MODULE-3 MultiThreads ASSIGNMENT-6

```java
package assignments;

import java.util.List;
import java.util.concurrent.ForkJoinPool;
import java.util.concurrent.RecursiveAction;

public class Assignment6 {

    private static final ForkJoinPool pool = new ForkJoinPool(); // ForkJoinPool for
parallel execution

    public static void main(String[] args) {
        System.out.println("Initializing Parallel Universe Explorer...");

        // Example list of tasks for initialization
        List<String> initializationTasks = List.of(
            "ONE","TWO","THREE","FOUR","FIVE"
        );

        // Create and invoke the RecursiveAction for initializing the explorer
        InitializationTask task = new InitializationTask(initializationTasks);
        pool.invoke(task);

        System.out.println("Initialization complete!");
    }
}

// RecursiveAction for initializing the explorer
class InitializationTask extends RecursiveAction {
    private final List<String> tasks;
    private static final int THRESHOLD = 2; // Threshold for splitting tasks

    InitializationTask(List<String> tasks) {
        this.tasks = tasks;
    }

    @Override
    protected void compute() {
        if (tasks.size() <= THRESHOLD) {
            // Perform tasks directly if below threshold
            tasks.forEach(task -> System.out.println("Executing task: " + task));
        } else {
            // Split tasks into subtasks and process them in parallel
            int mid = tasks.size() / 2;
            InitializationTask leftTask = new InitializationTask(tasks.subList(0,
mid));
            InitializationTask rightTask = new InitializationTask(tasks.subList(mid,
tasks.size()));
            invokeAll(leftTask, rightTask);
        }
    }
}
```