RT- Java Compte rendu

Exercice 1:

1. En Java, une classe ne doit hériter d'une autre classe que si le programmeur compte modifier ou améliorer la classe héritée par de nouvelles fonctionnalités.

Ceci dit, **Thread** est une classe, qui une fois héritée le programmeur peut redéfinir la méthode « *run()* » .

Runnable est une interface, permettant à une classe d'éxécuter un thread sans hériter de Thread. Cette méthode est préférable pour les classes dont les instances sont faites pour être éxécutés par des threads, sans pour autant être elles mêmes des threads.

2.

Méthode 1 : Thread:

```
public class EnumThread extends Thread {
   @Override
   public void run() {
      System.out.println("Thread started.");
      for(int i=1; i<27; i++){
         System.out.println("Enum: "+i);
         try {
            Thread.sleep(1000);
         } catch (InterruptedException e) {
            e.printStackTrace();
         }
      }
      System.out.println("Thread ended.");
   }
}
```

Méthode 2 : Runnable

```
public class AlphaRunnable implements Runnable {
   @Override
   public void run() {
      System.out.println("Runnable started.");
      for (char c='a'; c<='z'; c++){
         System.out.println("Runnable: "+c);
         try {
            Thread.sleep(1000);
         } catch (InterruptedException e) {
            e.printStackTrace();
         }
      }
      System.out.println("Runnable ended.");
   }
}
```

Main (classe motrice) :

```
public class Exercice1Main {
   public static void main(String[]args){
      new EnumThread().start();
      new Thread(new AlphaRunnable() ).start();
   }
}
```

Résultat d'éxecution :

```
Thread started.
Runnable started.
Runnable: a
Enum: 1
Runnable: b
Enum: 2
Runnable: c
Enum: 3
Runnable: d
Enum: 4
```

...

```
Enum: 21
Runnable: v
Enum: 22
Runnable: w
Enum: 23
Runnable: x
Enum: 24
Runnable: y
Enum: 25
Runnable: z
Enum: 26
Runnable ended.
Thread ended.
```

Les threads ont l'air d'être éxecuter en parallèle, comme s'il était deux process différent et indépendant reliés au programme les appelants.

Il me semble que ce parallélisme est géré par la JVM et l'OS.