1. 전체적인 계산방법
   1. 각각의 수들을 이중 ArrayList에 각 자릿수별로 입력

(ex. […..[1,0]. [1,1] .,,,,.[4,9,9,9,9],[5,0.0,0,0]])

* 1. 시작하는 수와 끝나는 수를 인자로 받아 그 사이의 수를 곱하는 50개의 쓰레드를

생성, 1부터 50000까지의 수를 1~1000, 1001~2000, 2001~3000….. 으로 50등분 하여 각각의 쓰레드에 넣어줍니다.

* 1. 50개의 곱셈결과값이 반환되면 두개의 수를 인자로 입력받아 두 수를 곱하는 25개의 쓰레드를 생성, 각각의 값을 두개씩 해당 쓰레드에 넣어줍니다.
  2. 마찬가지로 25개의 결과값을 다시 두수를 입력받고 곱한결과를 반환하는 쓰레드에 넣어줍니다.(이때 24개가 들어가고 25번째 결과값이 남는데 이건 다음결과 값으로

그대로 입력)

* 1. 이와 같은 방식으로 하나의 결과값이 나올 때까지 반복한후 해당 결과값을

출력합니다.

1. 세부 동작 구성
   1. Starmain에서 main함수 실행하면서 1.2에서 언급된 쓰레드클래스 Factorial12를 실행합니다.
   2. 각각의 쓰레드에서 Starmain의 call 함수를 실행하여 Starmain의 arr에 각각의 결과값을 ArrayList<Integer>형태로 저장합니다.
   3. .join을 통해 모든 쓰레드가 끝날 때 까지 기다린 후 1.3에서 언급한 쓰레드클래스 starTwo에 arr의 0, 1번 , 2, 3번, 4, 5번 …… 과 같이 넣어줍니다.
   4. 각각의 쓰레드에서 Starmain의 call 함수를 실행하여 Starmain의 arr2에 각각의 결과값을 ArrayList<Integer>형태로 저장합니다.
   5. .join을 통해 모든 쓰레드가 끝날 때까지 기다린 후 arr를 clear, 다시 Starmain에 arr2의 0, 1번, 1, 2번 …. 과 같이 넣어줍니다. 이 때 arr2에 있는 값들 중 마지막 25번째 값은 페어가 없으므로 쓰레드에 넣지 않고 지나갑니다.
   6. 각각의 쓰레드에서 Starmain의 call 함수를 실행하여 Starmain의 arr에 각각의 결과값을 ArrayList<Integer>형태로 저장합니다. 이때 2.5에서 지나간 25번째 값을 arr에 넣어줍니다.
   7. 위와 같은 방식으로 결과값이 1개가 나올 때 까지 계산을 하고 이후 해당 값을 텍스트 파일로 저장합니다.
2. 코드 리뷰
   1. starTwo클래스는 ArrayList<Integer> 형태로 이루어진 두 수를 받아 곱셈 후 return 하는 함수입니다. 예를 들어 102는 [1, 0, 2], 2561은 [2, 5, 6, 1] 과 같은 형태로 주어집니다. 이렇게 두개의 Array를 입력 받은 후에 각 수를 102\*1 + 102\*6\*10 + 102\*5\*100 + 102\*2\*1000과 같이 계산 하여 결과값을 찾아냅니다. 이때 덧셈은 plusarr 함수에서 이루어지는데 예를 들어 14+ 18의 경우 4+8 -> carry = 1 sum = 2 , 1+1 -> sum = carry(1) + 2 carry = 0 으로 각 자리수 별로 돌아가면서 carry와 sum을 계산하어 합을 찾아냅니다.
   2. Factorial12클래스는 int 변수 2개를 받아 그 구간내의 수들을 곱한 후 return 하는 함수입니다. 예를 들어 1, 1000이 입력되었다고 하면 해당 수들은 Factorial12 내의 allNum이라는 ArrayList<Integer>에 [1, 2, 3, …. , 1000] 과 같이 저장되고 starSum과 num이라는 ArrayList<Integer>에서 1과 2를 하나씩 뽑아가서 앞서 starTwo클래스의 곱셈방식을 활용하여 계산됩니다. 이렇게 되면 맨 처음에 1과 2를 뽑아가서 계산하고 num을 해당 결과값으로 업데이트, starsum을 3으로 업데이트, 또 다시 두수를 곱한 후 num, starsum 업데이트….. 이런식으로 끝까지 반복됩니다. 이후 결과값을 Starmain으로 return 하면서 끝납니다.
3. 소요시간
   1. starTwo 6개 실행까지는 반나절정도 걸렸으나 이후 기하급수적으로 자릿수가 늘어남에 따라 점점 오래 걸려 4일간 켜놓은 결과 정상적으로 결과값이 출력되었습니다. 여기서 각각의 곱셈 계산 부분 또한 쓰레드 처리를 해놓는다면 훨씬 더 단축된 소요시간을 기대할 수 있다고 생각합니다.