l.컬렉션 프레임워크 개요 in 자바

설명에 앞서 배열은 자바의 기본 자료구조이다. 자료구조란 데이터(값)를 저장하고 관리하는 방법인데, 배열은 같은 자료형의 데이터를 연속적으로 저장하는 구조이다. 기본 자료형 데이터는 물론 객체도 저장할 수 있다. 또한 배열의 저장된 데이터는 인덱스를 통해 접근할 수 있다.

Tip) 배열도 객체이다. 자바에서 모든 객체는 힙에 저장되고, 참조로 관리된다. 배열도 마찬가지로 배열을 생성하면 힙에 배열의 요소를 저장하는 메모리 공간이 할당된다. 그리고 스택에 생성된 참조 변수에 배열 객체의 주소가 저장되며 이를 통해 배열에 접근한다. 그 외 더 다양한 이유가 있지만, 참조를 통한 메모리 관리 방식이 자바에서 배열을 객체로 취급하는 가장 큰 이유이다.

배열에 데이터를 삽입하거나 삭제하려면 직접 데이터를 이동시키는 코드를 작성해야 한다. 그래서 데이터의 삽입이나 삭제가 쉽지 않다. 또한, 배열은 한 번 생성하면 크기가 고정되고, 저장 공간의 크기도 고정되므로 나중에 데이터를 추가하기도 어렵다. 이를 개선하고자 자바에서는 컬렉션을 제공한다. 컬렉션은 데이터를 그룹으로 묶어 하나의 객체로 관리할 수 있게 하는 자료구조이다. 컬렉션의 종류로는 리스트(list), 집합(set), 맵(map), 큐(queue), 스택(stack)등이 있다. 위에 설명한 대로 배열은 크기를 지정하면 크기를 변경할 수 없다. 하지만 컬렉션은 동적 크기를 가지고, 필요에 따라 크기를 자동으로 조정할 수 있으며, 데이터의 추가와 삭제도 쉽다. 또한 배열은 같은 자료형의 데이터로 구성되지만, 컬렉션은 다양한 자료형을 지원하며 여러 타입의 객체를 담을 수 있다. 배열은 인덱스를 통해 접근할 수 있지만, 다른 기능은 거의 제공하지 않는다(추가, 삭제, 검색, 정렬등). 하지만 컬렉션은 이 기능을 메서드로 제공하므로 데이터를 관리하기 편하다. 또한 다양한 자료구조와 알고리즘이 이미 구현되어 있다.

자바에서는 컬렉션을 다루기 위해 컬렉션 프레임워크를 제공한다. 컬렉션 프레임워크는 컬렉션을 효율적으로 다루기 위해 설계된 인터페이스와 클래스의 집합이다. 컬렉션이 데이터의 집합을 나타내는 개념이라면, 컬렉션 프레임워크는 컬렉션을 다루기 위해 설계한 코드 구조이자 도구이다. 컬렉션 프레임워크는 다양한 컬렉션을 손쉽게 생성하고 수정 및 관리할 수 있도록 도와준다. 또한 자료구조의 구현을 신경 쓰지 않고 표준화된 구로를 이용해 일관된 방식으로 작업할 수 있게 도와준다.

컬렉션 프레임워크의 구조는 아래와 같다.

Collection 인터페이스 – 컬렉션 프레임워크의 최상위 인터페이스로, 각 컬렉션의 공통된 동작을 정의, 하지만 Collection 인터페이스을 직접 구현한 클래스는 없음. 그 대신 자식 인터페이스들이 이를 상속해 기능을 확장함.

List 인터페이스 – 삽입된 순서대로 요소를 저장하는 리스트 컬렉션을 정의하는 인터페이스이다. 중복 요소를 허용하며, 각 요소에는 인덱스로 접근할 수 있다. 주요 구현 클래스는 다음과 같다.

ArrayList : 배열로 구현된 클래스

LinkedList : 이중 연결 리스트로 구현된 클래스

Vector: ArrayList와 유사한 리스트이나 현재는 잘 사용하지 않음

Stack : Vector 클래스를 상속받아 구현된 클래스이다. 나중에 들어온 요소가 먼처 처리되는 LIFO(last in first out,후입선출) 방식으로 요소를 관리한다.

Set 인터페이스 – 순서를 보장하지 않으며, 중복 요소를 허용하지 않는 집합 컬렉션을 정의하는 인터페이스이다. 수학의 집합 개념을 생각하면 이해하기 쉽다. 주요 구현 클래스는 아래와 같다.

HashSet : 해시 테이블 구조로 구현된 집합이다.

LinkedHashSet : HashSet의 변형 집합으로, 삽입된 순서를 유지한다.

TreeSet : 이진 트리 구조로 구현된 집합으로, 요소가 정렬된 상태로 저장된다.

이외에 Map 인터페이스, Queue 인터페이스, Deque 인터페이스가 있다.