

UEF4.3. Programmation Orientée Objet

Mme SADEG

s_sadeg@esi.dz

Ecole nationale Supérieure d'Informatique (ESI)

2010/2011

Chapitre VI

Les applets

Introduction

- Java permet de développer deux sortes de programmes: les applications et les applets.
- La vocation d'une applet est d'être téléchargé sur une machine donnée à partir d'une machine distante qui en fournit le code. Ce chargement est toujours provoqué par l'analyse d'un fichier contenant des commandes HTML.

Première applet (1)

- Une applet a beaucoup de points communs avec une application, qui crée une fenêtre graphique. Toutes les possibilités qui ont été vues auparavant s'appliquent à une applet. Néanmoins il existe quelques différences.
- Une applet est obligatoirement constituée d'une classe dérivée de JApplet. Au lancement d'une fenêtre on dispose automatiquement d'une fenêtre graphique ce qui n'est pas le cas avec une application. D'autre part, les dimensions initiales de cette fenêtre sont définies par des commandes du fichier HTML lançant l'applet, et non par le code de l'applet lui-même.

Première applet (2)

- Un fichier HTML, est formé de commandes qui décrivent le contenu d'une page web: textes, graphiques, liens hypertextes et , éventuellement, applets. Un fichier HTML qui lance une applet contiendra une commande particulière fournissant au minimum les informations suivantes:

CODE = Nom du fichier.class contenant le byte code de l'applet;
par exemple :CODE = "PremApplet.class"

WIDTH = largeur initiale de la fenêtre consacrée à l'applet, par exemple: WIDTH = 350

HEIGHT= Hauteur initiale de la fenêtre consacrée à l'applet, par exemple: HEIGHT = 100.

Première applet(3)

Lancement d'une applet

- Nous avons vu comment lancer une application, qu'elle soit à interface graphique ou console.
- Les applets par nature, sont destinées à être lancées dans une page web, par un navigateur qui exploite alors le fichier HTML correspondant. Une applet ressemble donc à une application graphique mais elle ne crée pas de fenêtre console.
- Pour faciliter la tâche de mise au point d'une applet, tous les navigateurs disposent d'un logiciel dit " visualisateur d'applets". Il permet d'exécuter une applet, sans qu'il soit nécessaire de se connecter au web. Il requiert le nom du fichier HTML . Bien entendu, le code source de l'applet doit avoir été préalablement compilé sur la machine locale.

Première applet(3)

création d'une applet

- Nous allons créer une applet qui affiche "Bonjour le monde !". Cela nécessite l'écriture d'au moins deux fichiers :
 - le fichier **source java** à compiler
 - et le fichier **html** qui permettra d'exécuter l'applet.

Le code source java (1)

- La classe que nous créons est une classe dérivée de la classe JApplet. Notre classe va utiliser un champ nommé msg de type String qui contiendra le message à afficher. Elle surchargera aussi deux méthodes héritées de la classe Applet qui ont un rôle spécifique :
 - la méthode **init** qui est appelé une seule fois au démarrage de l'applet et qui permet d'effectuer des initialisation,
 - la méthode **paint** qui est appelé à chaque fois qu'il est nécessaire d'afficher l'applet

obtient le code suivant :

On obtient le code suivant

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

public class hello extends JApplet {
    String msg;

    public void init() {
        msg="Bonjour de java !";
    }

    public void paint(Graphics g) {
        g.drawString(msg, 20, 20);
    }

}
```

Le code source java (2)

- L'affichage du texte, comme d'ailleurs tout affichage, se fait par l'intermédiaire d'une variable de type **Graphics** qui est ici fournie en paramètre dans la méthode paint. La classe Graphics, qui se trouve dans le package java.awt, fournit de nombreuses méthodes pour écrire du texte ou dessiner des figures géométriques.

Le fichier HTML

Il suffit de créer un fichier HTML minimal contenant la balise `<APPLET>`. Cette balise devra préciser au moins les trois attributs `CODE`, `WIDTH` et `HEIGHT`: Cela donne, par exemple, le fichier suivant :

```
<HTML>
```

```
  <BODY>
```

```
    <APPLET
```

```
      CODE="hello.class"
```

```
      WIDTH="300"
```

```
      HEIGHT="100"
```

```
    >
```

```
    </APPLET>
```

```
  </BODY>
```

```
</HTML>
```

Haut du formulaire

Bas du formulaire

- Enregistrer ce fichier en le nommant hello.html. On peut alors l'utiliser directement avec un navigateur.
- Il est aussi possible de passer par l'appletviewer qui se contentera d'afficher l'applet. Il faut alors entrer la commande : **appletviewer hello.html** dans une console DOS.

Utilisation de paramètres (1)

- De même qu'il était possible de passer des paramètres aux applications java, il est possible de le faire pour les applets.
- Cela se fait par l'intermédiaire du fichier html entre les balises `<APPLET>` et `</APPLET>`.
- On utilise la balise `<PARAM>` et ses deux attributs **NAME** et **VALUE**, ce qui permet d'attribuer à chaque paramètre un nom et une valeur.

Exemple : pour que le bonjour affiché par l'applet s'adresse à une personne déterminée, on utilisera un paramètre nommé "nom". Le fichier html sera complété de la façon suivante si l'on s'adresse à Mohammed.

<HTML>

<BODY>

<APPLET CODE="hello.class" WIDTH="300"
HEIGHT="100">

<PARAM NAME = "nom" VALUE ="Mohammed">

</APPLET>

</BODY>

</HTML>

Utilisation de paramètres (2)

- Evidemment l'applet devra récupérer le paramètre donné dans le fichier HTML pour en tenir compte.
- Elle dispose pour cela de la méthode **getParameter** qui attend en paramètre le nom du paramètre passé à l'applet (ce qui correspond à NAME) et qui renvoie sous forme de chaîne de caractères la valeur attribuée (ce qui correspond à Value).
- Dans le cas où le fichier HTML ne fait pas référence à un paramètre, la méthode **getParameter** renvoie la valeur **null** qui signifie qu'il n'y a pas eu réservation de mémoire pour la chaîne fournie en résultat.

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

```
public class hello extends JApplet {  
    String msg;
```

```
    public void init() {  
        msg="Bonjour de Java !!";  
        String parm=getParameter("nom");  
        if (parm!=null) msg=parm+", Java te dit bonjour !";  
    }
```

```
    public void paint(Graphics g) {  
        g.drawString(msg, 20, 20);  
    }
```

```
}
```


Différents stades de la vie d'une applet

Voici les différentes méthodes concernées par la vie d'une applet:

init(): Appelée automatiquement pour effectuer la première initialisation de l'applet, y compris la disposition des composants. Il faut toujours redéfinir cette méthode.

start(): Appelée chaque fois que l'applet est rendue visible du navigateur Web,. Appelée également après **init()**.

stop(): Appelée chaque fois que l'applet redevient invisible du navigateur Web, pour permettre à l'applet d'arrêter les opérations coûteuses. Appelée également juste avant **destroy()**.

destroy(): Appelée lorsque l'applet est déchargée de la page pour effectuer la libération finale des ressources lorsque l'applet n'est plus utilisée.

Restrictions imposées aux applets

- Les applets ont été conçues à l'origine pour être exécutées sur un site distant de celui qui a fourni le code.
- Dans ces conditions, les concepteurs de java avaient prévu des restrictions nécessaires pour assurer une sécurité absolue sur la machine d'exécution.

En particulier la JVM interdisait à une applet:

- d'accéder aux fichiers locaux
 - de lancer un programme exécutable local
 - d'obtenir des informations relatives au système local (autres que des informations banales tq: version de java utilisée,...)
-
- Toute tentative d'opérations de ce type provoquait une exception `SecurityException`.

Transformation d'une application graphique en une applet

- Tout d'abord il faut supprimer la méthode main car elle ne sera plus appelée lorsque le code sera lancé depuis un fichier HTML.
- Par ailleurs, il faut transformer l'objet fenêtre de type dérivé de JFrame créé par l'application (en général dans main) en un objet d'une classe dérivée de JApplet.
- Les actions réalisées dans le constructeur de la fenêtre principale de l'application seront reportées dans la méthode init de l'applet.
- Une exception aura lieu pour les appels à setSize, setBounds et setTitle qui ,n'ayant plus de raison d'être pour une applet, devront être supprimées.

Exemple

- Exécuter le code de l'exercice 1 de la série de TD-TP6

Interêt des applets

- Apporte de l'interaction et de l'animation dans les pages html en ajoutant aux pages web du code java téléchargé
- tout se télécharge
 - pas de configuration locale: évite d'installer, maintenir, mettre à jour l'application sur **tous les postes clients mais seulement sur le Serveur** Car Java est indépendant des plateformes ; une applet peut donc s'exécuter sur **tout poste client possédant la machine virtuelle Java**
- Utilisation de l'intranet
- Exécution sur le poste client donc moins d'encombrement du serveur
- **Mais**, Attention aux problèmes de sécurité

Exercice

- Transformer les applications graphiques de la série TD/TP 6 en applets