Exercice 1 :

public class Talkative implements Runnable {  
 private int number;  
  
 public Talkative(int number) {  
 this.number = number;  
 }  
 @Override  
 public void run() {  
 for (int i = 0; i < 100; i++) {  
 System.*out*.println(number);  
 }  
 }  
}

public class Main {  
 public static void main(String[] args) {  
 for (int i = 0; i < 10; i++) {  
 Thread thread = new Thread(new Talkative(i));  
 thread.start();  
 }  
 }

}

"C:\Users\rmitar\AppData\Local\Programs\Eclipse Adoptium\jdk-17.0.9.9-hotspot\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2023.3.2\lib\idea\_rt.jar=57028:C:\Program Files\JetBrains\IntelliJ IDEA 2023.3.2\bin" -Dfile.encoding=UTF-8 -classpath C:\Users\rmitar\IdeaProjects\TP6\_Threads\_01\out\production\TP6\_Threads\_01 Main

0

3

2

2

5

6

1

1

6) –

La sortie de l'exécution de ce programme ne sera pas dans l'ordre car les threads s'exécuteront de manière concurrente.

Exercice 2 :

public class Sommeur implements Runnable{  
 private int[] array;  
 private int start;  
 private int end;  
 private int sum;  
  
 public Sommeur(int[] array, int start, int end) {  
 this.array = array;  
 this.start = start;  
 this.end = end;  
 this.sum = 0;  
 }  
 @Override  
 public void run() {  
 for (int i = start; i < end; i++) {  
 sum += array[i];  
 }  
 }  
  
 public int getSomme() {  
 return sum;  
 }  
}

import java.util.concurrent.ExecutorService;  
import java.util.concurrent.Executors;  
import java.util.concurrent.TimeUnit;  
  
public class Main {  
  
 public static void main(String[] args){  
 int[] array = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}; // Example array  
 int numberOfThreads = 2; // Example number of threads  
 ExecutorService executor = Executors.newFixedThreadPool(numberOfThreads);  
 int length = array.length;  
 int chunkSize = length / numberOfThreads;  
 Sommeur[] sommeurs = new Sommeur[numberOfThreads];  
  
 for (int i = 0; i < numberOfThreads; i++) {  
 int start = i \* chunkSize;  
 int end = (i == numberOfThreads - 1) ? length : start + chunkSize;  
 sommeurs[i] = new Sommeur(array, start, end);  
 executor.submit(sommeurs[i]);  
 }  
  
 executor.shutdown();  
  
 try {  
 executor.awaitTermination(1, TimeUnit.MINUTES);  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
  
 int totalSum = 0;  
 for (Sommeur sommeur : sommeurs) {  
 totalSum += sommeur.getSomme();  
 }  
  
 System.*out*.println("Total sum: " + totalSum);  
 }  
}

"C:\Users\rmitar\AppData\Local\Programs\Eclipse Adoptium\jdk-17.0.9.9-hotspot\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2023.3.2\lib\idea\_rt.jar=57138:C:\Program Files\JetBrains\IntelliJ IDEA 2023.3.2\bin" -Dfile.encoding=UTF-8 -classpath C:\Users\rmitar\IdeaProjects\TP6\_Threads\_01\out\production\TP6\_Threads\_01 Main

Total sum: 55

Process finished with exit code 0