PROGRAMMATION D'APPLETS JAVA

pascal.nicolas@univ-angers.fr

applet

- programme Java
  - résidant sur un serveur web
  - référencé dans une page web
  - téléchargé sur le navigateur qui accède à la page web
  - exécuté par le navigateur
  - peut accéder uniquement à son serveur web d 'origine, sauf configuration spéciale des paramètres de sécurité (applet signées)

exemple de page web référençant une applet

```
<HTML>
<HEAD><TITLE> test du compteur</TITLE></HEAD>
<BODY>
<APPLET
        CODE="Compteur.class"
        WIDTH=500 HEIGHT=500
        ALIGN=middle>
</APPLET>
</BODY>
</HTML>
```

On peut aussi ajouter des couples

```
<PARAM NAME = "X" VALUE = "toto">
```

pour passer des paramètres à l 'applet à l 'intérieur de laquelle on les récupère grâce à getParameter("X")

programmation

- dériver la classe java.applet.Applet
- définir les méthodes contrôlant l'exécution
  - init() appelée à la création de l'applet
  - start() appelée lorsque la page web est (re)visitée
  - stop() appelée lorsque le navigateur quitte la page web
  - destroy() appelée lors de la destruction de l'applet (fin de session du navigateur)
- cycle : init() (start() stop())* destroy()

composants d'une interface utilisateur avec Abstract Window Toolkit

- java.awt.Component
  - java.awt.Button (bouton cliquable)
  - java.awt.Canvas (zone de dessin)
  - java.awt.Checkbox (boite à cocher)
  - java.awt.Choice (liste de choix, pop-up menu)
  - java.awt.Container (conteneur d 'autres objets de AWT)
    - java.awt.Panel (le conteneur le plus simple)
      - java.applet.Applet (une applet est un conteneur)
    - java.awt.ScrollPane (un conteneur avec des barres de défilement)
  - java.awt.Label (zone de texte non modifiable par l'utilisateur)
  - java.awt.List (liste de choix, à sélection multiple)
  - java.awt.Scrollbar (ascenseur)
  - java.awt.TextComponent (composants de type texte)
    - java.awt.TextArea (zone d 'édition de texte de plusieurs lignes)
    - java.awt.TextField (zone d 'édition de texte d 'une seule ligne)

placement des composants

- Selon un style défini par un LayoutManager
  - BorderLayout (selon 5 zones : centre, nord, sud, est, ouest)
  - CardLayout (pile de fiches)
  - FlowLayout (rangement ligne par ligne, de gauche à droite, au fur et à mesure des ajouts)
  - GridLayout (dans un tableau à 2 dimensions)
  - GridBagLayout (permet d'aligner les composants horizontalement et verticalement quelle que soit leur taille)

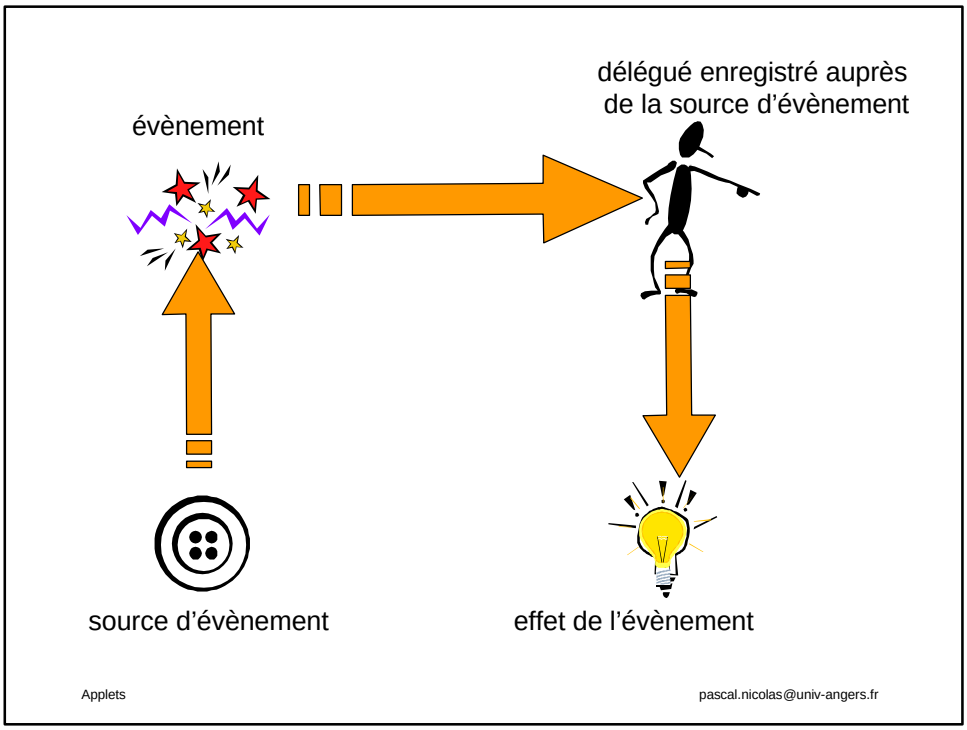
classes d'évènements

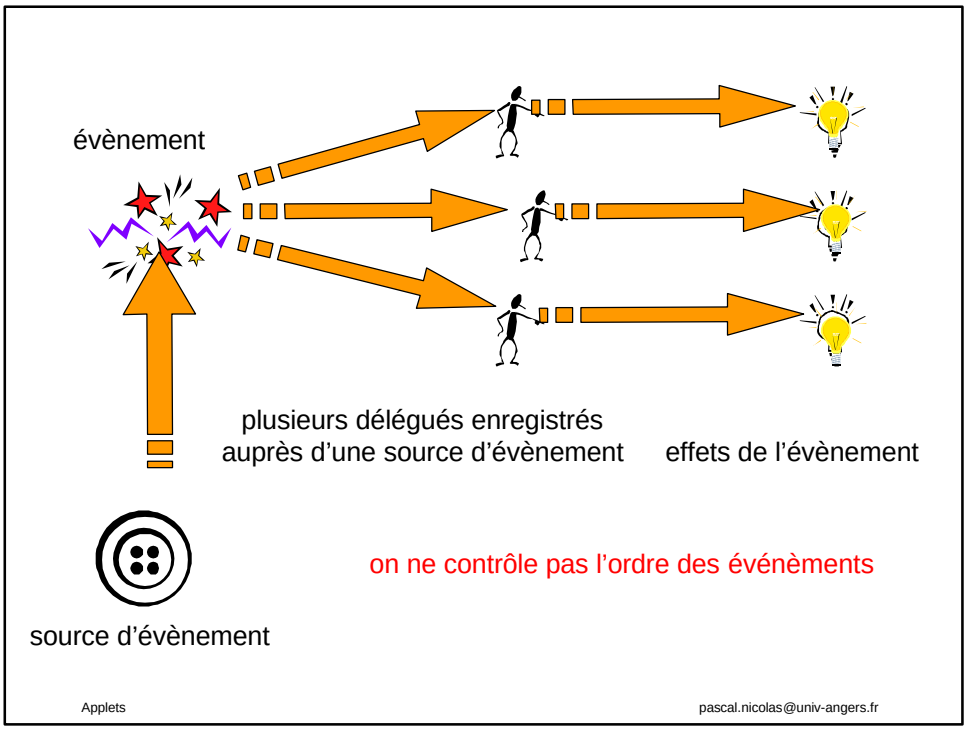
- java.awt.AWTEvent
  - java.awt.event.ActionEvent (clic sur les boutons, les menus, choix dans les listes, touche <enter> dans un champ texte)
  - java.awt.event.AdjustmentEvent (modification des objets ajustables, ex: barre de défilement)
  - java.awt.event.ComponentEvent
    - java.awt.event.ContainerEvent (ajout ou suppression d 'un composant)
    - java.awt.event.FocusEvent (entrée et sortie d 'un composant à l 'aide de la souris ou de la touche <tab>)
    - java.awt.event.InputEvent
      - java.awt.event.KeyEvent (utilisation de touches du clavier)
      - java.awt.event.MouseEvent (clic de souris)
    - java.awt.event.WindowEvent (iconification, activation, ouverture, fermeture de fenêtres)
  - java.awt.event.ItemEvent (sélection et déselection d 'un choix)
  - java.awt.event.TextEvent (modification du contenu d 'un composant texte)

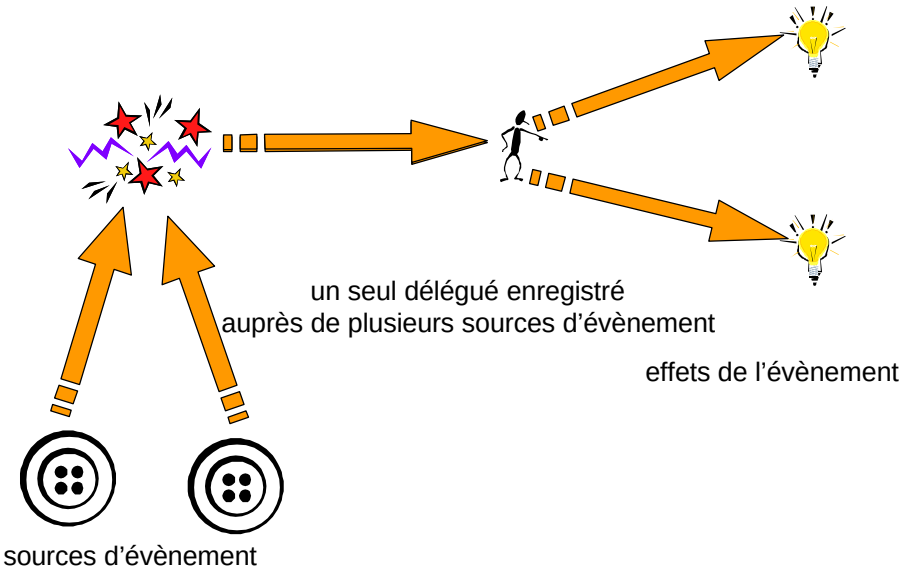
gestion des évènements

classe d'évènement xxxEvent ⟷ interface xxxListener

- définir une classe délégué qui implémente l'interface adéquate en définissant des méthodes pour récupérer et traiter les évènements
- créer un (des) objet(s) de cette classe
- enregistrer cet (ces) objet(s) auprès de(s) composants à surveiller, à l'aide de addxxxListener(xxxListener)


délégué enregistré auprès
de la source d'évènement
évènement
source d'évènement
effet de l'évènement
Applets
pascal.nicolas@univ-angers.fr


évènement
plusieurs délégués enregistrés
auprès d'une source d'évènement
effets de l'évènement
source d'évènement
on ne contrôle pas l'ordre des événèments
Applets
pascal.nicolas@univ-angers.fr



méthodes des délégués

- ActionListener
  - actionPerformed(ActionEvent)
- MouseListener
  - mouseClicked(MouseEvent)
  - mouseEntered(MouseEvent)
  - mouseExited(MouseEvent)
  - mousePressed(MouseEvent)
  - mouseReleased
- MouseMotionListener
  - mouseDragged(MouseEvent)
  - mouseMoved(MouseEvent)
- KeyListener
  - keyPressed(KeyEvent)
  - keyReleased(KeyEvent)
  - keyTyped(KeyEvent)
- ItemListener
  - itemStateChanged(ItemEvent)
- ...

exemple d'applet

visualisable sur : http://www.info.univ-angers.fr/pub/pn/Applets/compteur.html

```
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;

public class Compteur extends
    java.applet.Applet {

// des variables de classe
int Compteur = 0;
Font police = new
    Font("Helvetica",Font.BOLD,24);
Button Bplus, Bmoins;
Label Lab_compteur;

public void init() {
// on fixe la couleur de fond
setBackground(Color.blue);
// et la police d'affichage
setFont(police);
```

```
/* on crée une grille à 3 lignes et 1
    colonne */
setLayout(new GridLayout(3,1));
/* on crée le panneau de chaque ligne
    et on l'ajoute à la page */
Panel ligne1 = new Panel();
add(ligne1);
Panel ligne2 = new Panel();
add(ligne2);
Panel ligne3 = new Panel();
add(ligne3);

/* on crée les boutons et on les pose
    dans la ligne 1*/
Bplus = new Button("+");
Bplus.setBackground(Color.lightGray);
Bplus.setForeground(Color.red);
ligne1.add(Bplus);
Bmoins = new Button("-");
Bmoins.setBackground(Color.lightGray);
Bmoins.setForeground(Color.green);
ligne1.add(Bmoins);
```



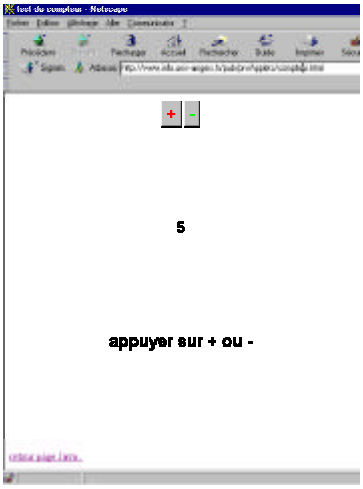
```
/* on remplit les autres lignes */
/* les espaces autour du compteur
    servent à réserver suffisamment de
    place pour des grands nombres */
 Lab_compteur = new
    Label('  '+String.valueOf(Compteur)
    + '  ' , Label.CENTER);
// on pose le compteur
 ligne2.add(Lab_compteur);
// et les explications
 ligne3.add(new Label("appuyer sur + ou
    -", Label.CENTER));

// on crée et on enregistre le délégué
// auprès des 2 boutons
 EcouteBouton EB= new EcouteBouton();
 Bmoins.addActionListener(EB);
 Bplus.addActionListener(EB);
} //fin de la méthode init
```

```
// classe interne d 'implémentation
// du délégué
class EcouteBouton implements
    java.awt.event.ActionListener {
public void actionPerformed (ActionEvent e) {
// évolution du compteur en fonction
// du bouton cliqué
if (e.getSource() == Bmoins)
 Compteur--;
else
 Compteur++;
Lab_compteur.setText(String.valueOf(Compteur);
 }
}

}
```

aspect de l'applet compteur



dessiner

- Les dessins dans une applet se font à l'aide de paint(Graphics G).
- G est un contexte graphique (la zone à dessiner, par défaut l 'applet toute entière)
- Les objets de Graphics possèdent des méthodes de dessin de texte, de points, de lignes, de rectangles, etc…
- repaint() (applicable aux canvas, panel et applet) va appeler update(Graphics G) qui va recevoir de la machine virtuelle le contexte G, dessiner un rectangle de la couleur du fond sur toute la zone à redessiner et appeler paint(Graphics G).
- Si on redéfinit update l'effacement n'aura plus lieu