**ListenableFuture из Guava**

C помощью метода ***Futures.allAsList()*** мы можем объединить ***несколько*** экземпляров ***ListenableFuture*** в ***один*** ***ListenableFuture***, который завершается только после успешного завершения всех объединенных потоков, каждый из которых выполняет свою задачу:

ExecutorService executorService = Executors.newCachedThreadPool(); // (неограниченный пул потоков, с автоматическим восстановлением потока)  
ListeningExecutorService listeningExecutorService =   
 MoreExecutors.listeningDecorator(executorService);  
  
ListenableFuture<String> future1 =   
 listeningExecutorService.submit(() -> "Hello");

ListenableFuture<String> future2 =   
 listeningExecutorService.submit(() -> "World");  
  
String greeting = Futures.allAsList(future1, future2).get()  
 .stream()  
 .collect(Collectors.joining(" "));  
assertEquals("Hello World", greeting);