

# TP2: JavaBeans

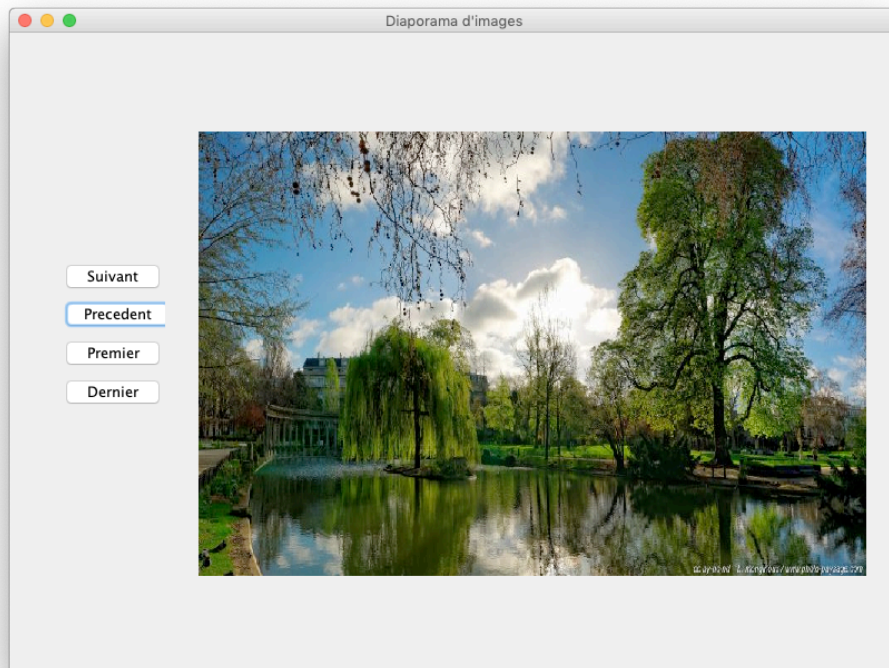
On se propose de développer une application basée sur des javabeans afin de visualiser des images contenues dans un répertoire.

- a) Définir un javabean capable d'afficher une image dont le nom du fichier est dans un premier temps codé en dur dans le programme source java. Tester ce javabean en incluant le composant dans le « contentPane » d'une JFrame.

Rappel: une image peut être chargée par la méthode :

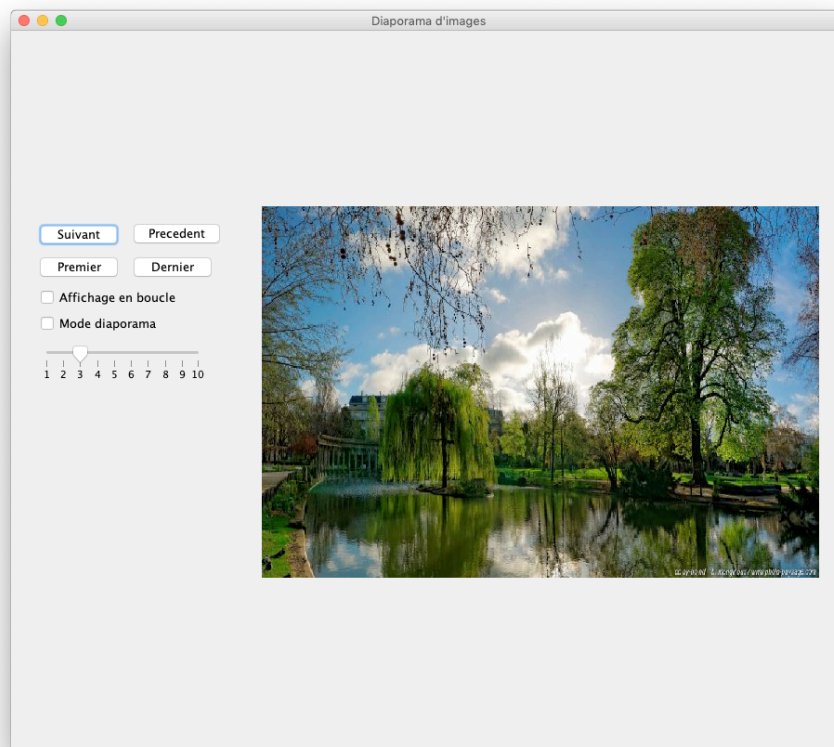
*Image Toolkit.getDefaultToolkit.createImage (String) ;*

- b) Le javabean doit désormais être capable d'afficher toutes les images au format gif ou jpeg contenues dans un répertoire dont le nom est pour l'instant codé en dur dans le fichier source java (on utilisera la classe *File* et l'interface *FilenameFilter* pour générer la liste de tels fichiers au sein d'un répertoire); on vérifiera que le nom spécifié désigne bien un répertoire. Ajouter les méthodes *void premier()*, *void dernier()*, *void suivant()*, *void precedent()*. Tester cette nouvelle version du javabean en connectant des composants *javax.swing.JButton* afin de pouvoir faire défiler les images comme sur la figure ci-dessous:



- c) Ajouter au javabeau les propriétés suivantes:
- . Une propriété de type booléen pour indiquer si les images doivent être affichées successivement sans intervention de l'utilisateur (diaporama).
  - . Une propriété de type booléen pour indiquer si la séquence d'images doit boucler ou non.
  - . Une propriété de type entier pour indiquer le temps en secondes entre deux images si la propriété diaporama est positionnée à la valeur vraie.

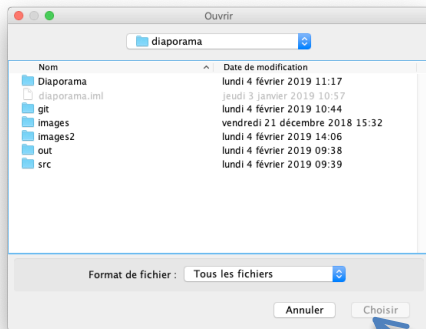
Modifier l'interface de l'application pour inclure des objets de type *javax.swing.JCheckbox* (pour le mode boucle et diaporama) et un composant de type *javax.swing.JSlider* pour la temporisation comme sur la figure ci-dessous:



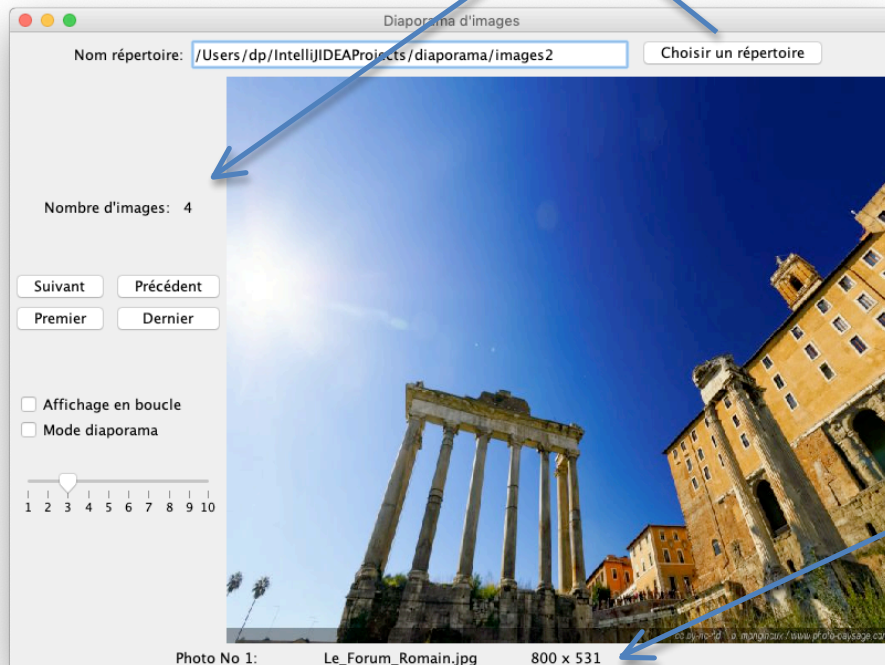
- d) Ajouter une propriété de type chaîne de caractères pour choisir le répertoire contenant les images à afficher. Le nom du répertoire pourra être saisi au clavier à travers un composant *javax.swing.JTextField* mais aussi être sélectionnée avec un composant *javax.swing.JFileChooser*. Un changement de valeur de cette propriété doit avoir pour effet de recréer la liste d'images et d'afficher dans un javabeau "beanTotalImages", étendant la classe *javax.swing.JLabel*, le nombre d'images contenues dans ce répertoire; on utilisera ici une propriété liée.

- e) On souhaite désormais qu'un événement **multicast** de type *ImageChangedEvent* soit généré à chaque changement d'image; l'événement généré devant permettre de récupérer le numéro, le nom, la largeur et la hauteur de l'image affichée. Modifier le javabean afin qu'il génère un tel événement. On créera un javabean "BeanInformations", héritant de la classe *javax.swing.JLabel*, capable d'afficher les informations. Le résultat obtenu pourra ressembler à la figure ci-dessous:

*JFileChooser*



BeanTotalImages notifié par  
l'événement  
PropertyChangeEvent



BeanInformations  
notifié par  
l'événement  
*ImageChangedEvent*