

# Programmation orientée objet en langage JAVA

## Les applications embarquées dans des pages HTML : les Applet

Claude Duvallet

Université du Havre  
UFR Sciences et Techniques  
25 rue Philippe Lebon - BP 540  
76058 LE HAVRE CEDEX  
Claude.Duvallet@gmail.com  
<http://litis.univ-lehavre.fr/~duvallet/>

## Introduction aux Applet Java

- En Java, une applet est application embarquée dans une page HTML.
- Une applet s'exécute dans un logiciel de navigation supportant le Java ou encore dans l'appviewer du JDK.
- Elles sont généralement constituées d'une IHM similaire à celles qui sont présentes dans les applications autonomes.
- Pour la conception des applet, il est possible d'utiliser soit le modèle AWT, soit le modèle SWING.
- La conception de l'applet commence par la construction d'une classe qui hérite soit de la classe `Applet` (AWT) ou `JApplet` (Swing) :
  - Cette classe doit comporter une méthode `public void init()` qui constituera la méthode d'entrée dans l'applet.
  - Il n'est pas nécessaire d'écrire une méthode `public void static main(String args[])`

## Une première applet

```
import javax.swing.*;
import java.awt.*;

public class MonApplet extends JApplet {

    public void init () {
        setLayout (new FlowLayout ());
        add (new JTextField ("Saisie d'une chaîne de caractères"));
        add (new JButton ("Bouton 1"));
        add (new JButton ("Bouton 2"));
    }
}
```



## La page HTML de l'applet

### Deux possibilités :

```
<APPLET code="MonApplet.class" width="500" height="200">  
Exemple d'applet.  
</APPLET>
```

### ou encore

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">  
  
  <body>  
    Présentation d'un formulaire réaliser en Java<br />  
    <object codetype="application/java"  
      codebase="."  
      classid="java:MonApplet.class"  
      width="400" height="350">  
      <param name="code" value="MonApplet.class" />  
      Votre navigateur ne permet pas d'exécuter des applet JAVA.  
    </object>  
  </body>  
</html>
```

## Les différentes option du TAG applet

Comme nous l'avons vu dans l'exemple, il est possible de spécifier des options lors de l'intégration de l'applet dans la page HTML :

**CODE** permet de préciser le code à charger (généralement une classe JAVA).

**WIDTH et HEIGHT** permettent de préciser la taille de la fenêtre dans la page HTML. L'utilisation des méthodes `setBounds()` ou `setSize()` ne sont pas suffisantes voire inutiles.

**CODEBASE** permet de spécifier le chemin relatif par rapport au dossier de la page contenant l'applet. Ce paramètre suit le paramètre CODE.

**HSPACE et VSPACE** permettent de fixer la distance en pixels entre l'applet et le texte.

**ALT** affiche le texte spécifié par le paramètre lorsque le navigateur ne supporte pas Java ou que son support est désactivé.

## Les différentes méthodes utilisables dans une applet

- Une classe dérivée de la classe JApplet hérite de ses méthodes.
- Il est possible de redéfinir ces méthodes en fonction des besoins.
- La classe doit être déclarée public pour fonctionner.
- En général, il n'est pas nécessaire de faire un appel explicite aux méthodes `init()`, `start()`, `stop()` et `destroy()` : le navigateur se charge d'appeler ces méthodes.
- L'appel est effectué en fonction de l'état de la page HTML contenant l'applet.

## Description des différentes méthodes

- **La méthode `init()`.** Cette méthode permet l'initialisation de l'applet : elle n'est exécutée qu'une seule et unique fois après le chargement de l'applet. On peut la comparer au constructeur et à la méthode `main()` d'une classe héritant de `JFrame`
- **La méthode `start()`.** Cette méthode est appelée automatiquement après le chargement et l'initialisation (via la méthode `init()`) lors du premier affichage de l'applet.
- **La méthode `stop()`.** Le navigateur appelle automatiquement la méthode lorsque l'on quitte la page HTML. Elle interrompt les traitements de tous les processus en cours.
- **La méthode `destroy()`.** Elle est appelée après l'arrêt de l'applet ou lors de l'arrêt de la machine virtuelle. Elle libère les ressources et détruit les threads restants.

## Transformer une applet en application autonome

- Il est possible de construire une application qui s'exécutera :
  - à la fois comme une applet,
  - et à la fois comme une application autonome fenêtrée.
- Il faut ajouter une méthode `main()` intégrant l'applet dans une fenêtre.

```
import javax.swing.*;  
import java.awt.*;  
  
public class MonAppletAutonome extends JApplet {  
  
    public void init () { ... } // Comme le premier exemple  
  
    public static void main(String args []){  
        MonAppletAutonome applet = new MonAppletAutonome ();  
        JFrame fenetre = new JFrame ("Exemple d'Applet Autonome");  
        fenetre.add (applet);  
        fenetre.setBounds (100,100,400,350);  
        applet.init();  
        applet.start();  
        fenetre.setVisible (true);  
    }  
}
```



## Création d'une applet écrivant dans un fichier

- Lorsqu'une applet doit écrire dans un fichier, cela pose quelques problèmes de sécurité.
- Par défaut, il n'est pas possible d'écrire dans un fichier à partir d'une applet.
- Il faudra donc signer le JAR qui contiendra la/les classe(s) de l'applet si le chargement se fait à partir d'un navigateur WEB.
- Si le lancement se fait avec l'appletviewer, il faudra écrire un fichier `java.policy` ou encore modifier celui qui est présent dans le jre à l'adresse `$(HOME)/jre/lib/security/java.policy` pour y ajouter des droits.

## Exemple (1/2)

- Soit une applet voulant écrire dans un fichier texte appelé `MonFichier.txt`.
- La classe principale de l'applet s'appelle `MonApplet` et elle est stockée dans une archive jar : `MonApplet.jar`.
- Exécution avec l'appletviewer :
  - Créer un fichier `java.policy.applet` :

```
grant {  
    permission java.io.FilePermission "MonFichier.txt", "write";  
};
```
  - Lancement de l'appletviewer :

```
appletviewer -J-Djava.security.policy=java.policy.applet MonApplet.html
```

## Exemple (2/2)

- Exécution dans un navigateur WEB :

- Générer un fichier de certificat :

```
keytool -genkey -keyalg rsa -alias nomalias
```

Entrer un mot de passe pour le "keystore" : un fichier .keystore est créé à la racine de votre compte s'il n'existe pas sinon il faut entre le même mot de passe que lors de sa création.

Entrer le mot de passe pour l'alias qui va être ajouter dans le "keystore".

- Signer le fichier jar contenant votre applet :

```
jarsigner MonApplet.jar nomalias.
```

- Vous pouvez désormais exécuter votre applet depuis un navigateur WEB.

## Page HTML avec un fichier jar

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">
  <head>
    <title>Exécution d'une applet</title>
  </head>
  <body>
    <p>
      Présentation de l'applet<br />
      <object codetype="application/java"
        codebase="."
        archive="MonApplet.jar"
        classid="java:MonApplet.class"
        width="700" height="450">
        <param name="code" value="MonApplet.class" />
        Votre navigateur ne permet pas d'exécuter des applet JAVA.
      </object>
    </p>
  </body>
</html>
```