

Spring Day 1

계층적 구조

계층적 구조 없이 한 곳에서 모든 작업을 한번에 처리하면

- 코드의 복잡성 증가
- 유지 보수의 어려움
- 유연성 부족
- 중복 코드 증가
- 낮은 확장성

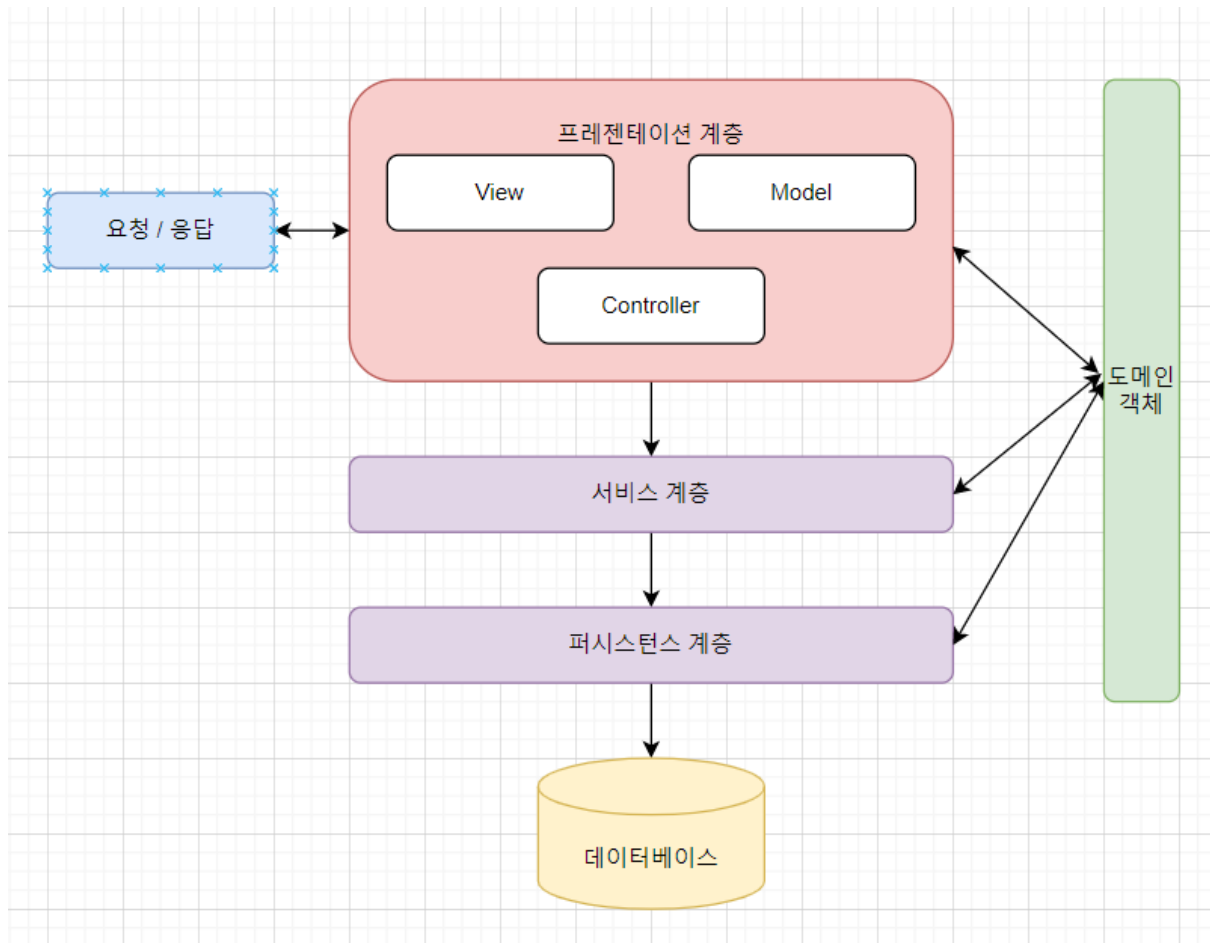
등의 문제가 발생할 수 있다

계층적 구조 종류

- 퍼시스턴스 persistence
- 서비스 service
- 프레젠테이션 presentation - [mvc](#)

으로 분리하여 사이트 구성

계층적 구조는 퍼시스턴스persistence 계층 서비스 service계층 프레젠테이션 presentation계층 으로 분리하여 사이트를 구성한다. mvc는 이중에서 프레젠테이션 계층에 속하게 된다.

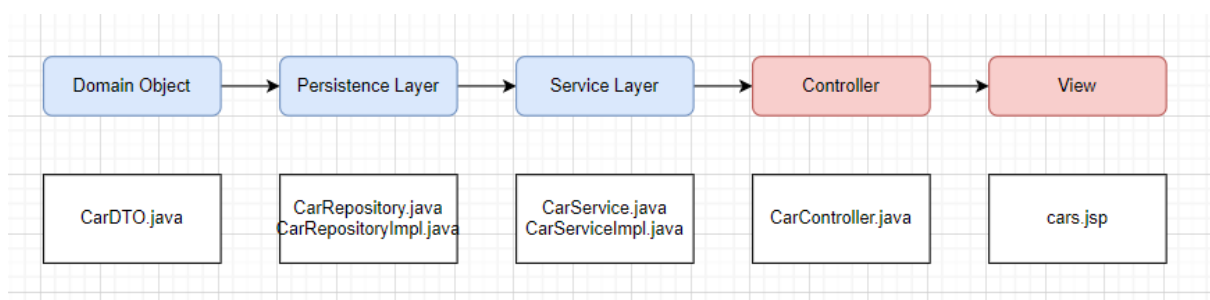


도메인 객체 : domain object 데이터 모델, 객체 정보를 저장한다. jsp 에서의 DTO와 같다.

퍼시스턴스 계층 : persistence layer 데이터 액세스 계층, 데이터베이스에 접근하여 데이터를 처리한다. jsp DAO 와 같다.

서비스 계층 : service layer 비즈니스 계층이라고도 하고 애플리케이션에서 제공하는 포괄적인 서비스를 표현한다. 프레젠테이션 계층과 퍼시스턴스 계층을 연결하는 역할을 한다.

프레젠테이션 계층 presentation layer 사용자에게 데이터를 입력 받거나 결과를 서버에 전달하고 사용자가에 보여주는 계층이다.



도메인 객체 (DTO)

```
package com.carshop.controller;

import lombok.AllArgsConstructor;
import lombok.Data;

@Data
@AllArgsConstructor
public class CarDTO {
    private String cid, cname, cprice, ccate, cdesc;

    CarDTO(){

    }
}
```

Interface

다형성

Animal.java

```
public class Animal {
    String name;

    void setName(String name) {
        this.name = name;
    }
}
```

Tiger.java

```
public class Tiger extends Animal implements Predator, Barkable{//상속, 구현
    public String getFood() {
        return "apple";
    }

    public void bark() {
```

```
        System.out.println("어흥");
    }
}
```

Predator.java

Interface

```
public interface Predator { //명령
    String getFood();
}
```

Barkable.java

Interface

```
public interface Barkable {
    void bark();
}
```

Bouncer.java

```
public class Bouncer {
    void barkAnimal(Barkable a) {
        a.bark();
    }

    // void barkAnimal(Animal animal) {
    //     if(animal instanceof Tiger) {
    //         System.out.println("어흥");
    //     } else if(animal instanceof Lion) {
    //         System.out.println("으르렁");
    //     } else if(animal instanceof Croc) {
    //         System.out.println("짹짹");
    //     } else if(animal instanceof Leop) {
    //         System.out.println("카악");
    //     }
    // }
}
```

ZooKeeper.java

```
public class ZooKeeper {

    void feed(Predator a) {
        System.out.println("feed " + a.getFood());
    }

    // void feed(Tiger tiger) {
    //     System.out.println("feed apple");
    // }
    //
    // void feed(Lion lion) {
    //     System.out.println("feed banana");
    // }
    //
    // void feed(Croc croc) {
    //     System.out.println("feed strawberry");
    // }
    //
    // void feed(Leop leop) {
    //     System.out.println("feed orange");
    // }
}
```

Sample.java

```
public class Sample {

    public static void main(String[] args) {

        ZooKeeper zooKeeper = new ZooKeeper();
        Tiger tiger = new Tiger();
        Lion lion = new Lion();

        // zooKeeper.feed(tiger);
        // zooKeeper.feed(lion);

        Bouncer bouncer = new Bouncer();
        bouncer.barkAnimal(tiger);
        bouncer.barkAnimal(lion);
    }

}
```

/cars

Car List

localhost:8080/cars

CarShop 홈 차량 보기 게시판

Search

차량 보기 심현민

c0001	c0002	c0003
소나타	그랜저	아반떼
2500 만원	3500 만원	2000 만원

Home Pricing FAQs About

© 2023 CarShop

/boards

Board List

localhost:8080/boards

CarShop 홈 차량 보기 게시판

Search

게시판 심현민

안녕하세요	첫 글	Hi
처음 뵙겠습니다	첫 글입니다	Hello
작성자1	작성자2	작성자3
2023.02.27	2023.02.27	2023.02.27

Home Pricing FAQs About

© 2023 CarShop

