

# Programmation avancée et répartie en Java : TP 3

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

## 1 Exercice 1

Le but de cet exercice est de mettre en pratique deux méthodes vues en cours pour créer des Threads. Pensez à compiler votre code après chaque question, et à l'exécuter lorsque possible (après les questions 3 et 5). Respectez les consignes vues lors du premier TP.

1. Créer une classe vide `Exercice1` dans un package approprié (toujours en respectant l'arborescence des sources vue au premier cours).
2. Dans la classe `Exercice1`, déclarer une classe interne `MonRunnable`, implémentant l'interface `Runnable`. Afficher un message en ligne de commande dans la méthode `run`.
3. Créer un constructeur dans la classe `Exercice1`. Dans ce constructeur, créer un objet `Thread` en donnant en argument au constructeur une instance de la classe `MonRunnable`. Lancer le thread en appelant sa méthode `start`. Ajouter une méthode `main` qui crée une instance de `Exercice1`.
4. Maintenant, créer une classe `MonThread`, toujours dans la classe `Exercice1`, qui étende la classe `Thread`. Ecrire un autre message dans sa méthode `run`.
5. Ajouter dans le constructeur de `Exercice1` le lancement d'une instance de la classe `MonThread`.

## 2 Exercice 2

1. Reprendre le Thread Compteur décrit dans le cours puis écrire un "main" qui lance plusieurs Compteurs en parallèle et observer l'ordonnancement.
2. Quel thread est actif et à quel moment ?
3. Reprendre de même la classe `Compteur2` du cours, et écrire une méthode `main` associée. Observer le résultat.

## 3 Exercice 3

Le but de l'exercice est d'afficher un texte animé (par un effet de déroulement).

1. Concevoir une fenêtre swing contenant un `JLabel` avec le texte de votre choix, et deux `JButton` "Démarrer" et "Arrêter". Utiliser le layout de votre choix.
2. Ajouter des gestionnaires d'événement pour ces deux boutons.

3. Ajouter un thread qui modifie toutes les 100 millisecondes (méthode `sleep`) le texte du `JLabel`, en décallant le caractère de droite pour le placer tout à gauche. Le thread devra s'exécuter tant qu'il n'est pas interrompu par le bouton "Arrêter". Utiliser la Javadoc de la classe `String` pour trouver les méthodes de traitement de chaînes nécessaires.

## 4 Exercice 4

1. En vous inspirant de l'exercice précédent, réaliser un composant Swing affichant une image animée.

## 5 Exercice 5

1. Créer une interface graphique Swing comprenant 2 `JPanels` dans un `BorderLayout`, au centre et à droite.
2. Dans le `JPanel` de droite, ajouter trois boutons, avec les texte "Ajouter un Thread", "Ajouter 10 Threads", et "Ajouter 100 Threads".
3. Créer une classe `Balle`, étendant `Thread`, telle que le constructeur ajoute un `JLabel` dans le `JPanel` central à une position aléatoire, et que la méthode `run` fasse bouger le `JLabel` à vitesse constante, en le faisant rebondir contre les bords du `JPanel`. On utilisera le `NullLayout` pour le `JPanel`.
4. Créer les gestionnaires d'événements pour les 3 boutons, tels qu'ils ajoutent des instances de la classe `Balle`.