

spring 한스푼(4주차)

3주차 review / 스프링? /
프로젝트 시작 / 스프링 구조

프로젝트 시작.

<https://start.spring.io/> or 검색창 – start spring

–프로젝트 설정–

Project – gradle – groovy

Language – JAVA

Spring Boot – 3.1.4

Packaging – Jar

Java – 17

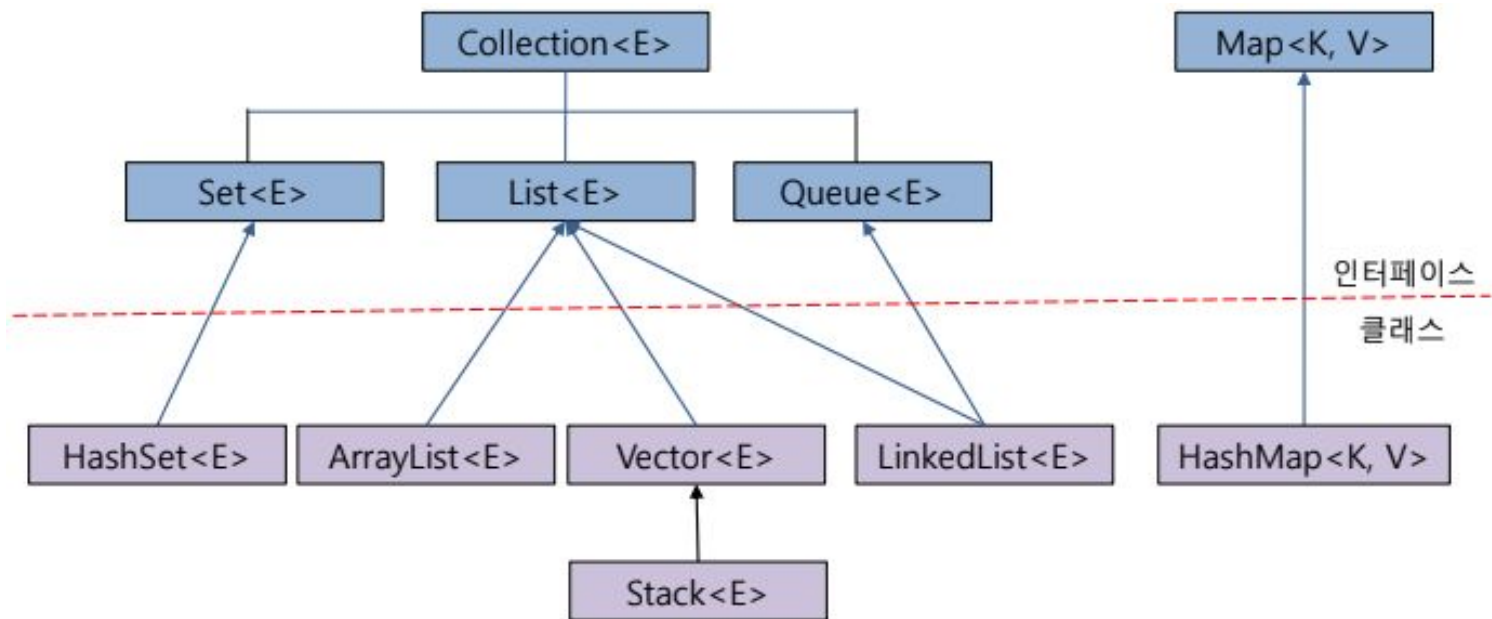
Dependencies – Spring web/Lombok/mysql/jap

review (3주차)

기본 클래스.

- java.lang 패키지
- Object 클래스

컬렉션



오류(예외 처리)

try-catch문으로 예외 처리를 한다.

```
public static void main(String[] args) {  
    int[] arr = new int[5];  
    try {  
        for (int i = 0; i <= 5; i++) {  
            arr[i] = i;  
            System.out.println(arr[i]);  
        }  
    } catch (ArrayIndexOutOfBoundsException e) {  
        System.out.println(e);  
        System.out.println("예외 처리 부분");  
    }  
}
```

```
orretto-17.0.8.1\bin\java.exe "-javaagent:C:\Program F
```

0

1

2

3

4

java.lang.ArrayIndexOutOfBoundsException: Index 5 out of bounds for length 5

예외 처리 부분

JAVA 익히기 (1)

Calc클래스를 만들어 계산기를 구현해보자.

Calc의 멤버 변수는 모두 private 되어있고 a,b로 구성되어있다.

메서드 add는 a와 b를 더한 값을 리턴하는 함수이다.

메서드 sub는 a에서 b를 뺀 값을 리턴하는 함수이다.

메서드 mul은 a와 b를 곱한 값을 리턴하는 함수이다.

메서드 div는 a에서 b를 나눈 값을 리턴하는 함수이고 오류가 발생되지 않게 예외가 처리되어 나눌 수 없다며 나누지 않고 끝난다.

메서드 setter는 멤버 a와 b를 값을 넣어주는 함수이다.

Calcmain 클래스를 만들어

메인 함수에서 100과 0을 각각 변수에 넣고 4개의 메서드를 실행.

JAVA 익히기 (2)

-계좌 관리 프로그램-

아래 사진같이 메인함수를 쓰고 실행창처럼 나올 수 있게
클래스를 만들어 실행해보자

```
6 ▶ public class BankMain {  
    4개 사용 위치  
7     public static List<Bank> list = new ArrayList<Bank>();  
8 ▶ public static void main(String[] args){  
9         Bank person1 = new Bank( name: "김철수", account_num: "1");  
10        list.add(person1);  
11        Bank person2 = new Bank( name: "홍길동", account_num: "2");  
12        list.add(person2);  
13        person1.deposit( amount: 10000);  
14        person1.withdraw( amount: 10001);  
15        person1.withdraw( amount: 10000);  
16        person1.remittance( num: "1", amount: 10000);  
17        person1.remittance( num: "2", amount: 10000);  
18        person1.deposit( amount: 10000);  
19        person1.remittance( num: "2", amount: 10000);  
20        person1.balance();  
21    }  
22 }
```

김철수님 10000원 입금 완료되었습니다. 잔액은 10000원 입니다
출금 할 수 없습니다. 금액을 다시 확인해 주세요.
김철수님 10000원 출금 완료되었습니다. 잔액은 0원 입니다
입금 서비스를 이용해주세요
송금 할 수 없습니다. 금액을 다시 확인해 주세요.
송금 할 수 없습니다. 금액을 다시 확인해 주세요.
김철수님 10000원 입금 완료되었습니다. 잔액은 10000원 입니다
홍길동님에게 10000원을 송금했습니다.
수신자를 찾을 수 없습니다
김철수님의 통장 잔고 금액은 10000원 입니다

Spring Boot



REST API??

REST (Representational State Transfer)

- HTTP URI(Uniform Resource Identifier)를 통해 자원(Resource)을 명시하고,
- HTTP Method(POST, GET, PUT, DELETE, PATCH 등)를 통해
- 해당 자원(URI)에 대한 CRUD Operation을 적용하는 것을 의미합니다.

REST 특징

- Server-Client(서버-클라이언트 구조)
- Stateless(무상태)
- Cacheable(캐시 처리 가능)
- Layered System(계층화)
- Uniform Interface(인터페이스 일관성)

하지만 API는 기계와 기계 대화하는 것. REST 특징에서 어긋난다.

1. URI는 동사보다는 명사를, 대문자보다는 소문자를 사용해야 한다.

ex `http://spring.onespoon.com/Runnig/` (x)

`http://spring.onespoon.com/run/` (o)

2. 마지막에 슬래시(/)를 포함하지 않는다.

ex `http://spring.onespoon.com/test/` (x)

`http://spring.onespoon.com/test` (o)

3. 언더바 대신 하이픈을 사용한다

ex `http://spring.onespoon.com/test_api` (x)

`http://spring.onespoon.com/test-api` (o)

4. 파일확장자는 URI에 포함하지 않는다

ex `http://spring.onespoon.com/api.jpg` (x)

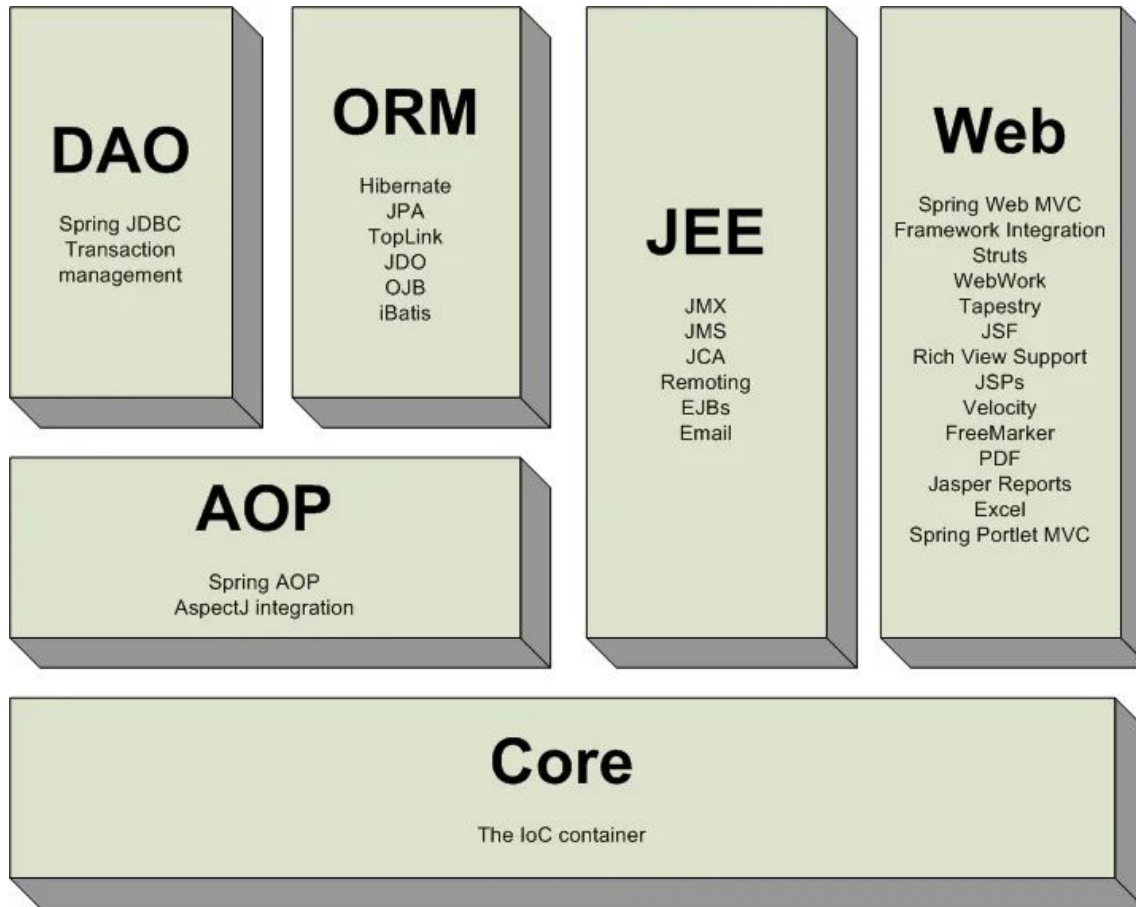
`http://spring.onespoon.com/api` (o)

5. 행위를 포함하지 않는다.

ex `http://spring.onespoon.com/delete-post/1` (x)

`http://spring.onespoon.com/post/1` (o)

Spring boot?



스프링 구조

- 의존성 주입(build.gradle)
 - 설정파일(application.properties)
 - 프로젝트 시작(???application)
 - 테스트(test 폴더)
-
- 파라미터(Controller)(ex : www.youtube.com/(영상주소))
 - 컨트롤러 서비스 구현(service)
 - 테이블관리(entity) (JDBC)
 - DB관리 (repository) (라이브러리 : JPA)
 - 보안설정 (security) (라이브러리 : spring security)
 - 그 외 (JWT, handling, ...)

의존성 주입(Beuild.gradle)

프로젝트에서 필요한 라이브러리가 있을 경우 추가하여 maven_repository에서 가져온다.

프로젝트에 빌드과정중 필요한 옵션이 있을 경우 코드를 추가하기도 한다.

예) web, DB, lombok, security, jwt, ...

의존성을 추가 했거나 제거시 꼭 마지막엔 코끼리 아이콘을 눌러 프로젝트 빌드 해준다.

설정 파일(application.properties)

프로젝트에 대한 설정 및 라이브러리에 대한 설정을 하는 곳이다.

추가로 git과 같은 프로젝트 업로드를 해선 안되는 비밀 설정들은 include하여 파일 설정을 추가로 가져와 구성하기도 한다. ex)DB 설정 및 security 설정

DEBUG 같이 레벨 설정을 하여 실행 창에서 DEBUG의 로그를 볼 수 있다.

어노테이션

프로그램 코드의 일부가 아닌 프로그램에 관한 데이터를 제공하고, 코드에 정보를 추가하는 정형화된 방법이다.

어노테이션을 사용하면 코드가 깔끔해지고 재사용이 가능하다.

라이브러리마다 다르기에 라이브러리를 검색하여 사용법을 찾아야한다.

ex) LOMBOK의 @Getter/@Setter (게터/세터)를 통해 해당 클래스의 멤버들을 자동으로 할당해줌.