```
import java.util.concurrent.*;
class Task implements Callable<String> {
   private int taskId;

   public Task(int id) {
      this.taskId = id;
   }
   @Override
   public String call() {
      try {
         System.out.println("Task " + taskId + " started by " + Thread.currentThread().getName());

         // Simulate work
         Thread.sleep(1000);

         // Example failure handling
         if (taskId == 3) {
            throw new RuntimeException("Error in Task " + taskId);
         }

         System.out.println("Task " + taskId + " completed");
         return "Result from Task " + taskId;

      } catch (Exception e) {
         return "Task " + taskId + " failed: " + e.getMessage();
      }
   }
}
```

```
public class TaskScheduler {
    public static void main(String[] args) throws Exception {

        int poolSize = 3;
        ExecutorService executor = Executors.newFixedThreadPool(poolSize);

        long startTime = System.currentTimeMillis();

        Future<String>[] futures = new Future[5];

        // Submit multiple tasks
        for (int i = 0; i < 5; i++) {
            futures[i] = executor.submit(new Task(i + 1));
        }

        // Monitor task execution
        for (int i = 0; i < 5; i++) {
            System.out.println(futures[i].get());
        }

        // Graceful shutdown
        executor.shutdown();
        executor.awaitTermination(5, TimeUnit.SECONDS);

        long endTime = System.currentTimeMillis();

        System.out.println("All tasks finished.");
        System.out.println("Total Time: " + (endTime - startTime) + " ms");
```

```
    }
}
```

```java
import java.util.concurrent.*;

class Task implements Callable<String> {
    private int taskId;

    public Task(int id) {
        this.taskId = id;
    }

    @Override
    public String call() {
        try {
            System.out.println("Task " + taskId + " started by " + Thread.currentThread().getName());

            // Simulate work
            Thread.sleep(millis: 1000);

            // Example failure handling
            if (taskId == 3) {
                throw new RuntimeException("Error in Task " + taskId);
            }

            System.out.println("Task " + taskId + " completed");
            return "Result from Task " + taskId;

        } catch (Exception e) {
            return "Task " + taskId + " failed: " + e.getMessage();
        }
    }
}
```

```
Task 1 started by pool-1-thread-1
Task 3 started by pool-1-thread-3
Task 2 started by pool-1-thread-2
Task 1 completed
Task 2 completed
Task 4 started by pool-1-thread-3
Task 5 started by pool-1-thread-2
Result from Task 1
Result from Task 2
Task 3 failed: Error in Task 3
Task 4 completed
Task 5 completed
Result from Task 4
Result from Task 5
All tasks finished.
Total Time: 2066 ms
PS C:\Users\Adity chavan\Desktop\nikita.html>
```