

Manuel d'utilisation de l'IDE Netbeans

Notion de projet NetBeans

Avec NetBeans (<http://www.netbeans.org>) vous travaillez toujours au sein d'un projet. Un projet est un ensemble de fichiers source JAVA et des informations associées (classpath, comment construire et exécuter le projet, etc...). L'IDE NetBeans stocke l'information associée à un projet dans un dossier projet (project folder) qui inclus un script build Ant (fichier `build.xml` équivalent d'un `Makefile`, Ant est un projet open source de la fondation Apache (<http://ant.apache.org/>) qui contrôle la compilation et l'exécution, et un répertoire `nbproject` qui contient différents fichiers de configuration de votre projet.

Avec NetBeans vous pouvez soit

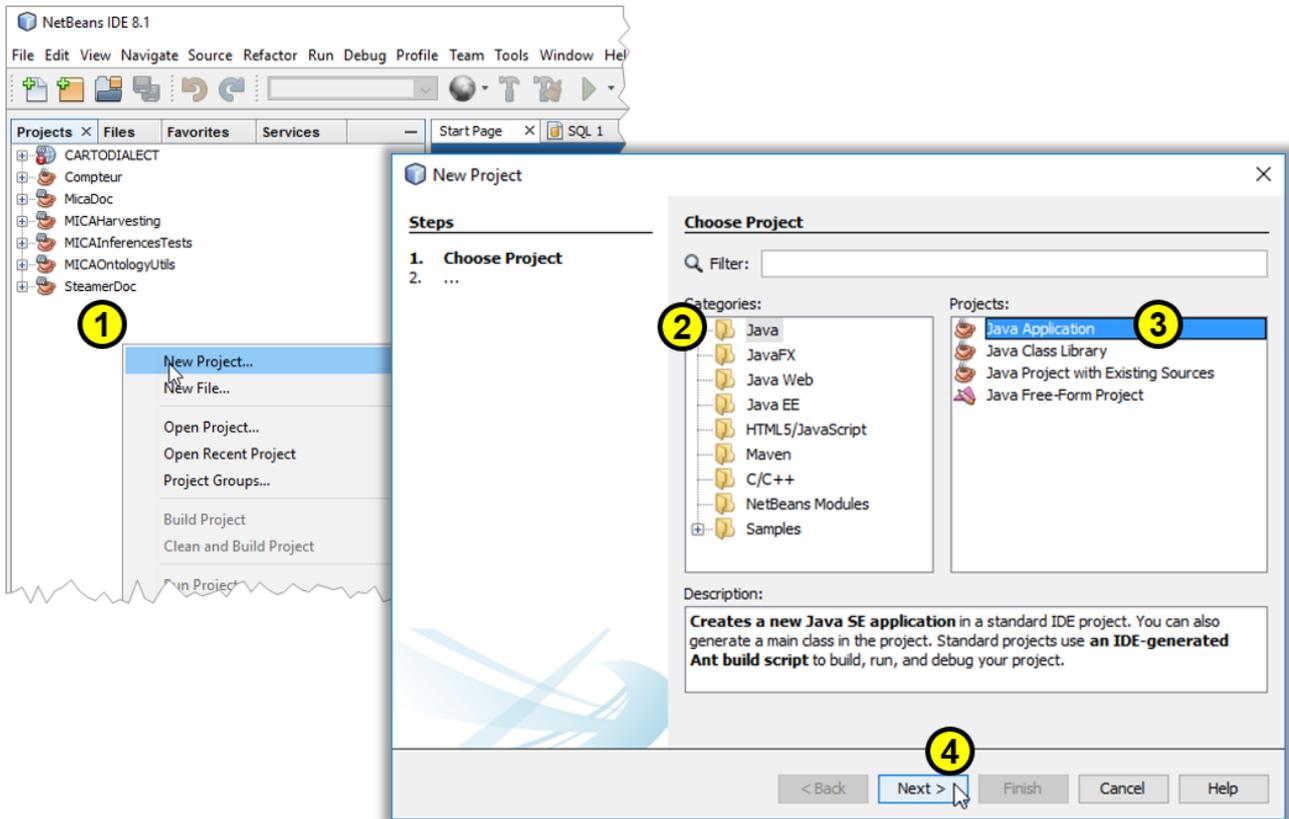
- créer des projets standards qui s'appuient sur des scripts Ant générés par NetBeans et sur une organisation standard des fichiers du projet.
- créer vos projets avec vos propre scripts et votre propre organisation de fichiers.

Dans cette introduction à NetBeans nous ne nous intéresserons qu'aux projets Standards, et particulièrement aux projets de type Application Java.

Création d'une application Java avec NetBeans

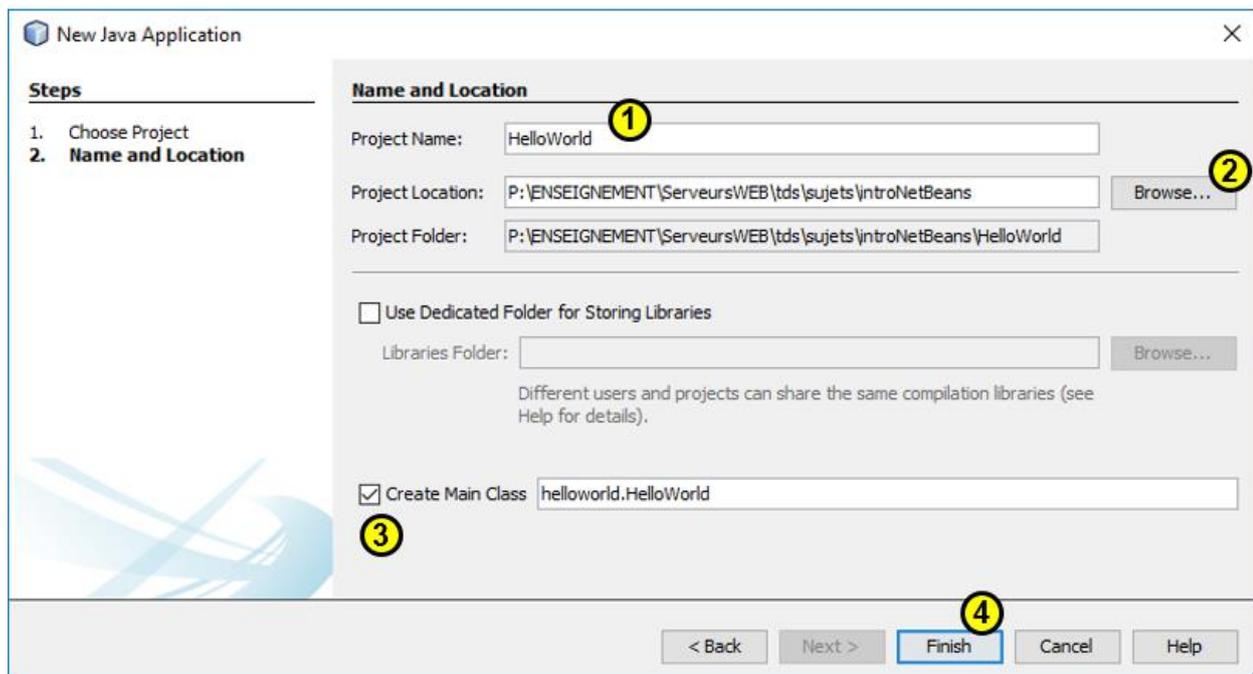
Pour créer un nouveau projet d'application JAVA :

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**.

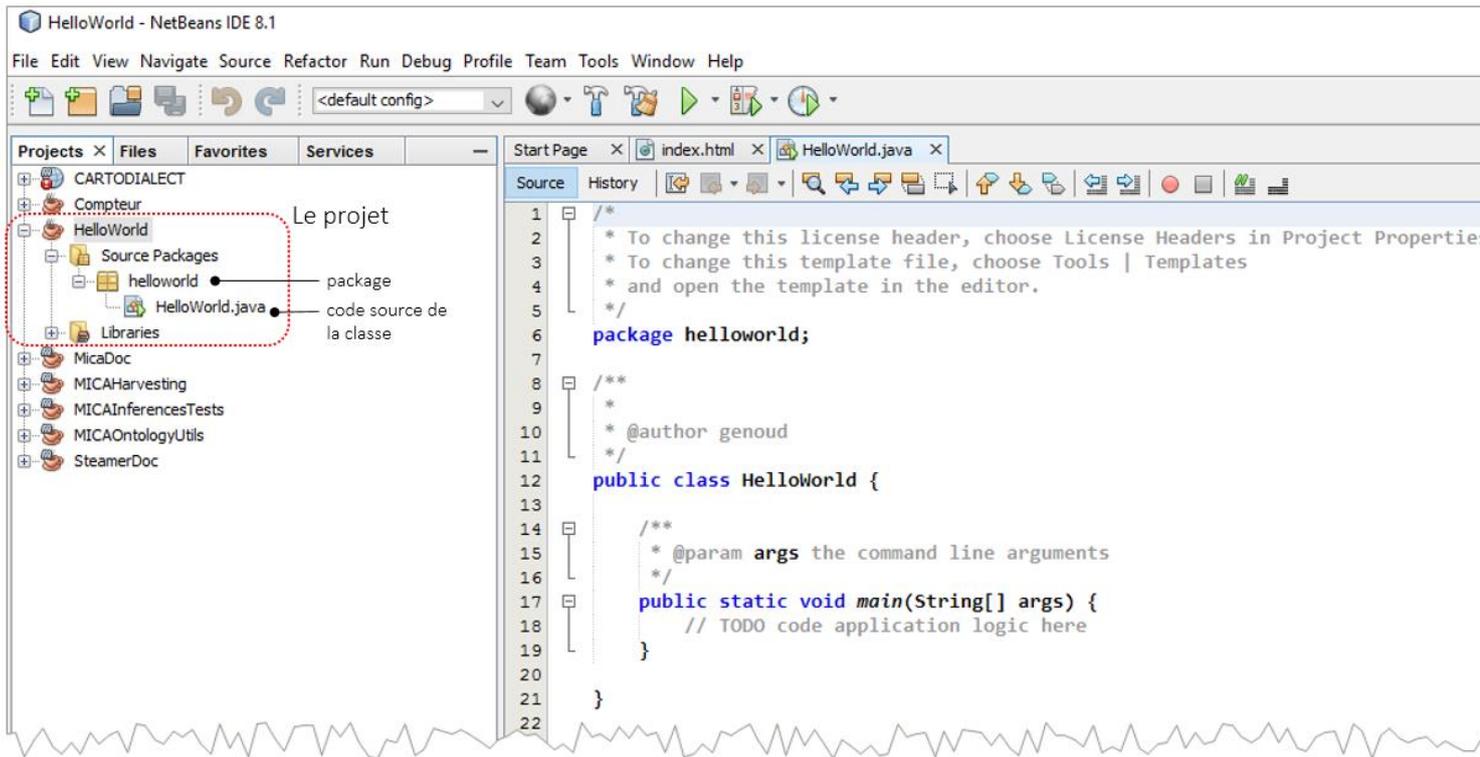


Une boîte de dialogue pour la création d'une nouvelle application Java (JSE) est alors affichée.

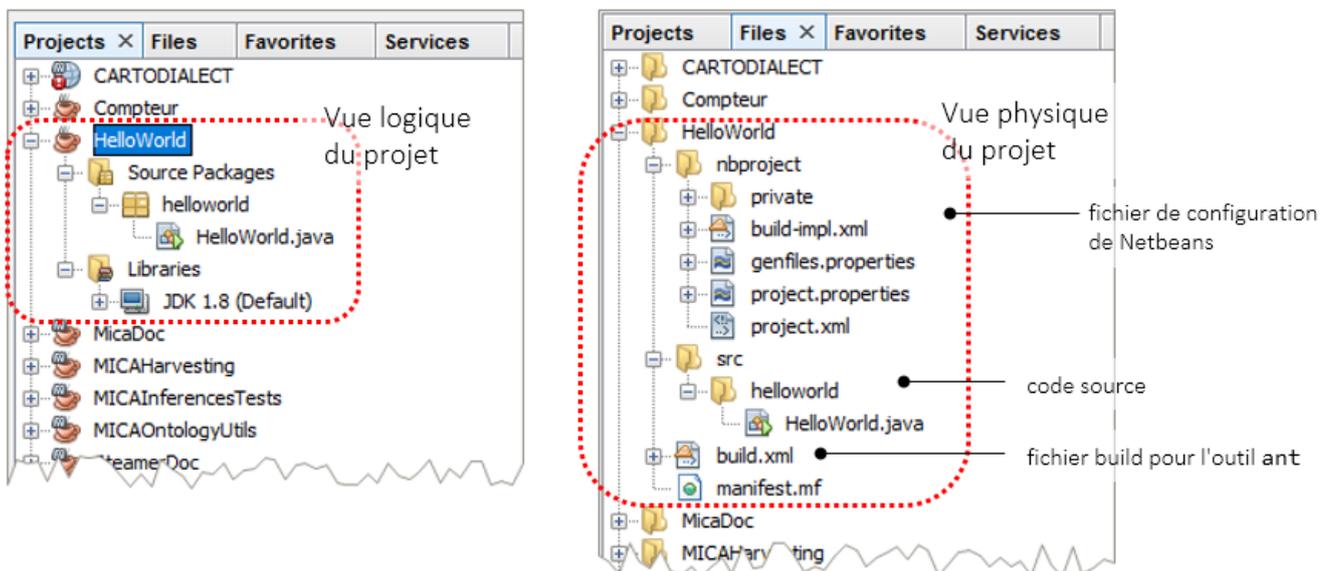
1. Sous `Project Name`, introduisez `HelloWorld`.
2. Avec `Project Location` choisissez un emplacement pour les fichiers de votre projet.
3. Demandez à Netbeans de créer automatique une classe contenant un programme principal (méthode `public static void main(String[] args)`).
Par défaut Netbeans vous propose de créer un 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.



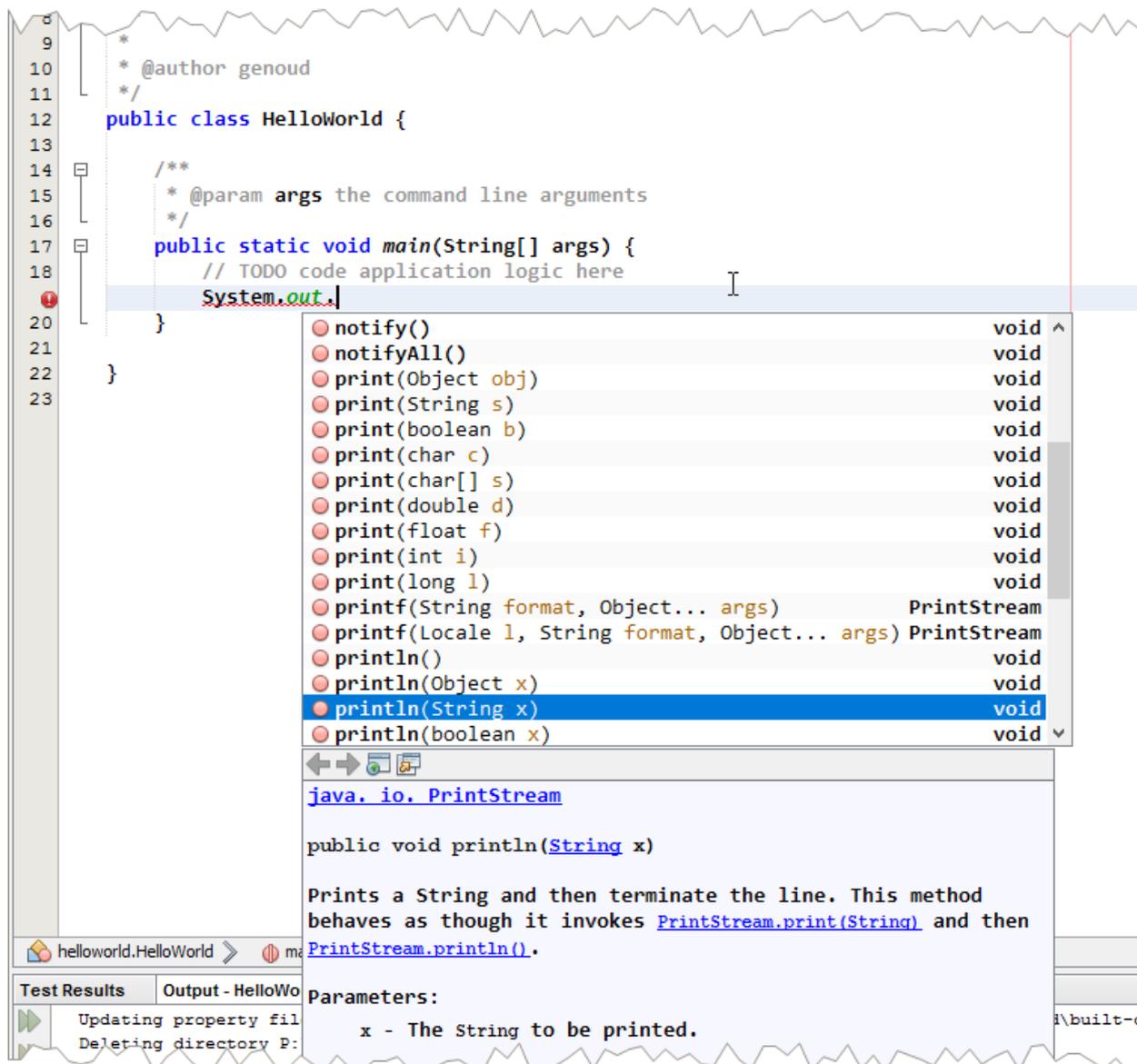
Vue logique d'un projet

Utilisation de l'éditeur.

Modifiez le programme HelloWorld en complétant la méthode main avec l'instruction

```
System.out.println("Bonjour Monde cruel!");
```

Lorsqu'il le peut, l'éditeur vous propose une complétion automatique du code. Par exemple arrêtez la frappe après avoir tapé `System.out.`, l'éditeur vous propose alors toutes les méthodes possibles avec la documentation javadoc associée.



L'éditeur offre la complétion de code pour toutes les classes et méthodes dans le classpath de compilation du projet. Vous pouvez à tout moment activer la complétion par `Ctrl-Espace`. L'éditeur n'attend pas toujours qu'une compilation ait été lancée pour vous signaler d'éventuelles erreurs. Au fur et à mesure que vous tapez votre texte, la syntaxe java est vérifiée et en cas d'erreur un petit point d'exclamation rouge est positionné en face de la ligne correspondante. En positionnant le curseur de la souris sur celui-ci vous aurez un message d'explication indiquant la nature de l'erreur. Pour apprécier cette fonctionnalité ne mettez pas de `;` à la fin de votre instruction `System.out.println.`

```

14  /**
15  * @param args the command line arguments
16  * expected
17  * -----
18  * (Alt-Enter shows hints)
19  * code application logic here
20  System.out.println("Bonjour monde cruel");
21  }
22  }

```

Corrigez l'erreur de syntaxe et sauvegardez le fichier HelloWorld.java (File->Save ou Ctrl-S). Si le fichier a été modifié et n'a pas été sauvegardé son nom apparaît en gras dans l'onglet correspondant.



Fichier non sauvegardé



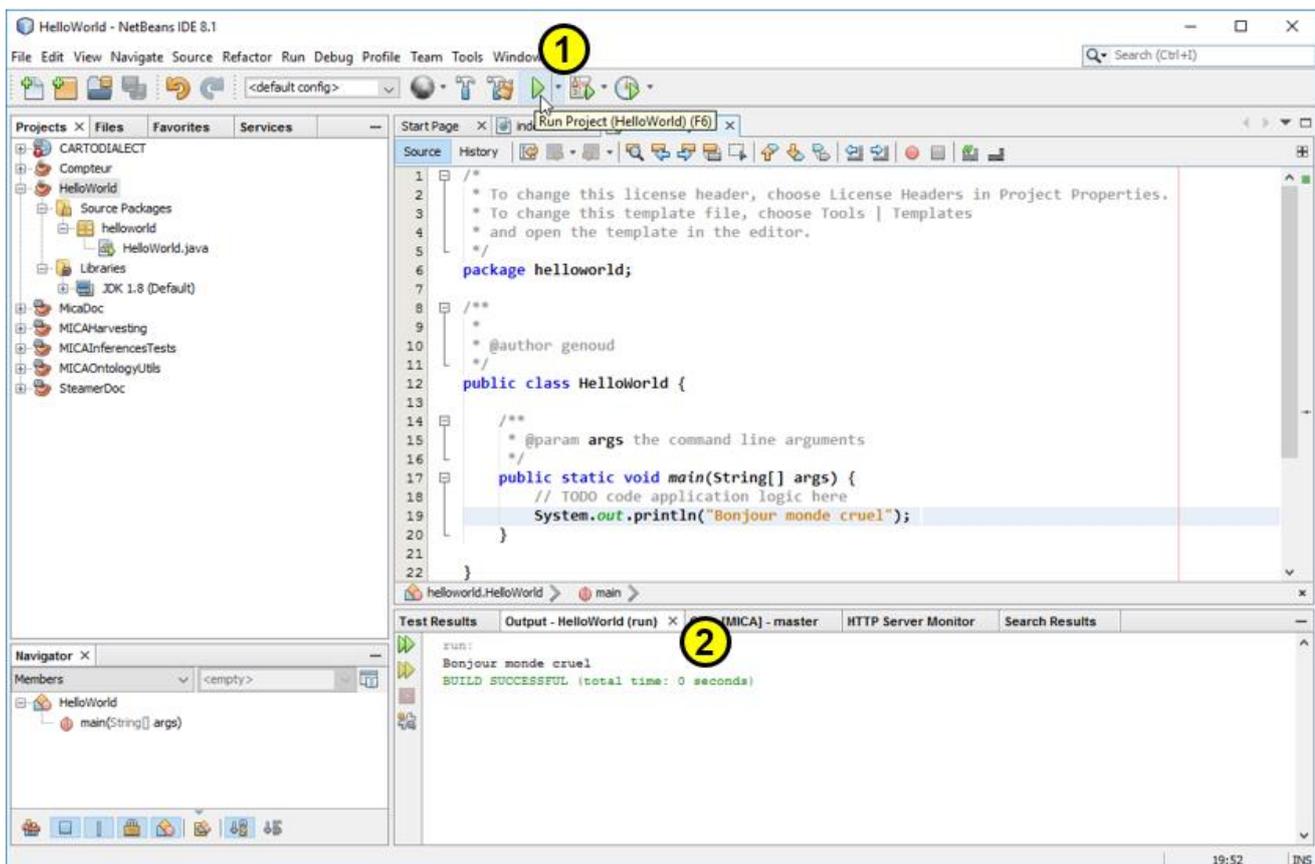
Fichier sauvegardé

Fichier non sauvegardé

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 pour effet d'exécuter les différentes commandes du script *Ant* build.xml pour compiler puis exécuter votre application. Les sorties de ces commandes *Ant* puis de votre application sont affichées sur la fenêtre Output en bas de la fenêtre de NetBeans.

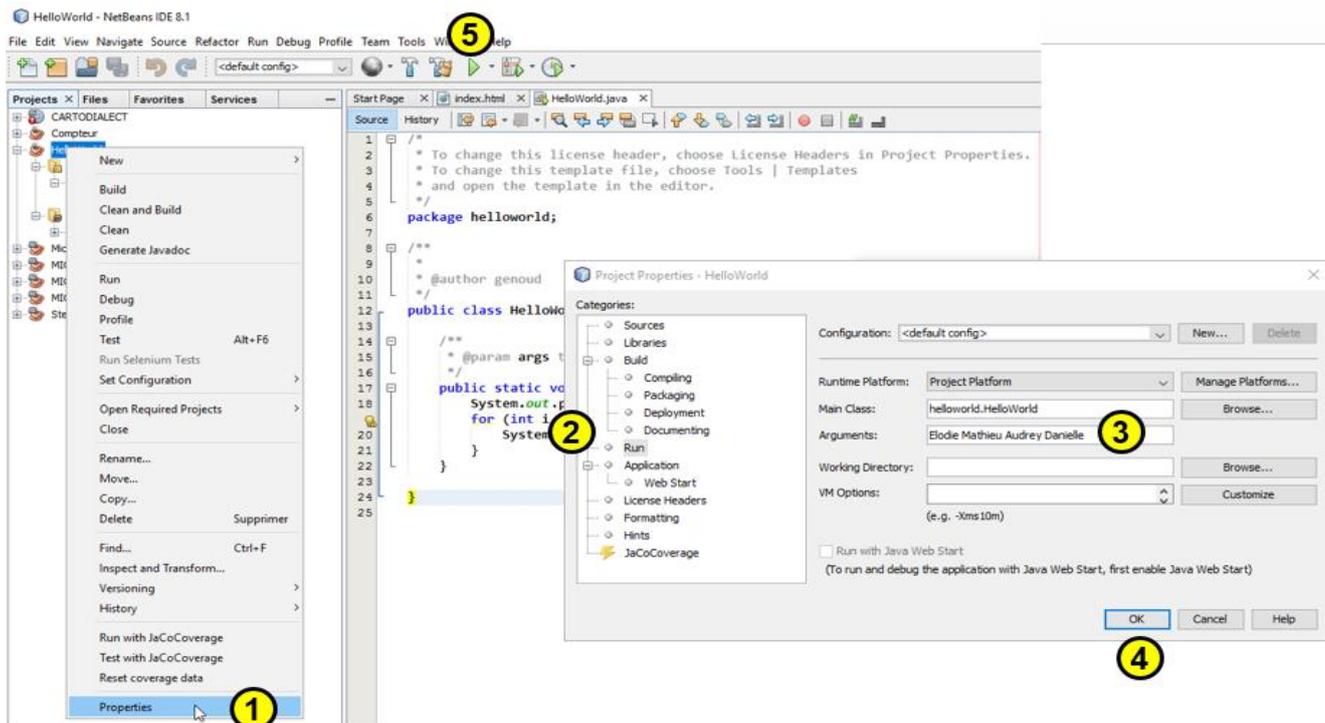


Nous allons maintenant compléter le programme HelloWorld afin qu'il puisse, comme nous l'avons déjà vu lors du TP n°1, lire une suite de noms sur la ligne de commandes et pour chaque nom lu afficher une chaîne Bonjour ...

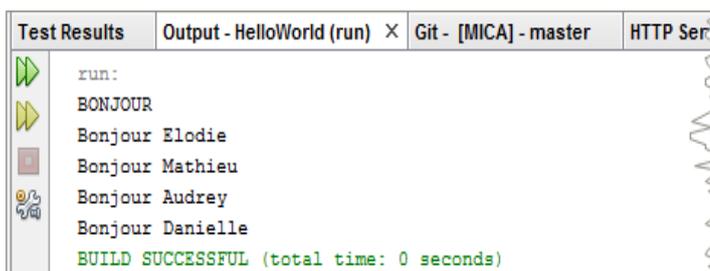
```
public static void main(String[] args) {
    System.out.println("BONJOUR");
    for (int i = 0; i < args.length; i++) {
        System.out.println("Bonjour " + args[i]);
    }
}
```

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 :



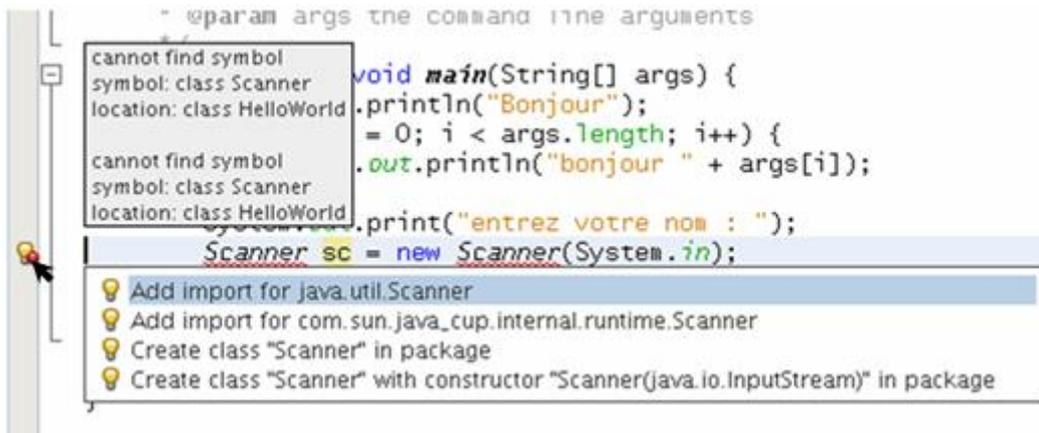
Complétez ensuite le main de votre classe HelloWorld avec le code suivant :

```

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);
}

```

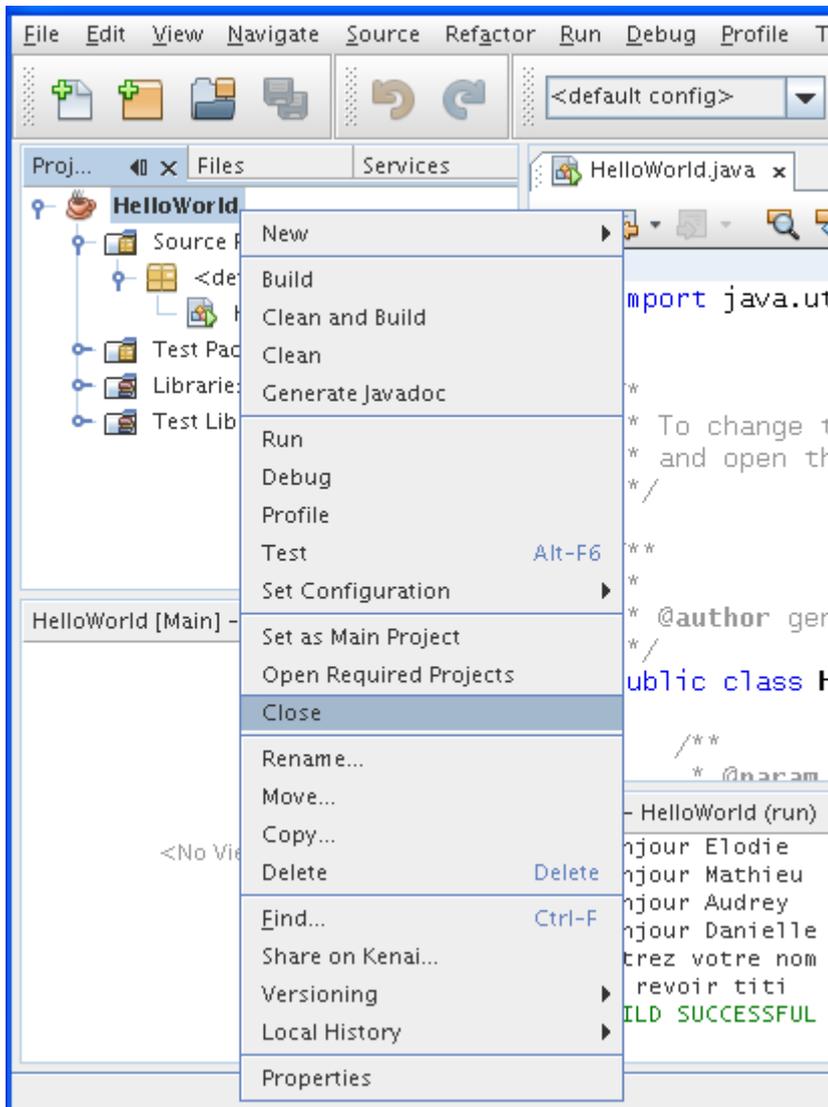
La classe `Scanner` est une classe du package `java.util`, pour pouvoir l'utiliser dans votre code il est nécessaire d'effectuer un import au début du programme. Pour cela, vous pouvez cliquer avec la souris sur l'icone qui apparait en face de la ligne où cette classe est utilisée. NetBeans vous propose 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).

Interrompre l'exécution d'un programme

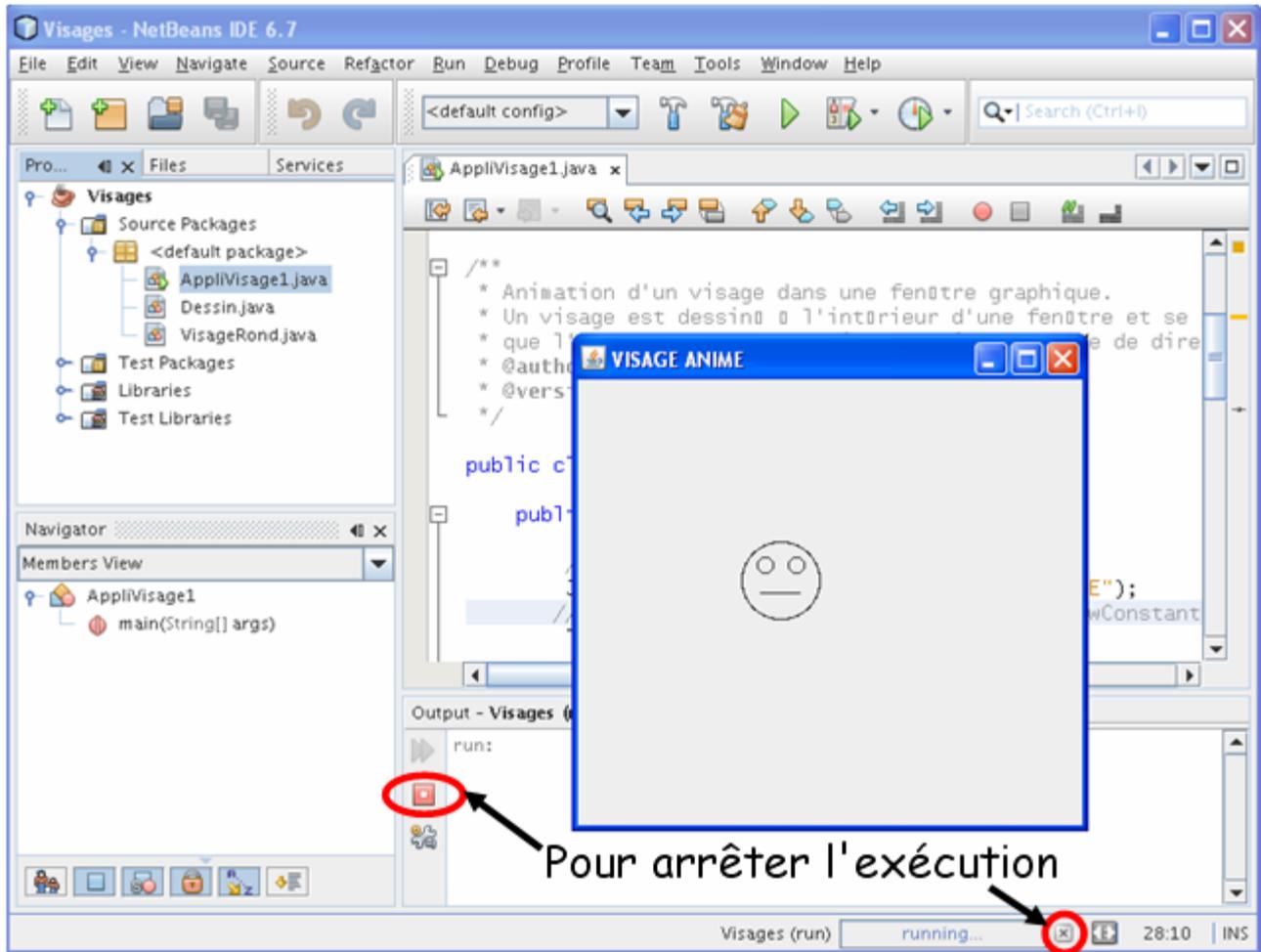
Fermez le projet `HelloWorld` (clic droit dans la fenêtre projet et item `Close Project` du menu contextuel) puis créez un nouveau projet `VisagesRond`.



Recopiez dans le répertoire `src` de ce nouveau projet les sources classes de l'application `AppliVisage` vue lors du TP n° 1 :

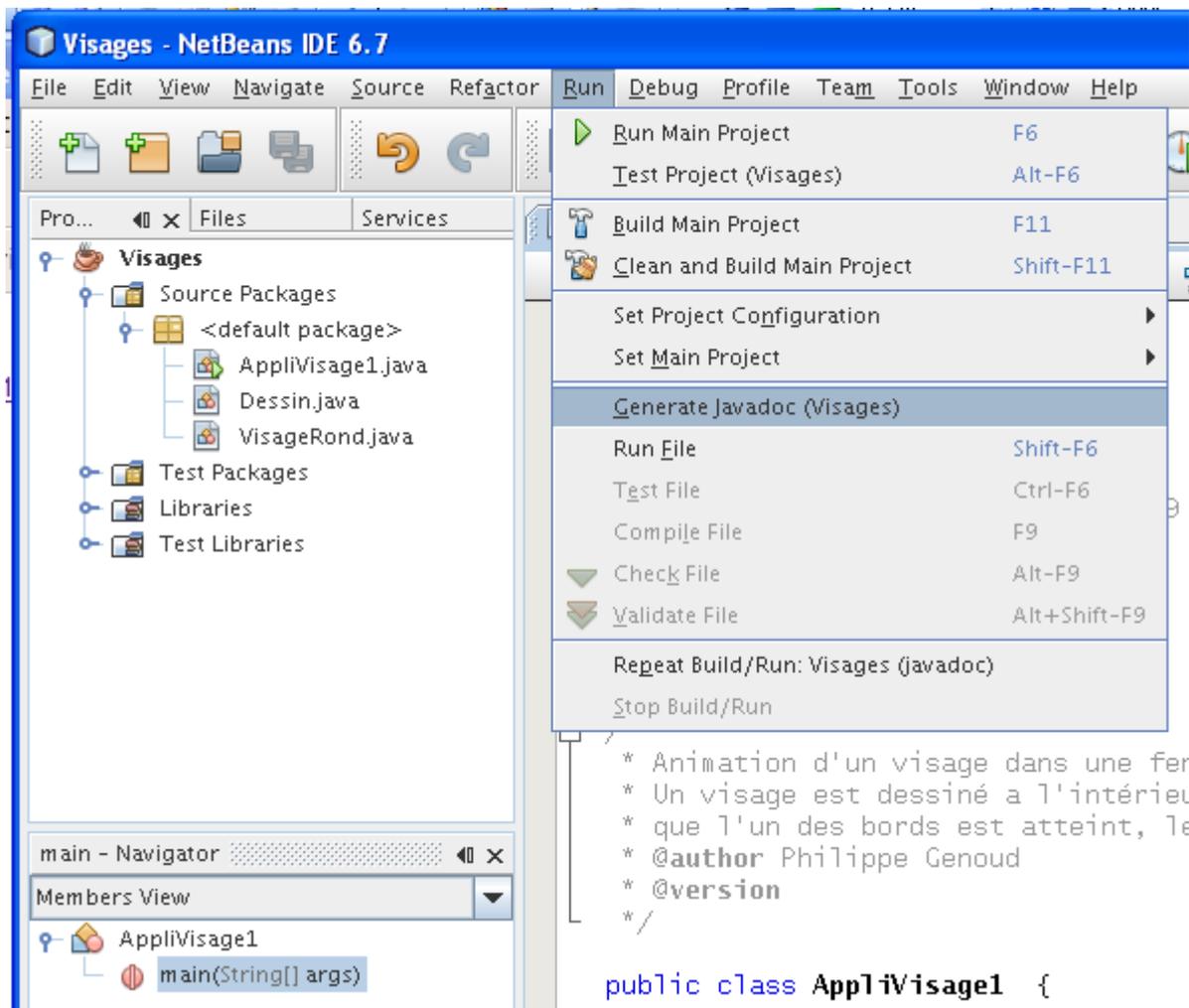
- [AppliVisage1.java](#)
- [VisageRond.java](#)
- [Dessin.java](#)

Compilez et exécutez l'application `AppliVisage1` (Run -> Run Main Project ou F6). L'application `AppliVisage1` est constituée d'une boucle infinie qui à chaque itération effectue un déplacement élémentaire du visage et réaffiche l'image. Lorsqu'il est lancé depuis une console, ce programme peut être arrêté en tapant `CTRL-C`. Le bouton situé à gauche de la fenêtre console vous permet d'interrompre l'exécution d'un programme depuis *NetBeans*.

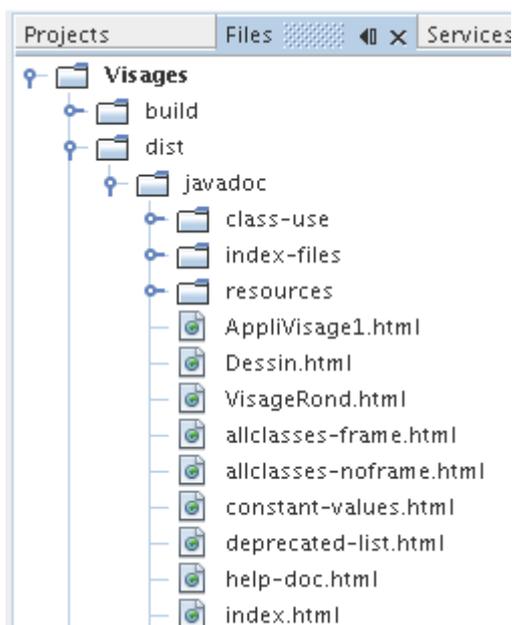


Générer la documentation Javadoc d'un projet

Pour générer la documentation javadoc d'un projet à partir des sources il suffit d'activer la commande Generate Javadoc depuis le menu Run.



Une fois la documentation générée *NetBeans* ouvre une fenêtre du navigateur par défaut affichant cette documentation. Les fichiers générés le sont dans le répertoire `dist/javadoc` situé sous la racine de votre projet. Pour les voir, cliquez sur l'onglet `Files` dans l'explorateur du projet et développez le nœud `dist`.

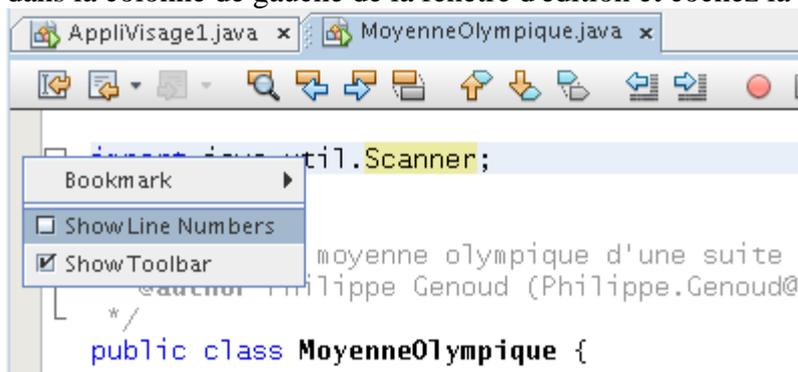


Utiliser le debugger de NetBeans

Créez un nouveau projet d'application Java, moyenne Olympique. Téléchargez la classe [MoyenneOlympique.java](#) qui calcule la moyenne olympique d'une série de notes et recopiez le dans le répertoire `src` de ce projet.

Lancez l'exécution de ce programme. Il a été conçu pour calculer la moyenne d'une série de notes en retirant la note la plus élevée et la note la plus faible. Trois notes au moins doivent être saisies. Cependant un petit bug s'est glissé dans ce programme, alors que la saisie de trois notes devrait suffire, le programme impose un minimum de 4 notes. Pour comprendre l'origine de cette erreur vous allez utiliser le debugger de NetBeans qui vous permettra de positionner des points d'arrêt, d'effectuer une exécution pas à pas et d'observer l'état des variables.

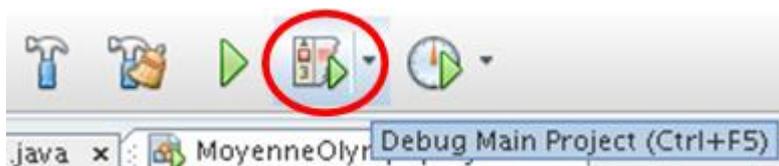
1. Faites apparaître les numéros de lignes dans la fenêtre de l'éditeur. Pour cela effectuer un clic droit dans la colonne de gauche de la fenêtre d'édition et cochez la case `Show Line Numbers`.



2. Placez un point d'arrêt à la ligne 53 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).

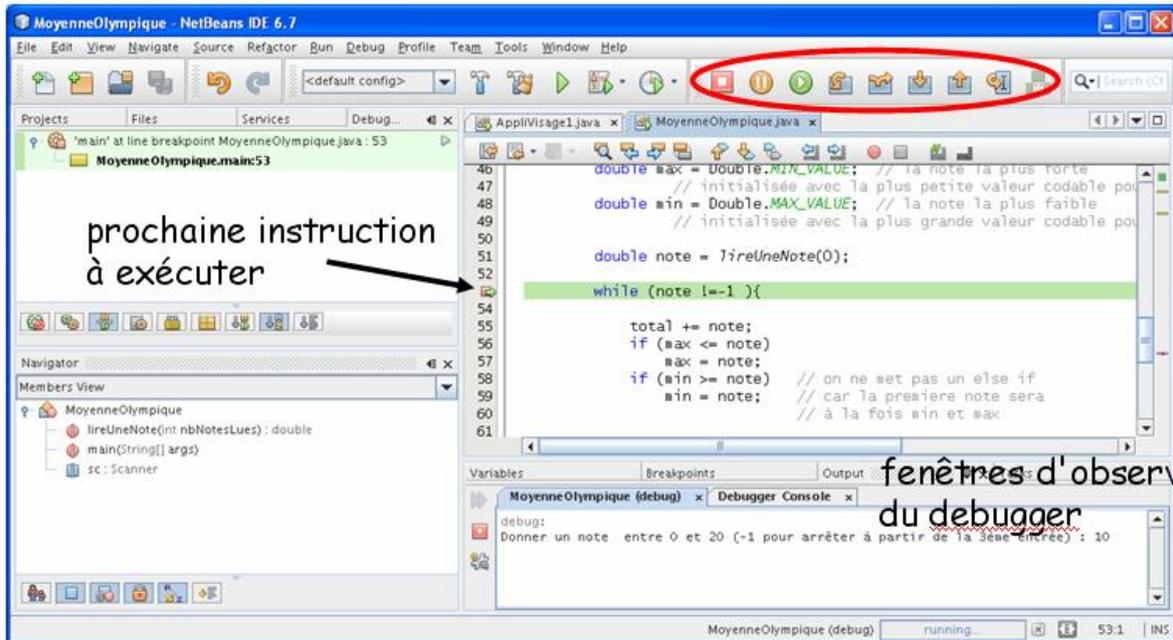
```
50
51     double note = LireUneNote(0);
52
53     while (note != -1 ){
54
55         total += note;
56         if (max <= note)
```

3. 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



prochaine instruction
à exécuter

fenêtres d'observation
du debugger

4. L'exécution est interrompue à la ligne 53 (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 | | |
| Static | | |
| args | String[] | #161(length=0) |
| total | double | 10.0 |
| nbNotes | int | 0 |
| max | double | 10.0 |
| min | double | 1.7976931348623157E308 |
| note | double | double |

5. Lorsque vous arrivez à la ligne 62 (appel de `lireUneNote` dans la boucle `while`) faites avancer l'exécution en rentrant dans la méthode `lireUneNote`. Pour cela cliquez sur le bouton "Step Into"



6. Pour aller directement jusqu'au prochain point d'arrêt cliquez sur le bouton "Continue".
7. 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.



La documentation en ligne de NetBeans

Nous avons vu dans ce qui précède le strict minimum pour l'utilisation de NetBeans dans l'écriture d'applications Java. Pour profiter au mieux des très riches fonctionnalités de cet environnement, nous vous conseillons de parcourir la documentation en ligne. Pour obtenir cette aide, menu `Help-->Help Contents`.

