

## Apprendre Java avec BlueJ, une autre approche (2<sup>ème</sup> partie - Netbeans)

**Reference :** *Objects First with Java – A Practical Introduction Using BlueJ*  
David Barnes & Michael Kölling

Auteur : Boichat Jean-Bernard Email : [jean-bernard@boichat.ch](mailto:jean-bernard@boichat.ch)

Version de cet article : 1.1.1

Publication récente : *Apprendre Java et C++ en parallèle – 4<sup>ème</sup> édition*

<http://www.eyrolles.com/Informatique/Livre/apprendre-java-et-c-en-parallele-9782212124033>

<http://www.boichat.ch/javacpp/>

### Plugin Bluej pour Netbeans

Cette partie est consacrée au plugin BlueJ dans Netbeans.

D'autres articles consacrés à BlueJ et écrit par l'auteur sont disponibles :

1<sup>ère</sup> partie - BlueJ

<http://www.boichat.ch/javacpp/more/articleBlueJ.pdf>

3<sup>ème</sup> partie - BlueJ et Greenfoot

[http://www.boichat.ch/javacpp/more/articleBlueJ\\_Greenfoot.pdf](http://www.boichat.ch/javacpp/more/articleBlueJ_Greenfoot.pdf)

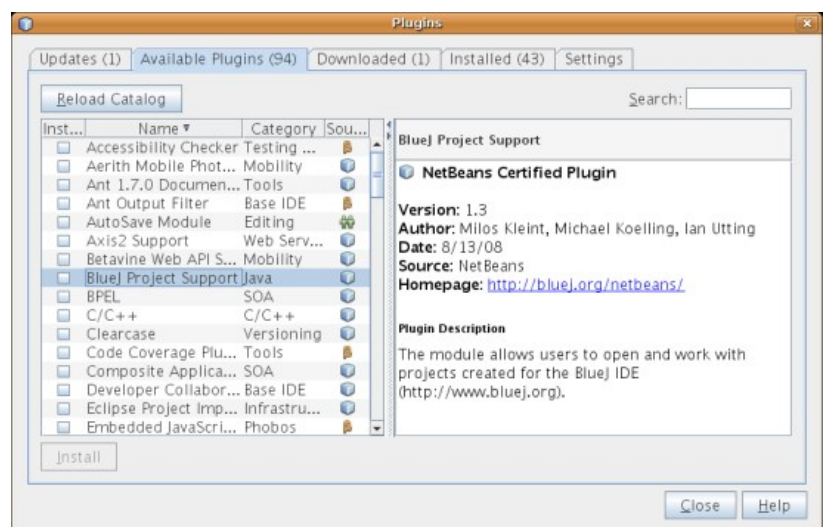
4<sup>ème</sup> partie - BlueJ et Eclipse

[http://www.boichat.ch/javacpp/more/articleBlueJ\\_Eclipse.pdf](http://www.boichat.ch/javacpp/more/articleBlueJ_Eclipse.pdf)

### Installation dans Netbeans

Netbeans<sup>1</sup>, et même en français<sup>2</sup>, est actuellement un des outils de développement Java les plus utilisés, avec Eclipse<sup>3</sup>, bien sûr. Dans mon livre sur Java et C++, nous trouverons d'ailleurs deux annexes entièrement dédiées à Netbeans (Windows et Linux) et aussi pour le C++.

Pour l'installation de **BlueJ** dans Netbeans 6.5 (c'est pareil avec la version 6.1), il suffit de se rendre dans le menu **Tools / Plugin**, et de patienter pour l'update du site de Netbeans :

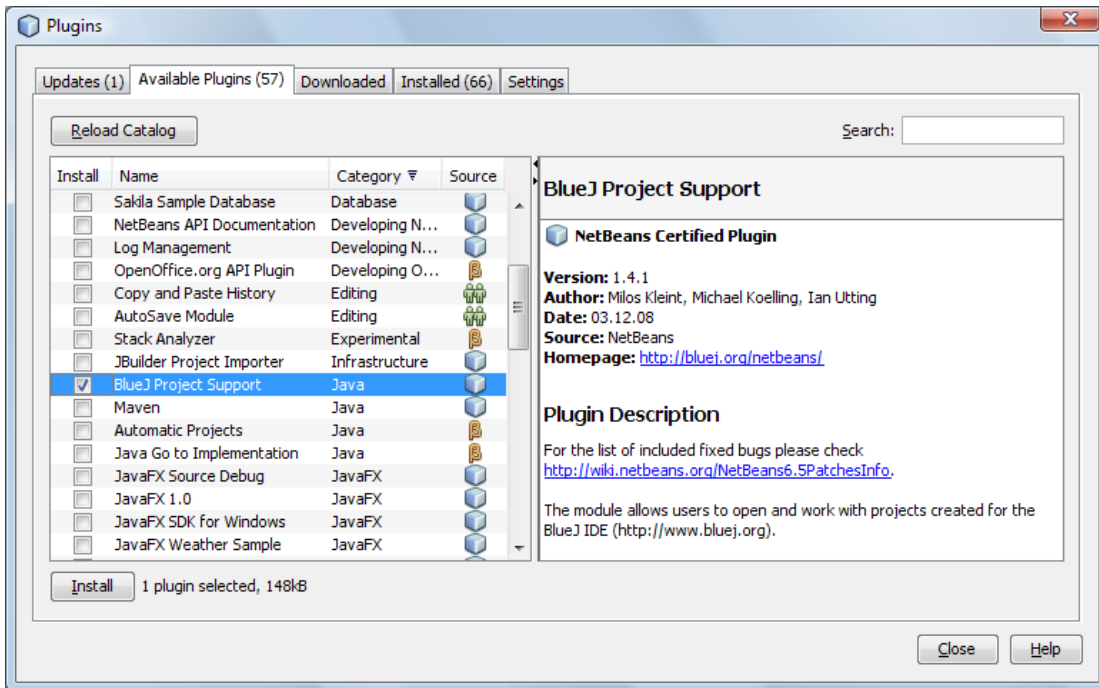


<sup>1</sup> <http://www.netbeans.org/>

<sup>2</sup> <http://fr.netbeans.org/>

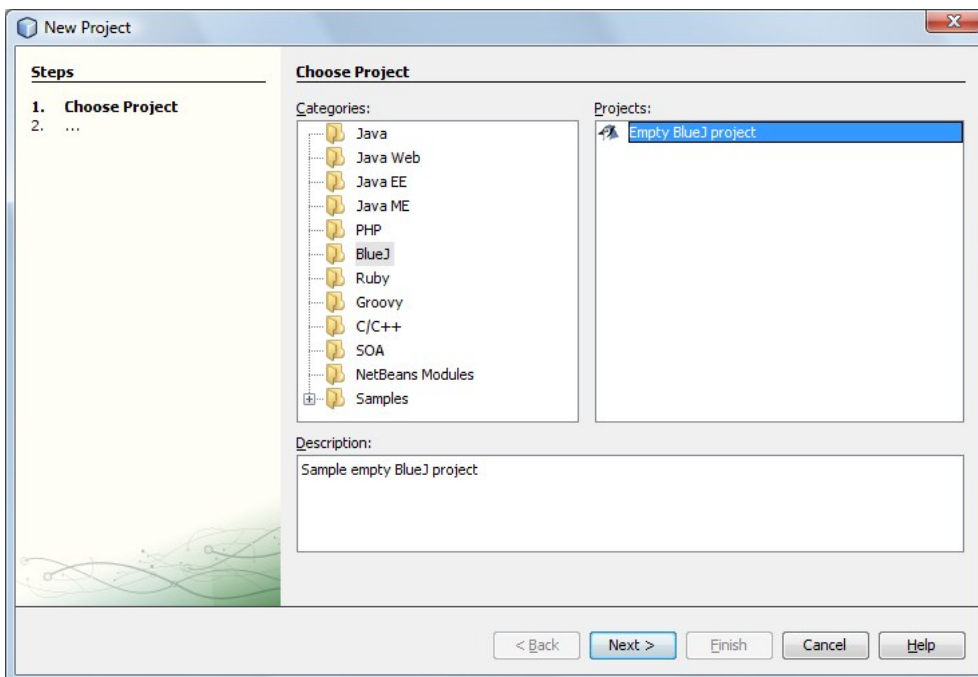
<sup>3</sup> <http://www.eclipse.org/>

Il faudra absolument vérifier que nous avons bien les toutes dernières versions du plugin et de Netbeans (menu : Help / Check for updates).

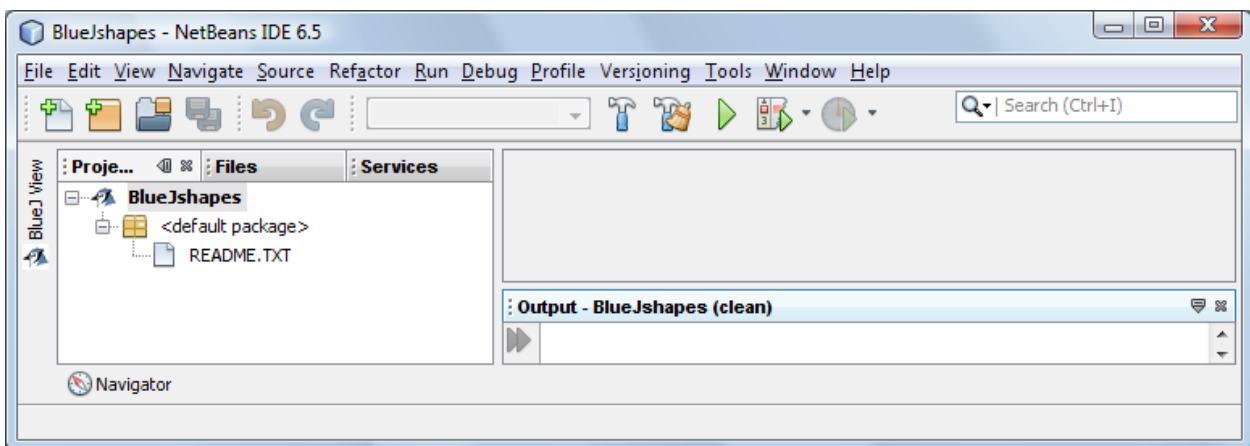


Nous sélectionnerons le BlueJ Project Support, Install, nous accepterons les termes de la licence et le tour sera joué !

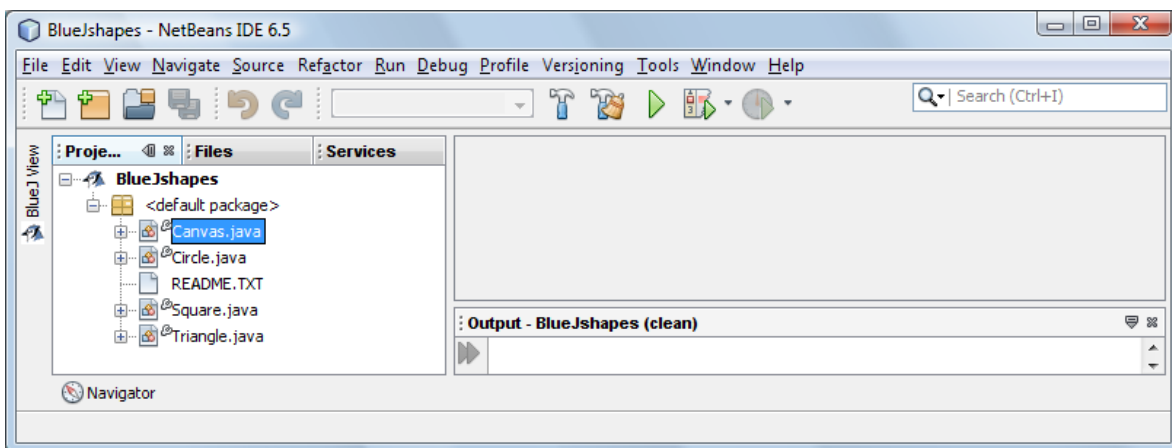
Nous n'allons pas décrire l'utilisation de Netbeans mais nous donnerons cependant quelques pistes. Un aspect qui est sans doute peu connu, c'est l'utilisation d'un groupe de projet (ici BlueJ), similaire à un workspace sous Eclipse. Nous allons créer un nouveau projet avec du code existant, c'est-à-dire celui du répertoire **shapes** que nous avons utilisé dans mon premier article sur BlueJ.



Nous pouvons seulement créer un projet vide (empty) que nous nommerons BlueJshapes :



Dans notre répertoire de travail ...\\Netbeans\\BlueJshapes nous copierons à la main avec l'explorateur de Window, nos 4 classes .java depuis notre répertoire shapes du premier article :

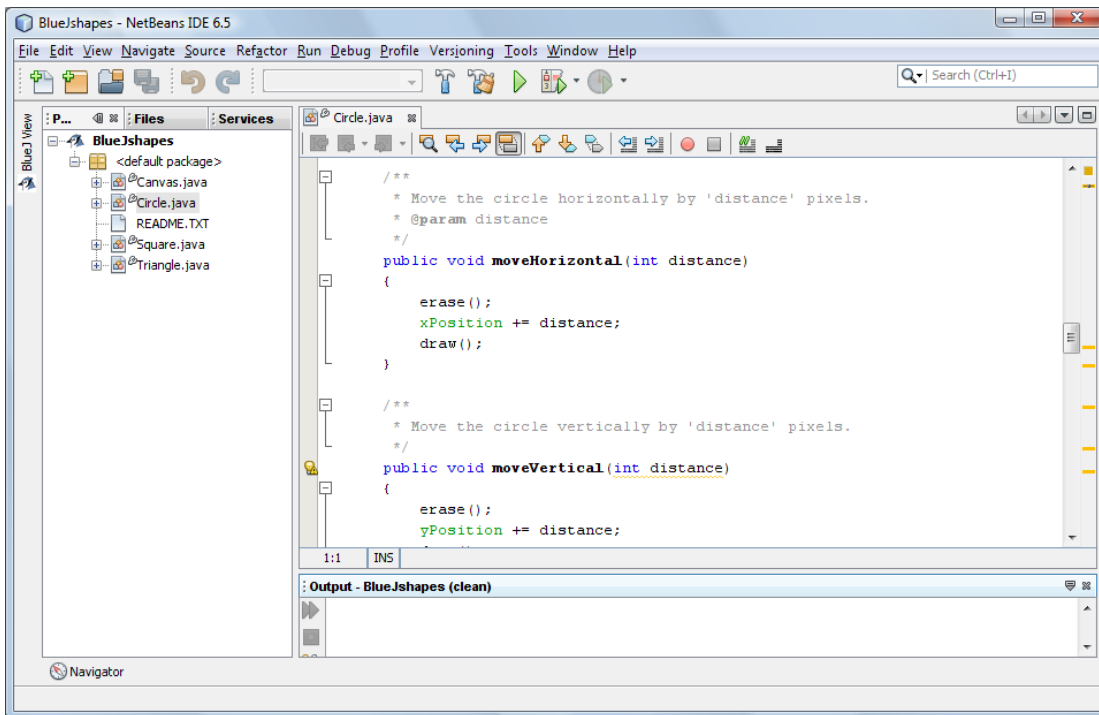


En double cliquant sur Circle.java nous aurons notre classe présente dans l'éditeur.

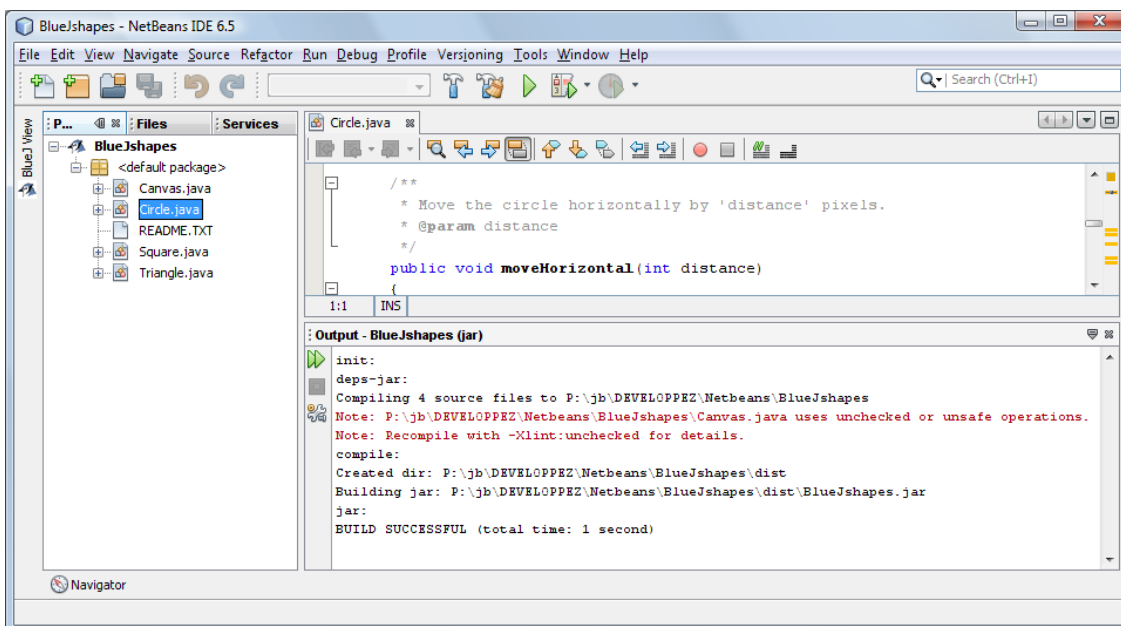
Netbeans est évidemment plus évolué que BlueJ et va nous indiquer, avec ces petites lampes jaunes, que la documentation Javadoc n'est pas parfaite. Cet outil puissant pourra nous aider à faire les corrections nécessaires, par exemple :

```
/**
 * Move the circle horizontally by 'distance' pixels.
 * @param distance
 */
public void moveHorizontal(int distance)
```

Il manquait effectivement la ligne param :



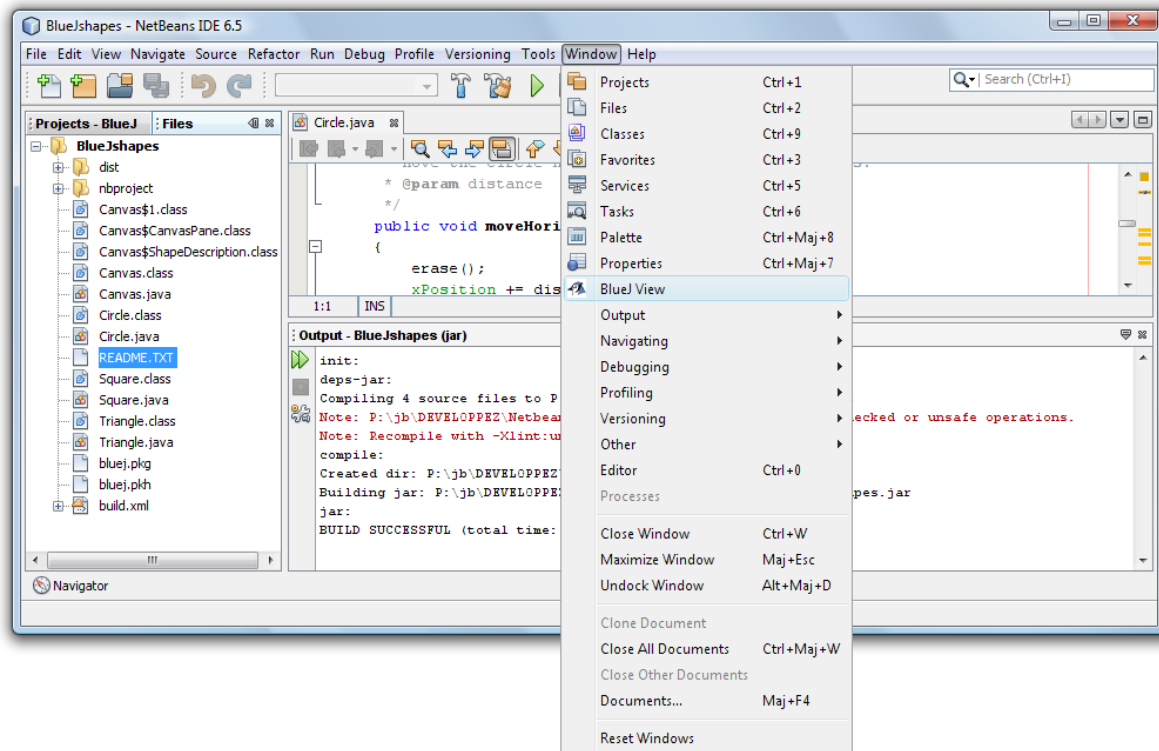
Nous pourrions évidemment compiler (build) le projet (et produire un fichier `BlueJshapes.jar`) ou encore essayer de l'exécuter !



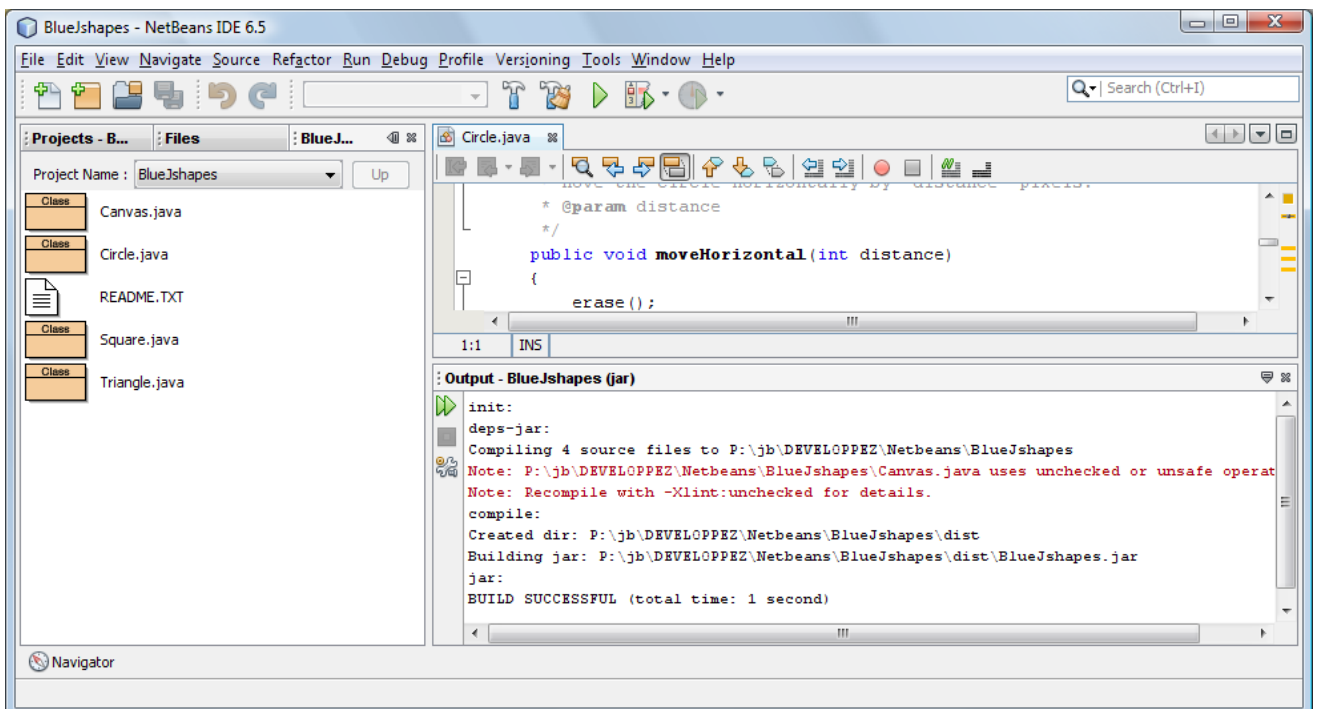
Mais cela ne va pas nous aider, car il n'y a pas d'entrée `main()` dans aucune des sources. Nous connaissons déjà ce problème et nous l'avons étudié dans le premier article. Il y a aussi des corrections que nous pourrions apporter à `Canvas.java` car nous avons ici une version de Java plus récente où il y a des dépréciations (compilation avec `-Xlint` requis :).

## Le plugin visuel de BlueJ dans Netbeans

Pour obtenir une « vue » BlueJ dans Netbeans, il nous faut sélectionner cette fonction avec Windows / BlueJ View :

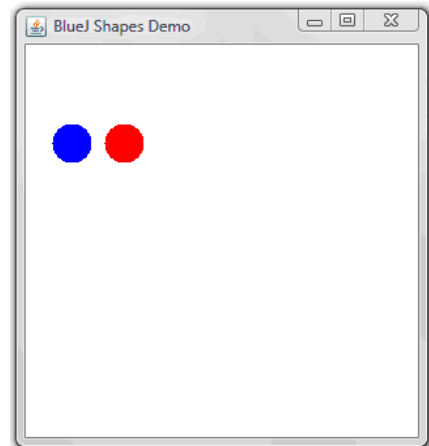


Nous obtiendrons alors une représentation connue (voir le premier article) :



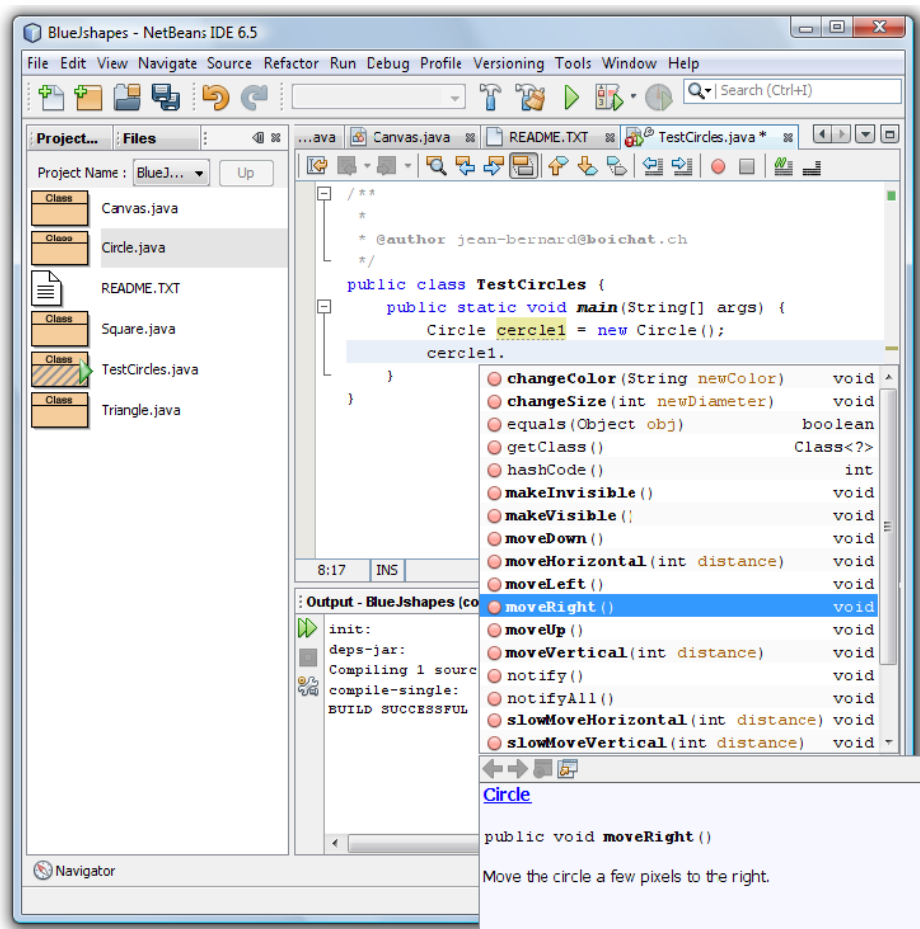
Dans notre premier article nous avons créé deux objets de la classe `Circle` afin d'obtenir ce résultat :

- un cercle bleu à son endroit original tel que définit par le constructeur
- un cercle identique mais en rouge et déplacer deux fois à droite



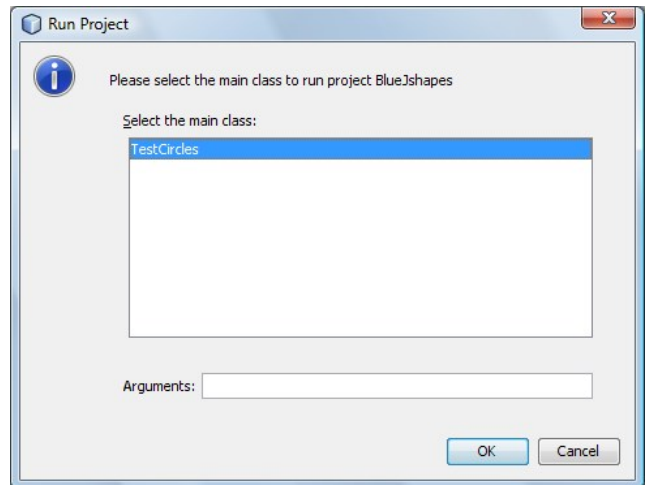
Pour obtenir cette « démonstration », nous devons créer ici une classe de test qui va contenir non seulement un `main()`, mais aussi toutes les actions que nous avons définies.

Un des avantages de Netbeans est que dès que nous avons entré le `.` suivant un objet de classe (ici `cercle1.`, il nous présente toutes les méthodes disponibles avec leur Javadoc (d'où un gros `++` si la documentation est bien écrite) :



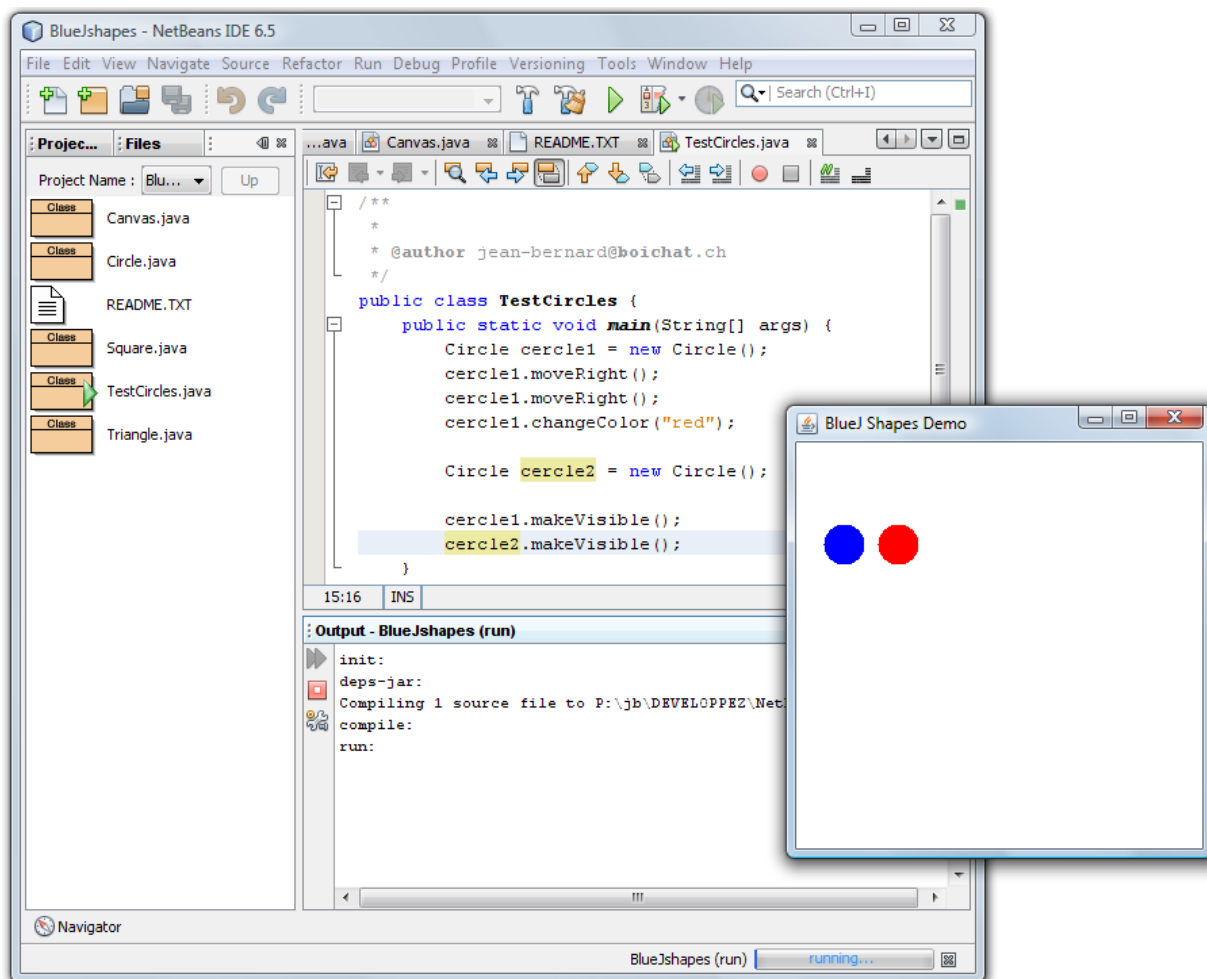
Lorsque nous aurons terminé notre travail et déciderons de compiler et exécuter notre code (menu Run ou F6), nous constaterons que Netbeans à enfin trouvé un point d'entrée :

Dès ce moment là, si nous reconstruisons le projet (F11) et notre application finale, c'est-à-dire `BlueJshapes.jar`, celle-ci possèdera une point d'entré `main()`. Nous pourrons alors la distribuer sur d'autres machines ayant une machine virtuelle Java (`jre`) avec au minimum la version 6.



Dans la présentation ci-dessous nous remarquerons bien que :

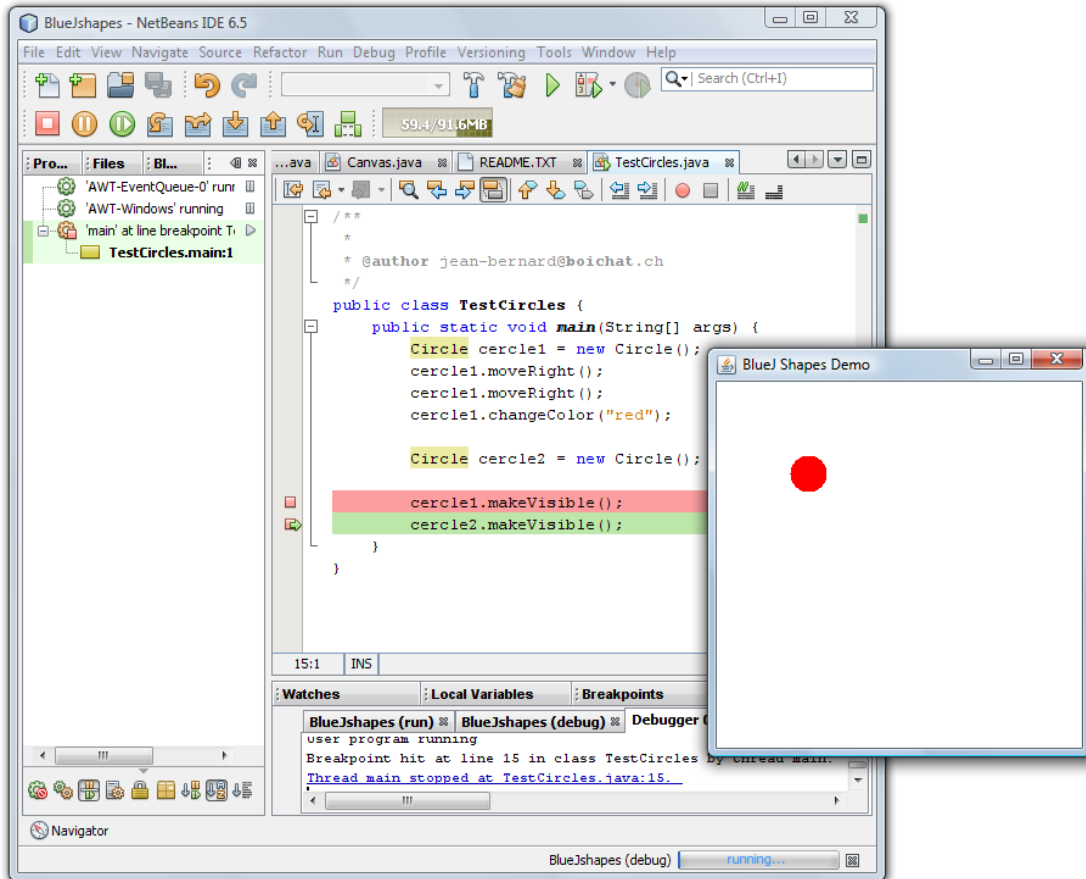
- les méthodes requises sur chaque objet `cercle1` et `circle2`
- nos deux objets dans la fenêtre d'exécution « BlueJ Shapes Demo »





Avec le débogueur (Ctrl-F5), et des points d'arrêts correctement positionné, nous pourrions constater que les objets de la classe `Circle` n'apparaîtront qu'au moment des `makeVisible()`. De plus la fenêtre `Window` n'apparaîtra qu'au tout premier `makeVisible()`.

Ces aspects et le pourquoi sont expliqués en détails dans le premier article.



## Conclusion

Pour conclure, nous indiquerons que Netbeans est évidemment plus appropriés pour vérifier pas à pas, avec un débogueur plus évolué que celui de BlueJ, des méthodes complexes où il est nécessaire de vérifier le contenu de plusieurs variables ou objets disséminés. Le jeu d'Othello, présenté avec des parties de code dans mon livre, est un exemple typique de l'utilisation efficace d'un outil comme Netbeans.

Par contre, lorsqu'une vue objet orienté du langage Java doit être présentée, principalement à des fins éducationnel, des « outils » comme BlueJ ou UML<sup>4</sup> évidemment gardent tout leurs intérêts.

<sup>4</sup> [http://fr.wikipedia.org/wiki/Unified\\_Modeling\\_Language](http://fr.wikipedia.org/wiki/Unified_Modeling_Language) et <http://wiki.netbeans.org/UML>