

# NFA031 : Programmation avec Java : notions de base

CM1-3 : Java, outils

Mickaël Martin Nevot

V1.1.0



Cette œuvre de [Mickaël Martin Nevot](#) est mise à disposition selon les termes de la [licence Creative Commons Attribution – Pas d'Utilisation Commerciale – Partage à l'Identique 3.0 non transposé](#).

# Programmation avec Java : notions de base

- I. Présentation
- II. Introduction aux objets
- III. Java
- IV. Types
- V. Outils

# Classe Object / méthode main

- `Object` :
  - **Classe de plus haut niveau** dans la hiérarchie d'héritage
  - Toute classe autre que `Object` possède une super-classe
  - Toute classe hérite directement ou pas de `Object`
  - Toute classe qui n'a pas de clause `extends` hérite de `Object`
- `main( ... )` :
  - Point de commencement d'exécution du code
  - Au moins une par application (**classe principale**) :

```
public static void main(String[] args) {  
    // Le code va commencer par s'exécuter ici.  
}
```

# A savoir

- Classe String :

```
String myString = "Hello!";  
myString += "How are you?";
```

- Classe File :

```
File myFile = new File("file.txt");
```

- Affichage (y compris types primitifs/références) :

```
System.out.println("a = " + a);
```



# Javadoc

- Outil standard pour créer une **documentation** d'API
- Génération automatique en HTML
- Utilisation ( $\neq$  commentaire `/* */`) :
  - Première ligne : uniquement `/**`
  - Lignes suivantes : un espace suivi de `*`
  - Dernière ligne : un espace suivi uniquement de `*/`
  - L'entité documentée est précédée par son commentaire
  - Tags prédéfinis

Bonne utilisation : expliquer n'est pas traduire !

# Javadoc : principaux tags

- `@author` : nom du développeur
- `@version` : version d'une classe/méthode
- `@param` : définit un paramètre de méthode :  
requis pour chaque paramètre
- `@since` : version du JDK de l'apparition de la classe/méthode
- `@return` : valeur de retour
- `@throws` : classe de l'exception et conditions de lancement
- `@deprecated` : marque la méthode comme dépréciée
- `@see` : référence croisée avec un autre élément

# Exemple de Javadoc

```
/**
```

- \* Valide un mouvement de jeu d'Échecs.
- \* @param beginCol Colonne de la case de départ
- \* @param beginRow Ligne de la case de départ
- \* @param endCol Colonne de la case de destination
- \* @param endRow Ligne de la case de destination
- \* @return vrai(true) si le mouvement d'échec est valide ou faux(false) sinon

```
*/
```

java.lang  
Class Object

java.lang.Object

```
public class Object
```

Class Object is the root of the class hierarchy. Every class has Object as a superclass. All objects, including arrays, implement the methods of this class.

Since:  
JDK1.0

See Also:  
[Class](#)

**Constructor Summary**

```
Object ()
```

**Method Summary**

protected Object	<a href="#">clone ()</a>	Creates and returns a copy of this object.
boolean	<a href="#">equals (Object obj)</a>	Indicates whether some other object is "equal to" this one.
protected void	<a href="#">finalize ()</a>	Called by the garbage collector on an object when garbage collection determines that there are no more references to the object.
Class	<a href="#">getClass ()</a>	Returns the runtime class of an object.
int	<a href="#">hashCode ()</a>	

## Method Detail

### getClass

```
public final Class getClass ()
```

Returns the runtime class of an object. That `Class` object is the object that is locked by `static synchronized` methods of the represented class.

#### Returns:

the object of type `Class` that represents the runtime class of the object.

### hashCode

```
public int hashCode ()
```

Returns a hash code value for the object. This method is supported for the benefit of hash tables such as those provided by `java.util.Hashtable`.

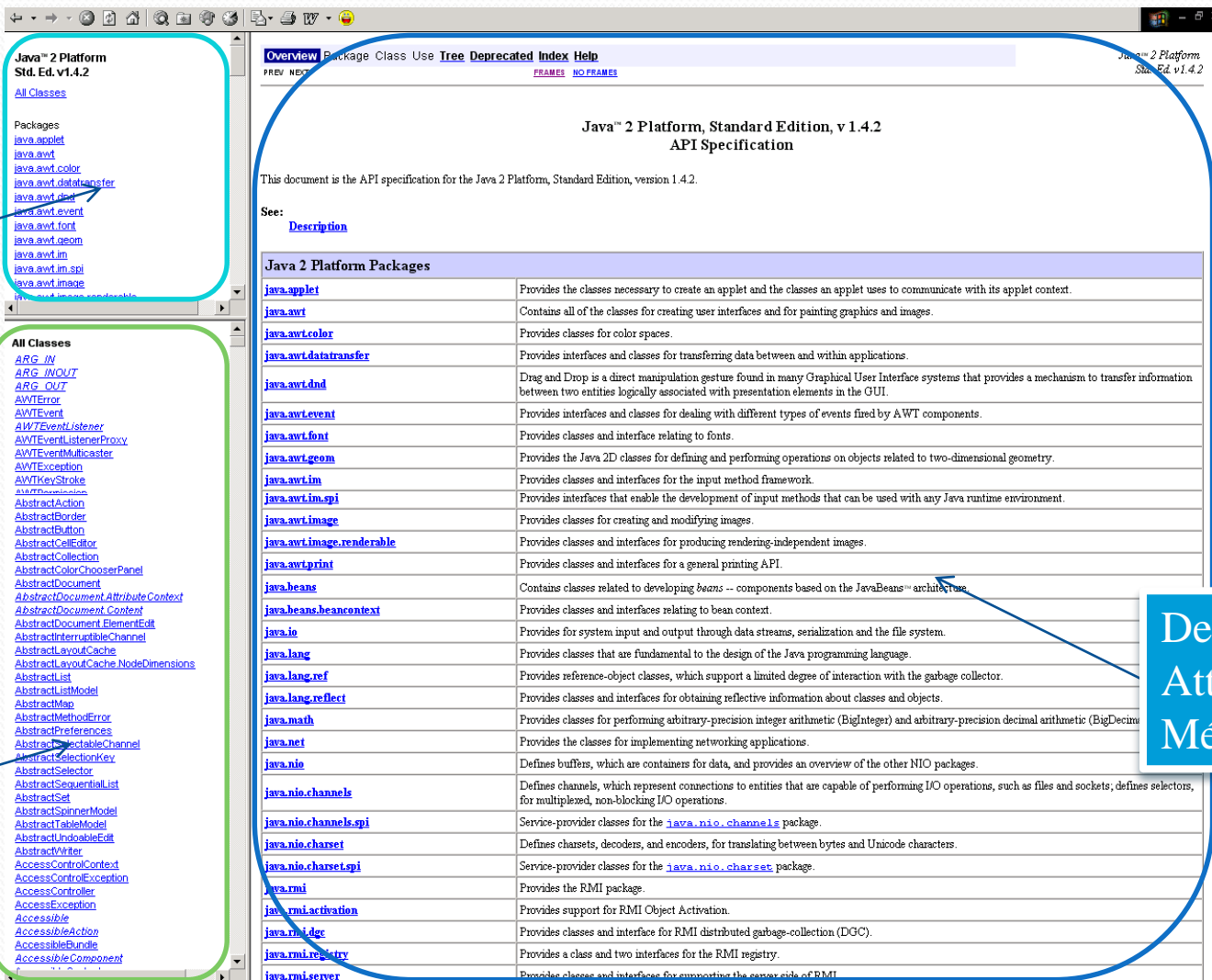
The general contract of `hashCode` is:

- Whenever it is invoked on the same object more than once during an



# API

Paquetages



**Overview** Package Class Use Tree Deprecated Index Help

PREV NEXT FRAMES NO FRAMES

## Java™ 2 Platform, Standard Edition, v 1.4.2 API Specification

This document is the API specification for the Java 2 Platform, Standard Edition, version 1.4.2.

See: [Description](#)

### Java 2 Platform Packages

<a href="#">java.applet</a>	Provides the classes necessary to create an applet and the classes an applet uses to communicate with its applet context.
<a href="#">java.awt</a>	Contains all of the classes for creating user interfaces and for painting graphics and images.
<a href="#">java.awt.color</a>	Provides classes for color spaces.
<a href="#">java.awt.datatransfer</a>	Provides interfaces and classes for transferring data between and within applications.
<a href="#">java.awt.dnd</a>	Drag and Drop is a direct manipulation gesture found in many Graphical User Interface systems that provides a mechanism to transfer information between two entities logically associated with presentation elements in the GUI.
<a href="#">java.awt.event</a>	Provides interfaces and classes for dealing with different types of events fired by AWT components.
<a href="#">java.awt.font</a>	Provides classes and interface relating to fonts.
<a href="#">java.awt.geom</a>	Provides the Java 2D classes for defining and performing operations on objects related to two-dimensional geometry.
<a href="#">java.awt.im</a>	Provides classes and interfaces for the input method framework.
<a href="#">java.awt.im.spi</a>	Provides interfaces that enable the development of input methods that can be used with any Java runtime environment.
<a href="#">java.awt.image</a>	Provides classes for creating and modifying images.
<a href="#">java.awt.image.renderable</a>	Provides classes and interfaces for producing rendering-independent images.
<a href="#">java.awt.print</a>	Provides classes and interfaces for a general printing API.
<a href="#">java.beans</a>	Contains classes related to developing <i>beans</i> -- components based on the JavaBeans™ architecture.
<a href="#">java.beans.beancontext</a>	Provides classes and interfaces relating to bean context.
<a href="#">java.io</a>	Provides for system input and output through data streams, serialization and the file system.
<a href="#">java.lang</a>	Provides classes that are fundamental to the design of the Java programming language.
<a href="#">java.lang.ref</a>	Provides reference-object classes, which support a limited degree of interaction with the garbage collector.
<a href="#">java.lang.reflect</a>	Provides classes and interfaces for obtaining reflective information about classes and objects.
<a href="#">java.math</a>	Provides classes for performing arbitrary-precision integer arithmetic (BigInteger) and arbitrary-precision decimal arithmetic (BigDecimal).
<a href="#">java.net</a>	Provides the classes for implementing networking applications.
<a href="#">java.nio</a>	Defines buffers, which are containers for data, and provides an overview of the other NIO packages.
<a href="#">java.nio.channels</a>	Defines channels, which represent connections to entities that are capable of performing I/O operations, such as files and sockets; defines selectors, for multiplexed, non-blocking I/O operations.
<a href="#">java.nio.channels.spi</a>	Service-provider classes for the <a href="#">java.nio.channels</a> package.
<a href="#">java.nio.charset</a>	Defines charsets, decoders, and encoders, for translating between bytes and Unicode characters.
<a href="#">java.nio.charset.spi</a>	Service-provider classes for the <a href="#">java.nio.charset</a> package.
<a href="#">java.rmi</a>	Provides the RMI package.
<a href="#">java.rmi.activation</a>	Provides support for RMI Object Activation.
<a href="#">java.rmi.dgc</a>	Provides classes and interface for RMI distributed garbage-collection (DGC).
<a href="#">java.rmi.registry</a>	Provides a class and two interfaces for the RMI registry.
<a href="#">java.rmi.server</a>	Provides classes and interfaces for supporting the server side of RMI.

**All Classes**

- [ARG\\_IN](#)
- [ARG\\_INOUT](#)
- [ARG\\_OUT](#)
- [AWTError](#)
- [AWTEvent](#)
- [AWTEventListener](#)
- [AWTEventListenerProxy](#)
- [AWTEventMulticaster](#)
- [AWTException](#)
- [AWTKeyStroke](#)
- [AWTMouseEvent](#)
- [AbstractAction](#)
- [AbstractBorder](#)
- [AbstractButton](#)
- [AbstractCellEditor](#)
- [AbstractCollection](#)
- [AbstractColorChooserPanel](#)
- [AbstractDocument](#)
- [AbstractDocument.AttributeContext](#)
- [AbstractDocument.Content](#)
- [AbstractDocument.ElementEdit](#)
- [AbstractInterruptibleChannel](#)
- [AbstractLayoutCache](#)
- [AbstractLayoutCache.NodeDimensions](#)
- [AbstractList](#)
- [AbstractListModel](#)
- [AbstractMap](#)
- [AbstractMethodError](#)
- [AbstractPreferences](#)
- [AbstractSelectableChannel](#)
- [AbstractSelectableKey](#)
- [AbstractSelector](#)
- [AbstractSequentialList](#)
- [AbstractSet](#)
- [AbstractSpinnerModel](#)
- [AbstractTableModel](#)
- [AbstractUndoableEdit](#)
- [AbstractWriter](#)
- [AccessControlContext](#)
- [AccessControlException](#)
- [AccessController](#)
- [AccessException](#)
- [Accessible](#)
- [AccessibleAction](#)
- [AccessibleBundle](#)
- [AccessibleComponent](#)

Classes

Description  
Attributs  
Méthodes



# Outils

- Éditeur :

- Eclipse : <http://www.eclipse.org>

- Ressources Java :

- API :

<http://download.oracle.com/javase/1.5.0/docs/api>

- Convention de nommage :

<http://java.sun.com/docs/codeconv/CodeConventions.pdf>

- Mots clefs réservés :

[http://download.oracle.com/javase/tutorial/java/nu  
tsandbolts/\\_keywords.html](http://download.oracle.com/javase/tutorial/java/nu<br/>tsandbolts/_keywords.html)



# Bonnes pratiques

- Penser à l'initialisation pour éviter une erreur
- Penser à construire les objets avant de les utiliser
- Penser à l'utilisation de `break` dans un `switch`
- Attention à l'encapsulation
- Utiliser le mot clef `this` autant de fois que possible
- Pas de mot clef `then` (en relation avec un `if`)
- Pas d'utilisation de variable d'instance ni du mot clef `this` dans une méthode de classe
- Pas d'héritage multiple

# Crédits

## Auteur

Mickaël Martin Nevot

[mmartin.nevot@gmail.com](mailto:mmartin.nevot@gmail.com)



Carte de visite électronique

## Relecteurs

Cours en ligne sur : [www.mickaël-martin-nevot.com](http://www.mickaël-martin-nevot.com)

