

Travaux dirigés – séance n°2 : Programmation événementielle et architecture MVC

Il manque un modèle à l'applet à laquelle on est arrivé !

Dans la documentation en ligne de java (<http://docs.oracle.com/javase/7/docs/api/>), chercher **List**. Cette interface spécifie une collection à laquelle on peut ajouter (add) un élément (de type E) et obtenir un élément par son numéro (get). Chercher la classe ArrayList.

On spécifie un dessin (classe Dessin) comme une collection de traces (classe Trace).

```
public class Dessin {  
    ...  
    public Dessin() { ... }  
    public void ajoute(Trace trace) { ... }  
    public int nbTraces() { ... }  
    public Trace element(int numero) { ... }  
}  
public class Trace {  
    ...  
    public Trace(int xDebut, int yDebut) { ... }  
    public void ajoute(int x, int y) { ... }  
    public int nbPoints(){ ... }  
    public Point element(int numero) { ... }  
}
```

Examiner ces classes dans le projet « Applet de dessin v1 avec modèle » pour en comprendre le fonctionnement.

Que faut-il ajouter à la classe **AppletDessin** et aux méthodes **mousePressed** et **mouseDragged** pour mémoriser le dessin ?

Quelle pourrait être la méthode **public void paint(Graphics g)** qui reproduit la totalité du dessin ?

Certaines réactions aux événements ont des répercussions sur la vue, d'autres sur le modèle. Ces réactions devraient être définies dans un écouteur.

En partant du projet « Applet de dessin v1 », définir un écouteur **EcouteurCrayon** qui implante les interfaces **MouseListener** et **MouseMotionListener** tel que le fait l'applet **AppletDeDessin** et modifier le code de **AppletDeDessin** pour qu'elle utilise cet écouteur.

Cette nouvelle version sera nommée « Applet de dessin v1 avec 1 outil ».

Intégrer à cette version le modèle pour obtenir une version « Applet de dessin v1 avec 1 outil et modèle ». Attention, la vue doit connaître le modèle et le contrôleur, le contrôleur doit connaître le modèle et la vue...

En partant de la version « Applet de dessin v1 avec 1 outil », définir un autre écouteur **EcouteurEtoile** qui implante les mêmes interfaces et modifier le code de **AppletDeDessin** pour qu'elle utilise cet écouteur. Cet écouteur permet de dessiner des segments rayonnant à partir du point d'enfoncement du bouton de la souris.

Travaux dirigés – séance n°2 : Programmation événementielle et architecture MVC

On convient qu'une frappe de la touche c fait passer l'applet à l'écouteur crayon et qu'une frappe de la touche e la fait passer à l'écouteur étoile. Implanter l'interface `KeyListener` dans `AppletDeDessin` pour permettre le changement d'outil.

Pourquoi l'ajout de l'écouteur étoile est-il sans effet ? Que faut-il faire pour que cela fonctionne ?

Compte-tenu des points communs entre `EcouteurCrayon` et `EcouteurEtoile`, proposer un héritage avec une classe de base commune à définir. Simplifier la classe `AppletDeDessin` grâce à cet héritage.