Problem 1:

Find the sum of array using multithreading capabilities. To this end, declare an array with N elements (for example $N = 10000$) and fill it with pseudorandom numbers. Create K thread objects of `PartialSum` class, each thread i should calculate the partial sum for your array (thread 0 calculates the partial sum from 0 to m-1, thread 1 calculates the partial sum from m to 2m-1, etc., where $m = N/K$.) When all the threads are finished, calculate the total sum from the partial sums. Calculate the sum of the array in conventional way, and compare your results. Create the thread objects in two ways, by omplementing `Runnable` interface and by subclassing `Thread` class.

Problem 2:

Create two thread objects that share the same array with N elements (for example $N = 5$). One thread, instantiation of the `GetNumbers` should read the numers from the keyboard, and fill an array with them. The second thread, instantiation of the `GetSum` should calculate the sum of the array (when the array is filled), print the, and then fill the array with zeros, and wait until the array has been filled again. Similarly, the first thread should wait with asking for new numbers, until the sum is calculated. This will happen very fast, so you may add an artificial delay to the second thread, using `Thread.sleep()` method, to check if the first thread is indeed waiting.