

# Informatique en Tronc Commun

Faculté des Sciences de Nice  
DEUG 2000-2001

Jérôme DURAND-LOSE  
Sandrine JULIA  
Jean-Paul ROY

**COURS 6**

## Les Bibliothèques [API] de Java

Application  
Programming  
Interface

◆ Des centaines de classes sont disponibles pour aider le programmeur, par ex. la classe Math. Il existe des « paquetages » spécialisées pour les fichiers, Internet...

↳ *package = ensemble de classes*

◆ Pour le **graphisme**, on utilise le paquetage **AWT** ou son extension SWING.

Abstract  
Windowing  
Toolkit

2

## Trouver de la doc dans l'API est vital !

◆ Dans les livres, par ex. les « Java in a Nutshell ».

Explorer, Netscape, ...

◆ En ligne [gratuit], avec un « navigateur », en rapatriant la documentation de SUN sur le JDK 1.3

↳ *disponible sur les machines du MIPS*

3

The screenshot shows a Netscape browser window displaying the Java API Specification page. The browser's address bar shows the URL: <http://java.sun.com/products/jdk/1.2/docs/api/index.html>. The page title is "Java™ 2 Platform, Standard Edition, v1.2.2 API Specification". The main content area features a navigation menu with "Overview", "Package", "Class", "Use", "Tree", "Deprecated", "Index", and "Help". Below the menu, the text reads: "This document is the API specification of the Java 2 Platform, Standard Edition, version 1.2.2." A "See:" section is followed by a "Description" section. The "Description" section contains a table with the following entries:

Package	Description
<a href="#">java.applet</a>	Provides the classes necessary to create classes an applet uses to communicate context.
<a href="#">java.awt</a>	Contains all of the classes for creating 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 tran between and within applications.

## Qu'est-ce qu'une APPLLET ?...

- ◆ Conçu dès l'origine en direction d'Internet, Java entretient un rapport étroit avec les navigateurs.
- ◆ Une **applet** est une classe Java qui s'exécute dans une page d'un navigateur.
- ◆ Par exemple dans la fenêtre de votre *homepage* sur le Web, pour dialoguer avec l'internaute, lancer des calculs, dessiner dans la fenêtre, lui montrer des images, faire des animations, lui jouer un morceau de musique, etc.

5

Une applet !

en Java...

6

## Le programme source de l'applet

```
1 import java.awt.Color;
2 import java.awt.Graphics;
3 import java.awt.Font;
4 import java.applet.Applet;

5 public class AppletCours6 extends Applet
6 {   public void paint(Graphics g)
7     {   // on choisit la couleur du fond, ici du jaune (rouge + vert)
8         setBackground(new Color(255,255,0));

9         // on dessine un rectangle
10        g.drawRect(10,20,130,50);

11        // on affiche du texte en blanc
12        g.setColor(new Color(255,255,255));
13        g.drawString("Une applet !",30,40);

14        // du texte dans une autre couleur et un autre jeu de caractères
15        g.setColor(new Color(0,0,255));
16        g.setFont(new Font("Monospaced",Font.BOLD,36));
17        g.drawString("en Java...",30,100);
18    }
19 }
```

7

## Dissection de ce programme

- ◆ Attention, il ne montre pas *toutes* les facettes de la programmation d'une applet !

*pas de méthodes  
init, start, stop, ...*

- ◆ Il commence par **importer** des classes contenues dans des paquetages, par exemple :

java.awt.Graphics → la classe Graphics, qui fait partie du package java.awt

↓  
le sous-package **awt**  
du package java...

8

- ♦ Il importe en particulier la classe `java.applet.Applet` qui fournit les fonctionnalités minimales d'une applet [intégration dans le navigateur, méthodes : `paint (...)`, etc]
- ♦ L'applet va s'exécuter dans un rectangle situé dans la fenêtre-hôte du navigateur.
- ♦ La méthode `paint (...)` fournie par défaut dans la classe `Applet` est censée expliquer comment se redessine ce rectangle. Or c'est nous qui sommes maîtres du jeu !
- ♦ Nous allons donc **étendre** la classe prédéfinie `Applet` en une sous-classe qui redéfinira la méthode `paint (...)`.

9

## Etendre une classe ?

- ♦ Il s'agit là du concept d'**héritage**, compliqué, à la limite du DEUG !...
- ♦ On dit qu'une classe B est une **sous-classe** de la classe A [ou qu'elle **étend** A, ou qu'elle **dérive** de A, ou qu'elle **hérite** de A] si toutes les méthodes (et champs) de A sont disponibles dans B, avec cependant la possibilité dans B d'en rajouter d'autres ou d'en redéfinir certaines.
- ♦ Ceci permet de profiter d'une classe existante [par ex. dans l'API] fournissant de bonnes fonctionnalités, en l'*étendant* un peu pour les besoins en cours.
- ♦ Fondamental en Programmation par Objets !...

10

- ♦ Une applet n'a pas de méthode `main (...)`. C'est le navigateur qui va lancer l'applet lors du chargement de la page HTML.
- ♦ La ligne 5 déclare une nouvelle classe héritant de la classe `Applet` et redéfinissant la méthode `paint (...)` :

```
public class AppletCours6 extends Applet
{
    public void paint(Graphics g)
    {
        . . . . .
    }
}
```

11

## On dessine dans un « contexte graphique »

- ♦ La méthode `paint (...)` reçoit comme argument un « contexte graphique » `g` :

Graphics g

- ♦ Un **contexte graphique** renferme les informations nécessaires aux opérations de dessin procurées par Java [couleur du crayon, épaisseur des lignes, police de caractères utilisée, etc].
- ♦ C'est le navigateur qui envoie le message `paint (g)` à l'applet en lui passant le contexte graphique `g`.

12

♦ La ligne 8 envoie un message à l'objet courant [l'applet, instance de la classe `AppletCours6`] et lui demande de régler sa couleur d'arrière-plan [le fond].

*voir l'API, doc de la classe `Applet` !...*

## Et les méthodes de la classe `Graphics`...

♦ A partir de là, on s'adresse au contexte graphique `g`, en lui envoyant des messages pour changer la couleur du crayon, dessiner un rectangle, dessiner (!) du texte, changer de police de caractères, etc.

*voir l'API, doc de la classe `Graphics` !...*

13

## Comment lancer l'applet ?

♦ Il faut dire au navigateur, dans le source HTML de la page à afficher, quelle est la classe à charger [code], ainsi que les dimensions [width, height] du cadre occupé par l'applet dans la page.

♦ Pour dialoguer, il n'est pas possible d'invoquer la classe `Console`. Une « vraie » applet utilisera toute la puissance de l'AWT pour construire une **interface utilisateur graphique** [boutons, zones de texte à remplir, etc]. Mais c'est bien plus difficile...

*GUI : Graphical  
User Interface*

14

## Le texte source de la page HTML

```
<html>
<title>AppletCours6</title>
<h1>Un exemple d'<i>applet</i> avec Java</h1>
<center>
  <applet code="AppletCours6.class" width=300 height=120>
  </applet>
</center>
<h1>Je veux voir le source !...</h1>
Cliquez sur ce <a href="AppletCours6.java">bouton</a> pour
voir comment est programmée cette <i>applet</i>...
</html>
```

15