

Transmission d'arguments : projet

Palindrome

Introduction

Dans ce chapitre nous allons transmettre des arguments à une application et à une applet Java, c'est à dire leur donner des valeurs, chiffres ou lettres, qu'elles attendent pour leur bonne exécution. Par exemple, le programme doit savoir s'il y a plus ou moins de 100g de chocolats dans la boîte, il attendra alors un argument spécifié par l'utilisateur. Cet argument est susceptible de changer à chaque exécution du programme, c'est pourquoi il ne peut être directement spécifié dans le code source.

Capter un argument de ligne de commande est pénible en Java, heureusement c'est quasiment inutile car la plupart de vos programmes posséderont une Interface Utilisateur et ne se contenteront pas de s'exécuter dans une fenêtre de ligne de commande (DOS pour Microsoft...) (la procédure pour récupérer un argument à partir d'une IU étant beaucoup plus simple).

Par contre, pour les applets, c'est bien utile et moins compliqué.

Transmettre un argument à une application

Un argument se transmet à une application à l'exécution :

- // sans transmission d'argument
- java NomDuProgramme
- // avec transmission d'argument
- java NomDuProgramme PremierArgument SecondArgument TroisièmeArgument EtQuaetera

On note que les arguments sont séparés d'un espace. Dans le cas d'un argument qui comprend plusieurs mots qui doivent être séparés, on utilise les guillemets : " Federico Garcia Lorca " est considéré comme 1 seul argument.

Récupérer un argument à partir d'une application

On doit créer une boucle for qui parcourt le programme à la recherche d'arguments :

- //partie théorique
- for (int i = 0 ; i < arguments.length ; i++) {...}
-
- //partie pratique
- class MesArguments {
- public static void main(String args[]) {
- for (int i = 0 ; i < arguments.length ; i++) {
- System.out.println ("Arguments " + i + " : " + arguments[i]);
- }
- }
- }
- }

Le petit exemple présenté ci dessus affiche les arguments avec leur n°. Vous pouvez compiler ce programme, il fonctionne.

Transmettre un argument à une applet

On transmet un argument à une applet par l'intermédiaire du code HTML :

- < PARAM NAME = * VALUE = "*" >
- < PARAM NAME = font VALUE = "Arial" >

Les * symbolisent les valeurs que vous devez spécifier, la première le nom du paramètre et la seconde sa valeur.

Récupérer un argument à partir d'une applet

Pour récupérer à partir d'une applet les arguments transmis par le HTML, on utilise la méthode `getParameter()`.

- `TypeDeDonnée NomDeLaVariable = getParameter (" nomDuParamètre ");`
- `String theFont = getParameter(" font ");`

Applet : *Palindorme*

L'applet Palindorme vient du livre Java 2 plate-forme de Roger Cadenhead et Laura Lemay. Elle récupère et affiche le texte spécifié par ...

- `<PARAM NAME = palindrome VALUE = "Ecrivez ici ce que vous voulez" >`

```
import java.awt.Graphics; //importation de la classe qui gère les graphismes
import java.awt.Font; // "" qui gère les polices
import java.awt.Color; // "" qui gère les couleurs

public class Palindrome extends java.applet.Applet { // c'est une applet !
    Font f =new Font("TimesRoman", Font.BOLD, 36); // définition de la police f
    String palindrome; // déclaration de la chaîne "palindrome"

    public void paint(Graphics screen) { // méthode qui servira à fixer les
paramètres d'affichage
        screen.setFont(f); // attribuer la police f à tous els textes de l'applet
        screen.setColor(Color.red); // "" la couleur rouge
        screen.drawString(palindrome, 5, 50); // écrire le texte contenu dans
palindrome aux coordonnées ...
    }

    public void init() { // équivalent de main() dans les applications: le prog
débute
        palindrome = getParameter("palindrome"); // on récupère le paramètre
        if ( palindrome == null) // s'il n'y en a pas ...
            palindrome = "Spécifiez un argument, SVP"; // ... afficher ceci
    }
}
```

