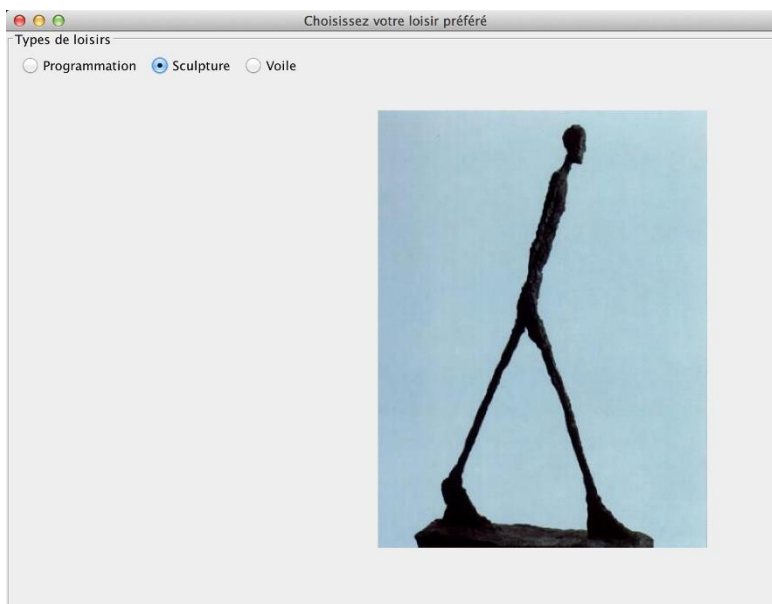


Programmation d'IHM – Mise en Pratique : Loisirs - Layouts

Démarre en Séance 1, à finir pour début de séance 2

L'objectif de cet exercice est de **programmer une interface H/M à la main**, sans l'aide de l'IDE. Vous allez développer une interface composée d'une fenêtre (JFrame) avec 3 radios boutons et une image. On demande à l'utilisateur son loisir préféré, et selon sa réponse, une image pré stockée s'affiche, comme le montre l'exemple suivant.



(Vous pouvez choisir des photos et d'autres loisirs).

Travail à faire

- Définir une 1^{ère} fenêtre qui dérive d'une *JFrame* et récupérer son *contentPane* ;
- Ajouter au *contentPane* les 3 radios boutons et l'image. L'image est un *JLabel* auquel on associe une image en envoyant le fichier au constructeur : `new JLabel(new ImageIcon("src/gif/monFichier.gif"))`; pour un fichier dans un répertoire gif dans *source*.
- **Tester 4 façons différentes** de disposer les composants sur la fenêtre (cf suggestions ci-dessous) : copier / coller 3 fois votre *JFrame* dans le package *source* et le modifier pour voir les différences de *layout*. La gestion des événements ne changera pas. Votre *main ()* testera chaque interface.
- Rédigez un court compte-rendu de TP expliquant vos choix et vos observations des différences pour la prochaine séance. Travail en binôme possible mais individuel préféré.
- Il y aura un corrigé en séance TD2.

Détails techniques

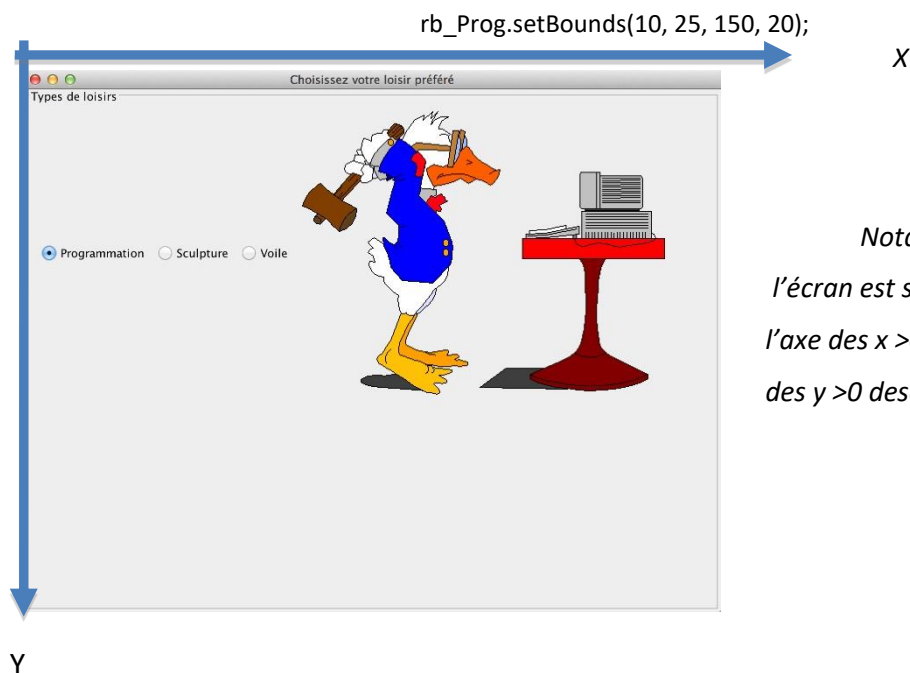
- Définir la JFrame, et décrire les composants (radio boutons, label image) comme des champs de classe.
- Dans le constructeur de la fenêtre, donner un titre et une taille à la fenêtre, définir les valeurs aux champs et les ajouter aux conteneurs comme vus dans les exemples du cours.
- Choisir les layouts qui donneront 4 rendus différents : on suggère un flowLayout, un BorderLayout, un GridLayout et un *positionnement absolu*. Notez les différences notamment lorsqu'on redimensionne la fenêtre.
- Définir un radio bouton et spécifier qu'il sera coché par défaut : `rb_Prog = new JRadioButton("Programmation", true);`
- Pour gérer l'exclusivité des radios boutons (ici on demande LE loisir préféré), il faut les ajouter à un même groupe de boutons, appelé ButtonGroup :

```
ButtonGroup groupeBoutons.add(rb_Prog);
```

- Pour avoir un panneau avec une bordure et un titre comme dans l'exemple, faire :

```
lePanneau.setBorder(BorderFactory.createTitledBorder("Types de loisirs"));
```

- Pour positionner en absolu un radio bouton, on donne les coordonnées par ex. en pixels x,y du point en haut à gauche du rectangle encadrant le radio bouton et son texte, la largeur puis les hauteur de ce rectangle, par ex :



Nota : le référentiel de l'écran est situé en haut à gauche, l'axe des $x > 0$ part à droite et celui des $y > 0$ descend vers le bas.

- Les événements à écouter pour les radios boutons sont des instances d'ItemEvent et on ajoute un ItemListener à chaque radio bouton. **Etudier la documentation de l'API Java** pour savoir le nom de **l'unique méthode** à réécrire pour faire le changement sur l'interface, dans l'écouteur : à chaque action sur les radio boutons, on va modifier l'image affichée. Utiliser :
`lbl_image.setIcon(new ImageIcon("src/gif/voiliers.jpg"));`
- Implémenter l'ItemListener de la façon qui vous convient le mieux (cf cours, on a vu 2 façons d'implémenter les écouteurs).