

10-1. java.lang

- 패키지(package): 개념적으로는 클래스의 묶음이고, 물리적으로는 디렉토리이다.
- 임포트(import): 컴파일러에게 현재 소스파일에서 사용된 클래스에 대한 정보를 알려 주는 명령어이다.
- java.lang 패키지: 자바에서 가장 기본이 되는 클래스들의 묶음이다.
- Object 클래스: 모든 클래스의 최상위 클래스이다.
- Math 클래스: 수학 연산 메소드들이 정의된 클래스이다.
- String, StringBuffer, StringBuilder: 문자열을 다루는 클래스로, String 클래스는 값이 변하지 않는 클래스이고, StringBuffer, StringBuilder 클래스는 값이 변할 수 있는 클래스이다.
- Wrapper 클래스: 기본 타입 값을 객체로 변환하여 사용하기 위한 클래스이다.

10-2. 자주 사용되는 클래스

- BigInteger: long 타입 범위를 넘어서는 정수를 다루기 위한 클래스이다.
- BigDecimal: float, double이 가지는 부동 소수점의 단점을 해결하기 위한 클래스이다.
- Date: 자바에서 날짜와 시간을 다루기 위해 가장 먼저 개발된 클래스이다.
- Calendar: Date 클래스 이후에 나온 날짜 및 시간을 다루는 추상 클래스이다.
- java.time 패키지: Date, Calendar 클래스의 단점을 해결하기 위해 개발된 패키지이다.

11-1. List

- List: 요소를 인덱스로 관리하고 중복을 허용하는 인터페이스이다.
- ArrayList: List 인터페이스를 구현한 클래스로, 가장 많이 사용되는 자료구조 중 하나이다.
- LinkedList: List 인터페이스를 구현한 클래스로, ArrayList와 다르게 'Node class'의 연결로 이루어져 있다.

11-2. Set & Map

- Set: List 인터페이스와 다르게 저장 순서에 따라 인덱스가 존재하지 않고 중복된 객체 저장을 허용하지 않는 인터페이스이다.
- HashSet: Set 인터페이스를 구현한 클래스 중 가장 많이 쓰이는 클래스이다.
- Iterator: 자료구조에 저장된 요소를 읽어올 때 사용하는 인터페이스이다.
- Map: 키(key), 값(value)으로 구성된 인터페이스로, 키(key) 값은 중복될 수 없다.
- HashMap: Map 인터페이스를 구현한 클래스로, 키(key) 값으로 동일한 객체인지 판단하기 위해 오버라이딩이 필요하다.

11-3. Queue & Stack

- Queue: FIFO(First In First Out, 선입선출) 자료구조로 데이터가 저장된 순서대로 데이터가 나가는 인터페이스이다.
- PriorityQueue: Queue 인터페이스의 구현 클래스이며, 데이터를 저장한 순서와 상관없이 우선순위에 맞게 데이터를 꺼낼 수 있는 자료구조이다.
- Deque(Double-Ended Queue): 데이터를 양쪽으로 추가 또는 삭제가 가능한 인터페이스이다.
- Stack: LIFO(Last In First Out, 후입선출) 자료구조로 최신 추가된 데이터 순서대로 데이터가 나가는 클래스이다.
- Comparable: java.lang 패키지에 있는 인터페이스로 Comparable 인터페이스를 구현한 클래스만 매개 변수 없이 sort() 메소드를 통한 정렬이 가능하다.
- Comparator: java.util 패키지에 있는 인터페이스로 해당 클래스의 정렬 기준을 새로 정하여 정렬 방법을 다양하게 제공하는 클래스이다.

12-1. 지네릭스(Generics)

- 지네릭스: 객체의 타입을 나중에 정하고 객체의 타입을 체크해 주는 기능이다.
- 지네릭 클래스: 멤버 타입을 일반화된 타입(<T>)으로 표시한 클래스이다.
- 와일드 카드: 제네릭 클래스에서 일반화된 타입(<T>)에 모든 타입을 대입할 수 있도록 하는 기능이다.

12-2. 열거형(Enum)

- 열거형: 관련이 있는 상수끼리 모여 있는 집합체(클래스)이다.

13-1. 스레드

- 프로세스: 운영체제로부터 자원을 할당받은 작업의 단위로, 실행 중인 프로그램이다.
- 스레드: 프로세스가 할당받은 자원을 이용하는 실행의 단위이다.
- Runnable: 추상 클래스인 run() 메소드를 가지고 있는 인터페이스로, Runnable 인터페이스 및 run() 메소드를 구현하여야 Thread 인스턴스를 만들 수 있다.
- 동기화 메소드: 하나의 스레드가 점유 중일 때 다른 스레드가 접근할 수 없는 메소드이다.

13-2. 스레드 제어

- 스레드 제어: 스레드의 상태를 제어하는 것을 말하며, 스레드는 실행 대기, 실행, 일시 정지, 종료 상태를 가질 수 있다.

실행 결과

```
1과 100 사이의 값을 입력하세요:50
더 작은 수를 입력하세요.
1과 100 사이의 값을 입력하세요:25
더 작은 수를 입력하세요.
1과 100 사이의 값을 입력하세요:12
더 큰 수를 입력하세요.
1과 100 사이의 값을 입력하세요:17
맞습니다.
시도 횟수는 4번입니다.
```

위와 같이 System.in 인자를 통해 Scanner 클래스를 활용하여 표준 입력, System.out을 통해 표준 출력, System.err를 통해 표준 에러를 출력할 수 있다.

- 스트림: 입출력장치와 프로그램 간의 연결통로이다.
- 바이트 기반 스트림: 1byte 단위로 데이터를 읽어 들이는 스트림으로, 주로 이미지, 동영상 등의 파일을 읽어 들일 때 사용한다.
- 문자 기반 스트림: 2Byte 단위로 데이터를 읽어 들이는 스트림으로, 유니코드 문자를 읽어들일 경우 사용한다.
- InputStream, OutputStream: 바이트 기반 스트림의 최상위 조상 클래스이다.
- BufferedInputStream, BufferedOutputStream: 바이트 기반 스트림에 연결되어 버퍼를 제공하는 보조 스트림이다.
- Reader, Writer: 문자 기반 스트림의 최상위 추상 클래스이다.
- BufferedReader, BufferedWriter: 문자 기반 스트림에 연결되어 버퍼를 제공하는 보조 스트림이다.

15-1. 람다식

- 람다식: 메소드를 식으로 표현한 것
- 함수형 인터페이스: 하나의 추상 메소드만 선언된 인터페이스
- java.util.function: 자주 사용되는 형태의 함수형 인터페이스 및 메소드를 제공하는 패키지
- Supplier<T>: 매개 변수가 없고 반환 타입이 지네릭 타입(<T>)인 get() 메소드를 가지고 있는 인터페이스
- Consumer<T>: 매개 변수가 1개 있고 반환 타입이 없는 메소드인 accept(T t)와 andThen(Consumer<? super T> after)를 가지고 있는 인터페이스
- Function<T,R>: 매개 변수와 반환 값 개수가 각각 1개이고 반환 타입이 지네릭 타입(<R>)인 apply(T t)를 가지고 있는 인터페이스
- Predicate<T>: Predicate 인터페이스는 Function 인터페이스의 반환 타입이 boolean이고 test(T t) 메소드를 가지고 있는 인터페이스

15-2. 스트림

- 스트림: 다양한 자료구조를 표준화된 방법으로 다루기 위한 도구
- Optional<T>: T타입 객체의 래퍼 클래스
- 중간 연산: 연산 반환 값이 스트림인 연산이고 여러 번 수행 가능
- map: 스트림에서 원하는 요소들을 변환시키거나 추려낼 때 사용하는 중간 연산
- 최종 연산: 연산 결과가 스트림이 아닌 연산으로 스트림을 소모하여 합이나 평균값, 컬렉션, 배열 등을 반환하며, 스트림마다 한 번만 수행 가능
- reduce: 스트림의 요소를 줄여나가면서 연산을 수행하는 최종 연산