

**Document de travail pour le TP4**

**NE PAS IMPRIMER CE DOCUMENT**  
il va être modifié et complété

## Gift wrapping (préparation)

La structure conseillée pour stocker les points est un tableau *2 lignes-n colonnes* : à l'indice  $i$ , la première ligne contient la coordonnée en  $x$  et la seconde ligne la coordonnée en  $y$  d'un même point qui sera nommé par son indice dans le tableau ( $i$ ).

Construire une classe avec :

- une **constante**, `maxAleat`, qui sera la valeur maximum des coordonnées de points tirées aléatoirement ;
- une **méthode d'initialisation** de l'ensemble des points. Le nombre de points à créer est passé en paramètre. Les 3 premiers points sont fixes et occupent respectivement les extrémités gauche et droite en bas ainsi que le centre haut du nuage de points, voir ci-dessous. Ces points sont stockés aux indices 1, 2 et 3 du tableau de points. Les autres points sont choisis aléatoirement **sans répétition**<sup>1</sup>. Le nombre d'éléments à créer doit être supérieur ou égal à 4.



- une **méthode de recherche d'un point** (donné par ses coordonnées  $x$  et  $y$ ) dans le tableau de points ;
- une **méthode d'affichage textuel** simple (un point par ligne : indice, coordonnées  $x$  et  $y$ ) d'un ensemble de points. Vous devez avoir en paramètre le tableau et son indice maximum d'utilisation ;
- une **méthode d'affichage graphique** sur la base de la classe `Frame` de Java. Vous pouvez utiliser le schéma de programme, *2013-pourGiftWrapping.schema*, disponible sur *ufrsciencestech*. Vous trouverez dans ce schéma de programme une classe `giftWrapping` ainsi qu'une classe `pourFermer` qui permet d'utiliser le bouton de fermeture de la fenêtre graphique. Voir Annexe ;

<sup>1</sup> Une des hypothèses faites en TD : il n'y a pas deux occurrences du même point dans le tableau.

## Gift wrapping (par minimalisation des angles polaires)

Divers outils trigonométriques ( $\Pi$ ,  $\sin$ ,  $\cos$ ,  $\tan$  et leurs réciproques  $\arcsin$ ,  $\arccos$ ,  $\arctan$ , ...) sont disponibles dans la classe *Math* : par exemple *Math.PI*, *Math.asin(x)*<sup>2</sup>. Vous trouverez la documentation correspondante à l'adresse <http://docs.oracle.com/javase/1.4.2/docs/api/java/lang/Math.html/>

*Attention aux cas où ces opérations ne sont pas définies  
(par exemple division par zéro sur les calculs de tangente et cotangente)*

Vous devez ajouter à votre classe, les méthodes suivantes :

- calcul du **suivant droit** d'un point sur la demi-enveloppe convexe droite du nuage de points ;
- calcul du **suivant gauche** d'un point sur la demi-enveloppe convexe gauche du nuage de points ;
- détermination de la **demi-enveloppe convexe droite** du nuage de points. Le point de départ est `DEBD`, le point final est `HAUT` (indices respectifs 2 et 3 dans le tableau de points) ;
- détermination de la **demi-enveloppe convexe gauche** du nuage de points. Le point de départ est `DEBG`, le point final est `HAUT` (indices respectifs 1 et 3 dans le tableau de points) ;
- **recopie séquentielle** d'un tableau de points dans un autre (pour reconstruire l'enveloppe convexe globale). L'indice de début dans le tableau destination ainsi que le nombre d'éléments du tableau source doivent être passés en paramètre. Vous devez renvoyer comme résultat l'indice du dernier élément dans le tableau destination ;
- **recopie séquentielle inverse** d'un tableau de points dans un autre (pour reconstruire l'enveloppe convexe globale). Le tableau source est utilisé du dernier au premier indice. L'indice de début dans le tableau destination ainsi que le nombre d'éléments du tableau source doivent être passés en paramètre. Vous devez renvoyer comme résultat l'indice du dernier élément dans le tableau destination ;

Vous devez suivre l'algorithme ci-dessous pour l'ensemble du traitement :

```
initialisation de l'ensemble de points PTS (indices de 1 à max)
initialisation de l'ensemble de booléens dispo (indices de 1 à max)
calcul de la demi-enveloppe convexe droite ECD (indices de 1 à nbECD)
calcul de la demi-enveloppe convexe gauche ECG (indices de 1 à nbECG)
construction de l'enveloppe globale ENV (points DEBG, DEBD et tableaux ECD, ECG)
attention : le point HAUT doit apparaître une seule fois dans ENV
affichage du nuage de points et de l'enveloppe globale
voir 2013-pourGiftWrapping.schema sur ufrsciencetech
```

Extension possible par rapport au TD Vous pouvez éliminer la génération séparée des points `DEBG`, `DEBD` et `HAUT`. Il faut alors :

- calculer les points d'ordonnée maximale et déterminer le plus à gauche et le plus à droite : ces deux points extrêmes remplaceront le point `HAUT` dans les méthodes de calcul des demi-enveloppes convexes. Tous les points intermédiaires trouvés seront marqués non disponibles ;
- calculer les points d'ordonnée minimale et déterminer le plus à gauche et le plus à droite : ces points remplaceront `DEBG` et `DEBD` dans les méthodes de calcul des demi-enveloppes convexes.

<sup>2</sup> Voir annexe.

- Tous les points intermédiaires trouvés seront marqués non disponibles ;
- vérifier la méthode de construction de l'enveloppe convexe globale sur les points en haut et en bas (selon le nombre de points trouvés, un ou plusieurs).

## Annexe

Vous trouverez dans la classe Math les constantes et opérations suivantes (voir documentation sur <http://docs.oracle.com/javase/1.4.2/docs/api/java/lang/Math.html/>) :

```
static double    PI
static double    acos(double a)
static double    asin(double a)
static double    sqrt(double a)
```

Vous trouverez dans le schéma de programme *2013-pourGiftWrapping.schema* disponible sur *ufrsciencestech*, les éléments suivants pour la gestion du graphique<sup>3</sup> :

- pour pouvoir utiliser le bouton de fermeture de la fenêtre graphique :
  - import de la bibliothèque `java.awt.event.*`
  - définition d'une classe `pourFermer` contenant une méthode `windowClosing()` qui indique comment réagir au clic sur le bouton de fermeture : en appelant la méthode `System.exit()`
  - lancement d'une méthode de surveillance des clics sur la fenêtre graphique `addWindowListener()` qui va appeler la méthode `windowClosing`

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

class pourFermer extends WindowAdapter
{
    public void windowClosing ( WindowEvent e )
    {
        System.exit(0);
    }
} // fin de la classe

// dans la méthode main avant de lancer l'affichage
// pour gérer la fermeture de la fenêtre
gW.addWindowListener(new pourFermer() );
```

- pour dessiner les points (sous forme de petits rectangles, plus visibles) :
  - méthode `fillRect( x, y, largeur, hauteur)`
- pour dessiner les segments qui constituent l'enveloppe convexe :
  - méthode `drawLine( x1, y1, x2, y2 )`

<sup>3</sup> La gestion du bouton de fermeture de la fenêtre graphique est mise en place automatiquement sous Netbeans.