

Exercice 1 – Créer des threads, visualiser le parallélisme

Les applications graphiques permettent de mettre en évidence l'exécution concurrente de tâches. Pour ces applications, nous utiliserons un paquetage `graphic` minimal basé sur Swing, qui permet d'afficher dans une fenêtre des points et des lignes. Les caractéristiques de la classe `Window` du paquetage `graphic` sont données dans la Figure 1.

Constructor Summary	
Constructors	
Constructor and Description	
<code>Window(int width, int height, java.lang.String title)</code>	Creates a window with dimensions width x height, a title title and no background color
<code>Window(int width, int height, java.lang.String title, java.lang.String color)</code>	Creates a window with dimensions width x height, a title title and a background color color

Method Summary	
Methods	
Modifier and Type	Method and Description
void	<code>drawLine(java.awt.Point pt1, java.awt.Point pt2)</code> Draws a black line between points pt1 and pt2
void	<code>drawLine(java.awt.Point pt1, java.awt.Point pt2, java.lang.String color)</code> Draws a line of color color between points pt1 and pt2
void	<code>drawPoint(java.awt.Point pt)</code> Draws point pt in black
void	<code>drawPoint(java.awt.Point pt, java.lang.String color)</code> Draws point pt with color color
void	<code>fill(java.lang.String color)</code> Fills the background of the window with color color
void	<code>plotLine(java.awt.Point pt1, java.awt.Point pt2)</code> Plots a black line from point pt1 to point pt2
void	<code>plotLine(java.awt.Point pt1, java.awt.Point pt2, java.lang.String color)</code> Plots a line of color color from point pt1 to point pt2

FIGURE 1 – La classe `Window`

Le paquetage `graphic` est localisé dans le répertoire ci-dessous :

```
/Infos/lmd/2020/licence/ue/LU3IN001-2020oct/TME
```

Pour utiliser la bibliothèque avec une compilation en ligne de commande, vous devez ajouter ce répertoire à la variable `CLASSPATH`. Pour cela, ajoutez les commandes ci-dessous au fichier `.bashrc` qui se trouve dans votre répertoire de connexion (créez-le s'il n'existe pas) :

```
CLASSPATH=$CLASSPATH:/Infos/lmd/2020/licence/ue/LU3IN001-2020oct/TME
export CLASSPATH
```

Il faut ensuite exécuter les commandes de `.bashrc` à partir d'un terminal :

```
source .bashrc
```

Pour utiliser la bibliothèque avec *Eclipse*, il faut accéder aux propriétés du projet (clic droit), puis

Java Bulid Path > Libraries > External Class Folder

ajouter le dossier `TME` dont le chemin est donné ci-dessus, exécuter `Apply` puis `Close`.

Question 1

Pour vous familiariser avec la bibliothèque graphique, écrivez un programme qui ouvre une fenêtre graphique et trace dedans un triangle. Vous tracerez les côtés du triangle en appelant la méthode `plotLine`.

Nous souhaitons maintenant paralléliser l'exécution du dessin en faisant tracer chaque côté du triangle par un `thread` différent. Dans un premier temps, nous construisons une solution basée sur une sous-classe de `Thread`.

Question 2

Créez une classe `DessineLigne` qui est une sous-classe de `Thread`. Une instance de cette classe doit permettre de tracer une ligne entre deux points dont les coordonnées sont des variables d'instance.

Écrivez une fonction `main` qui réalise le tracé du triangle en créant un `thread` pour dessiner chacun des côtés. Vérifiez que les différents tracés s'exécutent bien de manière parallèle.

L'autre manière de créer un `thread` est de passer au constructeur de la classe `Thread` un objet qui implémente l'interface `Runnable`.

Question 3

Proposez une nouvelle réalisation du tracé du triangle dans laquelle le tracé d'un côté est exécuté par un objet appartenant à la classe `DessineLigne` qui implémente l'interface `Runnable`.