과제3) 자바 람다식과 스트림

1. 객체지향 언어에서 인터페이스는 그 시그너처와 선언이 변하지 않는다는 전제로 하여 다형성을 정의하는 객체간의 규약입니다. 자바8의 인터페이스는 인터페이스를 업그레이드하는 개념을 도입하였습니다. 새롭게 변경된 인터페이스에 대한 내용을 각각 기술하세요. (30점)
   1. 인터페이스에서 구현 메서드를 작성할 수 있는 방법을 기술하세요.

인터테이스에서 구현 메서드를 작성 할 수 있는 방법은 comparator 인터페이스를 구현하건 comparator 구현 객체를 통해 새롭게 생성한다.

첫번째 인터페이스 클래스 생성

하나의 함수를 가진 펑셔널 인터페이스를 파라미터로 넘겨줌

두번째 익명클래스를 통해 구현함

세번째 람다식

* 1. 자바9 에서 새롭게 추가된 구현 메서드를 작성할 수 있는 방법을 기술하세요.

자바 9에서 새롭게 추가된 구현 메서드를 작성할 수 있는 방법은

* 1. 함수형 인터페이스에 대해서 설명하세요.

함수형 인터페이스는 람다식으로 변경해서 파라미터로 넣어줄 수 있다.

함수형 인터페이스는 나중에 실행 할 수 있게 전달하는 코드블럭으로 함수를 메소드로 전달하기 위해 java8부터 새롭게 정의된 덧으로 람다 표현식이라고 부르기도 한다.

전체 리스트에 있는 요소들에 적용한다. 그리고 그 결과를 출력함

1. 문제로 제공되는 stream-data.txt 파일에 대해서 다음을 각각 수행하세요. (예외는 별도로 처리하지 않습니다.) (50점)
   1. stream-data.txt 파일은 하나의 숫자를 행으로 가지고 있는 텍스트 파일입니다. 각 행을 읽어서 Stream<String> 타입으로 작성하세요.
   2. A를 IntStream 타입으로 변환하는 코드를 작성하세요.
   3. B 에서 작성된 코드에서 map 중간연산을 사용해서 각 요소를 2배로 해서 새롭게 스트림을 생성합니다.
   4. C에서 생성된 스트림에서 50미만의 숫자는 제외하고 새롭게 스트림을 생성합니다.
   5. 최종적으로 모든 요소의 합을 구해서 출력합니다. (System.out.println)
2. 2번 문제를 스트림을 사용하지 않고 for 문을 사용해서 결과를 도출하는 방식을 Imperative Programming(명령형 프로그래밍, 스트림을 사용하여 처리하는 방식을 선언형이라고 함) 이라고 한다. 다음을 각각 수행할 수 있도록 코드를 작성합니다. (예외는 별도로 처리하지 않습니다.) (20점)
   1. stream-data.txt 파일을 읽어 List<Integer> 혹은 ArrayList<Integer> 타입으로 데이터를 읽어옵니다.
   2. for 문을 사용하여 요소 중 50을 초과하는 요소의 총 합을 구해서 출력합니다. (각 요소를 2배로 곱하지 않습니다.)