*성명: 무키모프 함담 / MUQIMOV HAMDAM*

*학번: 2019117367*

*소속: 과학기술대학 소프트웨어학과*

*학년: 3학년*

**과목: JAVA 2**

**Chapter 7**

**SUMMARY (1,4,5)**

2. **제네릭은** 일반화 시킨 타입의 매개 변수를 이용하여 클래스, 인터페이스, 메소드를 일반화시키는 기법이다. 그러므로 일반화시킨 타입에 구체적인 타입을 지정하여 특정 타입으로만 사용할 수 있다.

3**. 컬렉션은** JDK 1.5 버전 이후 제네릭 기법으로 만들어졌다.

4. 컬렉션의 요소로는 오직 **객체**만 사용된다.

5. JDK 1.5부터는 **자동 박싱 / 언박싱이** 지원되어 기본 타입을 컬렉션에 바로 삽입하고 추출할 수 있다.

6. **Vector<E>**는 배열을 가변 크기로 다룰 수 있게 한 벡터 컬렉션으로서, 객체의 삽입, 삭제, 이동이 쉽고, 배열처럼 인덱스 번호로 원소를 접근할 수 있다.

7. **ArrayList<E>** 역시 배열을 가변 크기로 다룰 수 있는 하는 컬렉션으로서 Vector<E>와 거의 유사하나 멀티스레드 동기화를 지원하지 않는다. 다음은 Point 클래스의 객체만 다루는 ArrayList를 생성하는 코드이다.

ArrayList<Point> a = new ArrayList<Point>();

8.9. HashMap<K, V>은 **키와 값의** 쌍을 하나의 원소로 다루는 해시맵 컬렉션으로서, 인덱스로는 검색할 수 없다. 다음은 키가 문자열이고 , 값이 정수인 해시맵은 생성하는 코드이다.

HashMap<String, Integer> h = new HashMap<String, Integer>();

10. Iterator <E>는 Collection<E> 인터페이스를 구현하여 컬렉션의 원소를 순차적으로 검색할 수 있는 컬렉션이다. Collection<E>는 구현한 모든 컬렉션에 **iterator()** 메소드를 호출하면 Iterator 객체를 리턴하며, 이 객체를 이용하여 인덱스 없이 컬렉션의 요소를 순차 검색할 수 있다.

Vector<Integer> v = new Vector<Integer>();

Iterator<Integer> it = v.iterator();