

M1 Programmation avancée et répartie en Java : TP 5

Jean Fortin et Frédéric Gava
gava@u-pec.fr ou jean.fortin@gmail.com

1 Exercice 1 : une Applet Java

L’invocation de la méthode `paint(Graphics g)` assure la réactualisation du contexte graphique de l’applet. Elle sera appelée à chaque fois qu’il est nécessaire de redessiner le contenu de l’applet. C’est cette méthode que nous allons surcharger pour faire de “beaux” dessins.

1. Créer une classe `FigureGraphique` qui étend la classe `Applet`. Vous redéfinirez la méthode `paint` : celle-ci devra afficher le texte `Hello World!` à l’aide de la méthode `drawString` de la classe `Graphics` (faire `import java.awt.Graphics`). Compilez.
2. Créez un fichier d’extension `html` contenant le code permettant d’invoquer l’applet `FigureGraphique`. Exemple :

```
<HTML>
  <HEAD>
    <TITLE> Ma page web </TITLE>
  </HEAD>
  <APPLET CODE=FigureGraphique.class WIDTH=400 HEIGHT=400> </APPLET>
</HTML>
```

3. Vous pouvez évaluer vos qualités artistiques en exécutant l’applet avec `appletviewer` ou `firefox`. Dans le cas où vous utilisez un navigateur, forcez le rechargement de l’image (F5) car des paramètres de cache peuvent vous empêcher de voir les différences entre 2 exécutions de l’applet.
4. Écrivez une classe `Figure` et 2 héritières : `Carre` et `Rectangle`.
5. Cherchez dans la doc de `Graphics` des méthodes qui vous permettront de dessiner des carrés et des rectangles de couleur. Modifiez votre classe `FigureGraphique`. Un attribut de vecteur de `Figure` sera initialisé avec des carrés et des rectangles. La méthode `paint` permettra de dessiner les `Figure` du vecteur.
6. Pour comprendre comment disposer vos figures dans une fenêtre, faites varier les valeurs des différents paramètres de taille et de position de vos figures et observez les résultats obtenus.

2 Exercice 2 : Applet Java et Swing

Une applet java peut être programmée en utilisant la classe `JApplet`, au lieu de `JFrame`. Au lieu de la méthode `main`, les méthodes suivantes contrôlent le déroulement de l’applet :

```
- public void init() { ... }
```

- `public void start() { ... }`
- `public void stop() { ... }`
- `public void destroy() { ... }`

Ces méthodes peuvent être surchargées par le programmeur, et sont appelées automatiquement. Ainsi, la définition des contrôles/layouts présents dans la JApplet peut se faire dans la méthode `init`.

1. Créer une applet Java, comprenant une zone de texte et un bouton. Un clic sur le bouton devra modifier le texte affiché.
2. Lancer l'applet dans le viewer.
3. Créer une page html, comprenant une balise `<APPLET>` pour intégrer l'applet dans la page web. (voir page 304 des notes du cours pour la syntaxe de cette balise).

3 Exercice 3 : des codes JNI, des méthodes natives

Les codes C sont réputés pour la vitesse d'exécution notamment pour le calcul numérique.

1. Écrire une méthode native qui calcul la soustraction de deux entier et lève une exception si l'entier résultant est inférieur à 0.
2. Écrire une méthode native qui calcul la transposée d'une matrice de floats. 2 manières.
 - (a) la transposée remplace la matrice donnée
 - (b) on retourne une nouvelle matrice transposée de celle donnée en paramètre