TP : Les Threads

Exercice 1

Un "compteur" a un nom (CP1 par exemple) et il compte de 1 à max (nombre entier positif quelconque). Il marque une pause aléatoire entre chaque nombre (de 0 à 5000 millisecondes par exemple).
Un compteur affiche chaque nombre (CP1 affichera par exemple, "CP1 : 3") et il affiche un message du type "*** CP1 a fini de compter jusqu'à 10" quand il a fini.

1. Ecrivez la classe compteur avec 2 constructeurs : l'un initialise le nombre entier à 10, et l'autre l'initialise par la valeur entré en paramètre du constructeur.

Faites 2 versions : une o ù les threads sont créés avec une classe fille de Thread, et une où ils sont créés avec une instance d'une classe à part qui implémente Runnable.

2. testez les deux classes en lançant plusieurs compteurs qui comptent jusqu'à 10 ou bien jusqu'a max.

Exercice 2: problème d'accès concurrent

Voici 2 classes Compte (correspond à un compte bancaire) et Operation (thread qui effectue des opérations sur un compte bancaire).

```
package tpThread;
public class Compte {
  private int solde = 0;

  public void ajouter(int somme) {
    solde += somme;
    System.out.println(" ajoute " + somme);
  }

  public void retirer(int somme) {
    solde -= somme;
    System.out.println(" retire " + somme);
  }

   public void operationNulle(int somme) {
    solde += somme;
    System.out.println(" ajoute " + somme);
    solde -= somme;
    System.out.println(" retire " + somme);
  }

  public int getSolde() {
    return solde;
  }
}
```

```
public class Operation extends Thread {
  private Compte compte;

  public Operation(String nom, Compte compte) {
    super(nom);
    this.compte = compte;
  }

   public void run() {
    while (true) {
      int i = (int) (Math.random() * 10000);
      String nom = getName();
      System.out.print(nom);
//        compte.ajouter(i);
//        compte.retirer(i);
      compte.operationNulle(i);
      int solde = compte.getSolde();
      System.out.print(nom);
      if (solde != 0) {
  System.out.println(nom + ":**solde=" + solde);
  System.exit(1);
      }
    }
  }

  public static void main(String[] args) {
    Compte compte = new Compte();
    for (int i = 0; i < 20; i++) {
      Operation operation = new Operation("" + (char)('A' + i), compte);
      operation.start();
    }
  }
}
```

A/ Examinez le code et faites exécuter la classe Opération. Constatez le problème : opération effectue des opérations qui devraient laisser le solde du compte inchangé, et pourtant, après un moment, le solde ne reste pas à 0. Expliquez.

B/ Modifiez le code pour empêcher ce problème.

C/ Dans le code de Operation, remplacez l'opération nulle par 2 opérations ajouter et retirer qui devraient elles aussi laisser le solde du compte à 0 (elles sont en commentaire dans le code). Lancez l'exécution et constatez le problème. Modifiez le code pour que ça marche.