

# 11. 다음으로

#1.인강/0.자바/4.자바-중급2편

## 학습 내용 정리

### 전체 목차

#### 1. 제네릭 - Generic1

- /프로젝트 환경 구성
- /제네릭이 필요한 이유
- /다형성을 통한 중복 해결 시도
- /제네릭 적용
- /제네릭 용어와 관례
- /제네릭 활용 예제
- /문제와 풀이1

#### 2. 제네릭 - Generic2

- /타입 매개변수 제한1 - 시작
- /타입 매개변수 제한2 - 다형성 시도
- /타입 매개변수 제한3 - 제네릭 도입과 실패
- /타입 매개변수 제한4 - 타입 매개변수 제한
- /제네릭 메서드
- /제네릭 메서드 활용
- /와일드카드1
- /와일드카드2
- /타입 이레이저
- /문제와 풀이2
- /정리

#### 3. 컬렉션 프레임워크 - ArrayList

- /배열의 특징1 - 배열과 인덱스

- /빅오(O) 표기법
- /배열의 특징2 - 데이터 추가
- /직접 구현하는 배열 리스트1 - 시작
- /직접 구현하는 배열 리스트2 - 동적 배열
- /직접 구현하는 배열 리스트3 - 기능 추가
- /직접 구현하는 배열 리스트4 - 제네릭1
- /직접 구현하는 배열 리스트5 - 제네릭2
- /정리

## 4. 컬렉션 프레임워크 - LinkedList

- /노드와 연결1
- /노드와 연결2
- /노드와 연결3
- /직접 구현하는 연결 리스트1 - 시작
- /직접 구현하는 연결 리스트2 - 추가와 삭제1
- /직접 구현하는 연결 리스트3 - 추가와 삭제2
- /직접 구현하는 연결 리스트4 - 제네릭 도입
- /정리

## 5. 컬렉션 프레임워크 - List

- /리스트 추상화1 - 인터페이스 도입
- /리스트 추상화2 - 의존관계 주입
- /리스트 추상화3 - 컴파일 타임, 런타임 의존관계
- /직접 구현한 리스트의 성능 비교
- /자바 리스트
- /자바 리스트의 성능 비교
- /문제와 풀이1
- /문제와 풀이2
- /정리

## 6. 컬렉션 프레임워크 - 해시(Hash)

- /리스트(List) vs 세트(Set)

- /직접 구현하는 Set0 - 시작
- /해시 알고리즘1 - 시작
- /해시 알고리즘2 - index 사용
- /해시 알고리즘3 - 메모리 낭비
- /해시 알고리즘4 - 나머지 연산
- /해시 알고리즘5 - 해시 충돌 설명
- /해시 알고리즘6 - 해시 충돌 구현

## 7. 컬렉션 프레임워크 - HashSet

- /직접 구현하는 Set1 - MyHashSetV1
- /문자열 해시 코드
- /자바의 hashCode()
- /직접 구현하는 Set2 - MyHashSetV2
- /직접 구현하는 Set3 - 직접 만든 객체 보관
- /equals, hashCode의 중요성1
- /equals, hashCode의 중요성2
- /직접 구현하는 Set4 - 제네릭과 인터페이스 도입

## 8. 컬렉션 프레임워크 - Set

- /자바가 제공하는 Set1 - HashSet, LinkedHashSet
- /자바가 제공하는 Set2 - TreeSet
- /자바가 제공하는 Set3 - 예제
- /자바가 제공하는 Set4 - 최적화
- /문제와 풀이1
- /문제와 풀이2
- /정리

## 9. 컬렉션 프레임워크 - Map, Stack, Queue

- /컬렉션 프레임워크 - Map 소개1
- /컬렉션 프레임워크 - Map 소개2
- /컬렉션 프레임워크 - Map 구현체
- /스택 자료 구조

- /큐 자료 구조
- /Deque 자료 구조
- /Deque와 Stack, Queue
- /문제와 풀이1 - Map1
- /문제와 풀이2 - Map2
- /문제와 풀이3 - Stack
- /문제와 풀이4 - Queue
- /정리

## 10. 컬렉션 프레임워크 - 순회, 정렬, 전체 정리

- /순회1 - 직접 구현하는 Iterable, Iterator
- /순회2 - 향상된 for문
- /순회3 - 자바가 제공하는 Iterable, Iterator
- /정렬1 - Comparable, Comparator
- /정렬2 - Comparable, Comparator
- /정렬3 - Comparable, Comparator
- /컬렉션 유틸
- /컬렉션 프레임워크 전체 정리
- /문제와 풀이
- /정리

## 로드맵 소개

### 실전 자바 로드맵

- 김영한의 자바 입문 - 코드로 시작하는 자바 첫걸음 (오픈)
- 김영한의 실전 자바 - 기본편 (오픈)
- 김영한의 실전 자바 - 중급 1편 (오픈)
- 김영한의 실전 자바 - 중급 2편 (오픈)
- 김영한의 실전 자바 - 고급편 (3분기 중 오픈 예정, 1, 2편 예정)

### 김영한의 실전 자바 - 고급편

- 람다, 스트림, IO, 네트워크, 멀티스레드

## 실전 데이터베이스 로드맵

- 자바 로드맵 이후

## 백엔드 개발자 로드맵 소개



### 김영한 백엔드 개발자 자바 스프링 JPA 실무 로드맵

백엔드 개발자 로드맵 소개 영상 링크: <https://youtu.be/ZgtvcyH58ys>

### 스프링 완전 정복 로드맵

- 스프링을 완전히 마스터 할 수 있는 로드맵
- URL: <https://www.inflearn.com/roadmaps/373>

### 스프링 부트와 JPA 실무 완전 정복 로드맵

- 최신 실무 기술로 웹 애플리케이션을 만들어보면서 학습
- URL: <https://www.inflearn.com/roadmaps/149>

## 하고 싶은 이야기

자바 백엔드 개발자가 되고 싶다면 반드시 스프링을 학습해야 함

**Q:** 스프링을 배우는 것과 백엔드 개발자가 무슨 관계?

### 실무 개발자가 되는 학습 방향

프로그래밍 언어를 배우는 3단계

- 기본 문법 및 개념 이해 - 자바 입문편
  - 기본 문법, 변수, 타입, 조건문, 반복문, 함수 사용법 등

- 고급 개념과 해당 언어의 라이브러리 활용 - **자바 기본편 ~ 고급편**
  - 객체지향 개념, 고급 문법, 자바의 주요 라이브러리
- 프레임워크 및 생태계 탐색 → **백엔드 개발자 로드맵**
  - 스프링 프레임워크를 포함한 다양한 오픈소스 학습
  - 실제 프로젝트나 업무에서의 적용에 중점
  - 어떤 프레임워크를 사용하고, 어떤 라이브러리를 조합해서 사용하는 것이 좋은지 등등
  - 개발 생태계 내에서의 베스트 프랙티스

스프링은 백엔드 생태계의 중심에서 오픈소스 생태계를 통합함

따라서 스프링을 학습하면 프레임워크 및 생태계의 베스트 프랙티스를 배울 수 있음

어떤 프레임워크를 사용하고, 어떤 라이브러리를 조합해서 사용하는 것이 좋은지 등등

스프링이 중심에서 다 통합을 하기 때문에 이런 부분을 자연스럽게 배움

**스프링을 배운다는 것은 사실 스프링만 배우는 것이 아니라, 실무 자바 백엔드 개발의 생태계 자체를 배우는 것**

#### **백엔드 개발자 로드맵 진행 방법**

- 자바 중급 2편까지 학습(최소 기준)
- 스프링 로드맵 진행
- 이후 자바 고급편 나오면 고급편 진행 후 스프링 로드맵 진행

#### **스프링 로드맵 학습시 주의사항**

스프링은 너무 거대함, 학습의 방향을 잘 잡아야 함

먼저 숲을 보고, 그 다음에 나무를 보는 방향으로 학습해야 함

- **스프링 입문 - 코드로 배우는 스프링 부트, 웹 MVC, DB 접근 기술(무료)로 시작**
  - 스프링은 너무 거대해서 처음부터 차례로 다 배우려면 코끼리(스프링) 다리만 보다가 스프링이 뭔지도 모르고 포기함
  - 일단 코끼리(스프링)가 뭔지 일단 알고 진행해야함
  - 이 강의는 코끼리(스프링)가 뭔지 대략적으로 이해하는 것이 목표
  - 너무 깊이있게 이해하려고 하지말고, 코드를 따라하면서 아~ 스프링이라는게 대략 이런 것이구나 정도만 느끼면 충분함
  - 익숙해 지는 것이 필요하기 때문에 이해가 어려워도 한 번 정도는 복습하며 코드를 따라하는 것 추천
  - 이 강의를 통해 스프링이 뭔지 큰 그림을 그리기만 하면 됨, 스프링의 세세한 내용은 로드맵의 다음 강의부터 하나씩 학습
- **스프링 핵심 원리 - 기본편**
  - 이 강의부터 스프링의 기초부터 하나씩 배움
- **HTTP, 스프링 MVC**
  - 스프링을 웹 환경에서 사용하는 방법

- **스프링 DB 접근 기술**

- 데이터베이스를 사용해서 스프링을 활용하는 방법
- 이 시점에 DB를 알면 더 좋지만, DB를 몰라도 진행할 수 있을 정도로 DB 관련 내용은 쉽게 구성되어 있음

### **좋은 공부 방법**

- 처음에는 코드를 따라한다 생각하고 처음부터 끝까지
- 복습할 때는 정리하면서 복습(당연히 코딩!)
- 이렇게 하면 전체적인 그림을 그린 상태에서 공부할 수 있음
- 많은 개발자가 복습하고 정리하는 단계에서 이해

## **하고 싶은 이야기 정리 링크**

### **개발 인생 전반의 이야기**

#### **EO 인터뷰 영상**

- 한국 개발자 최고 1타강사 김영한의 인생 [1부]: [https://youtu.be/\\_HTj5b59Em0](https://youtu.be/_HTj5b59Em0)
- 한국 개발자 최고 1타강사 김영한의 인생 [2부]: <https://youtu.be/MNyNRraMU8Y>

#### **개발바닥 - 시골 청년 개발왕 되다**

- 1편: <https://youtu.be/Pb69UQ6f8n0>
- 2편: <https://youtu.be/b4QP5RsuJts>
- 3편: <https://youtu.be/l0h1pQ96u2g>

### **취업과 이직에 대한 고민**

#### **인프론 - 어느 날 고민 많은 주니어 개발자가 찾아왔다, 성장과 취업, 이직 이야기**

- <https://youtu.be/QHlyr8soUDM>

#### **인프런 최초 20만 명 달성 기념 QA**

- <https://youtu.be/psXdWq008DA>

#### **인프런 최초 30만 명 달성 기념 QA**

- <https://inf.run/81ogv>