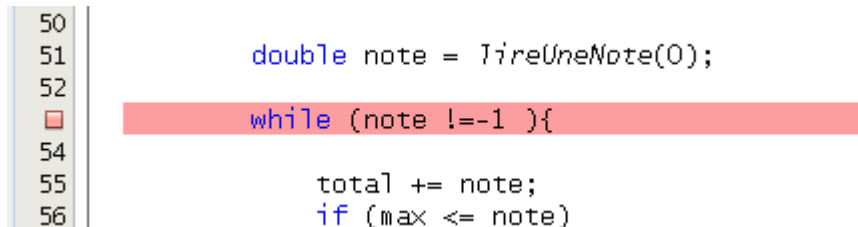
5. Utiliser le debugger de NetBeans

- a. Faites apparaître les numéros de lignes dans la fenêtre de l'éditeur. Pour cela effectuer un clic droit dans la colonne de gauche de la fenêtre d'édition et cochez la case **Show Line Numbers**.



- b. Placez un point d'arrêt à la ligne 30 par exemple du programme (la boucle "while"). Pour cela cliquer dans la colonne de gauche de l'éditeur de texte au niveau de cette ligne.

Une fois ce point d'arrêt fixé il apparaît sous la forme d'un carré rose au début de la ligne. (Pour le supprimer il suffit de cliquer dessus).

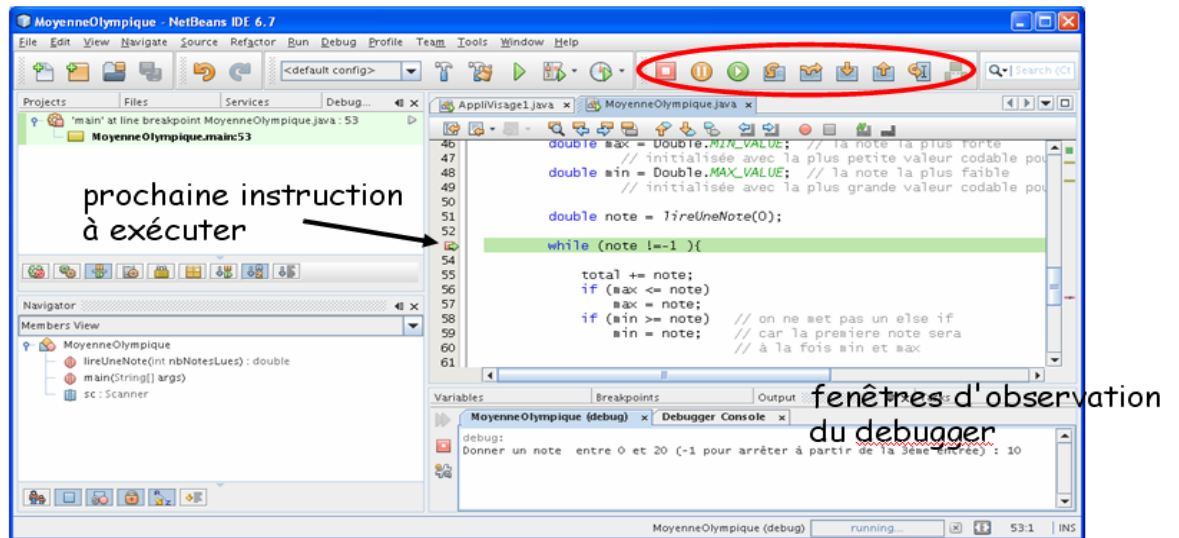


- c. Lancez ensuite une exécution sous le contrôle du debugger. Pour cela cliquez sur le bouton **Debug** la barre d'outils de NetBeans.



Dans la console d'exécution, saisissez la première. Après cette lecture l'exécution s'interrompt sur le point d'arrêt fixé précédemment. Votre espace de travail est alors organisé de la manière suivante.

barre d'outils du debugger



d. L'exécution est interrompue à la ligne 30 (qui n'a pas encore été exécutée).

Pour faire avancer l'exécution pas à pas cliquez sur le bouton "Step Over".



Répétez cette opération plusieurs fois et observez l'état des variables du programme. Pour cela faite en sorte que la fenêtre d'observation Variables sont au premier plan.

Variables	Breakpoints	Output
Name	Type	Value
<Enter new watch>		
Static		
args	String[]	#66(length=3)
sc	Scanner	#253
nom	String	"xxxxxxxxxx" #66(length=3)

e. On peut utiliser d'autres outils comme :

- Lorsque on peut avancer l'exécution en rentrant dans une méthode en


cliquant sur le bouton "Step Into"



- Pour aller directement jusqu'au prochain point d'arrêt cliquez sur le

bouton "Continue".



- f. Continuez l'exécution sous le contrôle du debugger jusqu'à ce que vous ayez compris la cause de l'erreur. Lorsque vous voulez interrompre l'exécution du debugger cliquez sur le bouton **Finish Debugger Session** 

3.6. Documentation en ligne de NetBeans

Pour profiter aux mieux des très riches fonctionnalités de l'environnement de java sous NetBeans accédez au documentation en ligne. Pour obtenir cette aide, menu **Help-->Help Contents**.

