

Java Programming II

Lab4

514770-1

Fall 2025

10/1/2025

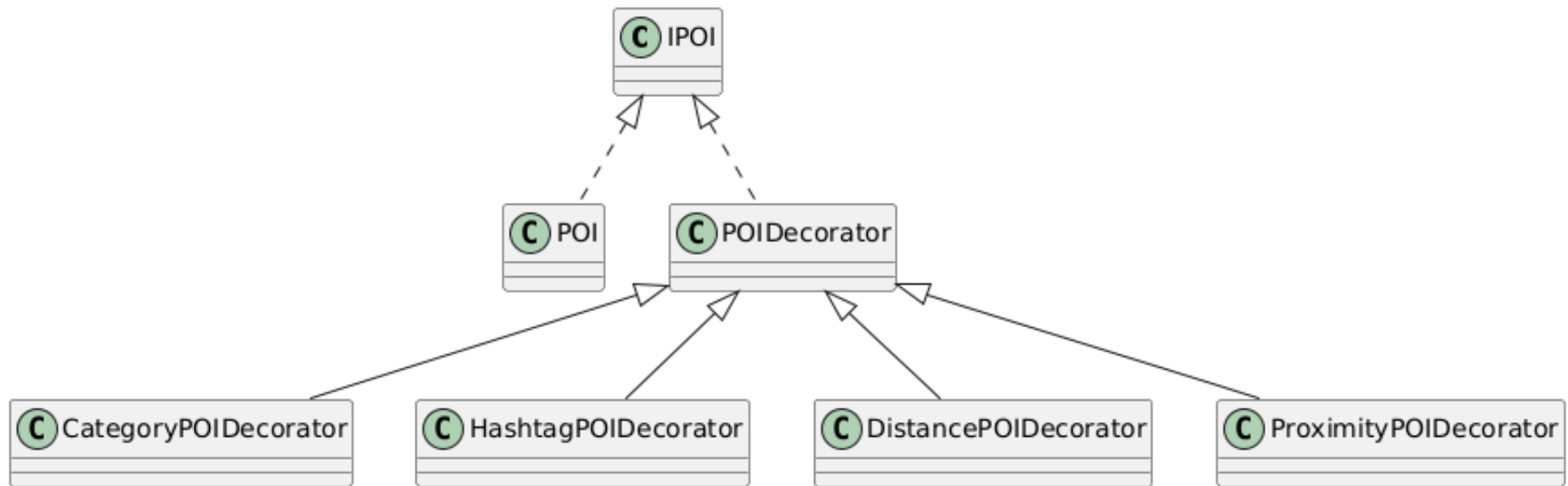
Kyoung Shin Park
Computer Engineering
Dankook University

Lab4

- ❑ Decorator 패턴을 사용하여 POI에 데코레이더로 처리한다.
 - **IPOI** 인터페이스는 **String getInformation()**를 갖는다.
 - **POIDecorator** 추상 클래스는 implements **IPOI** 하고 IPOI decoratedPOI 멤버를 갖는다. 그리고 정적 유틸리티 함수 POI unwrapPOI(IPOI node) 갖는다.
 - **POI** 클래스는 implements **IPOI** 을 하고 getInformation()에서 정보를 제공한다.
 - **CategoryPOIDecorator, HashtagPOIDecorator, DistancePOIDecorator, ProximityPOIDecorator** 클래스는 extends **POIDecorator** 하고 다음같이 처리한다.
 - ❑ String category 정보 추가
 - ❑ String... tags 정보 추가
 - ❑ Location currentLocation에 따른 거리(distance) 정보 추가
 - ❑ Location currentLocation, double radiusMeter에 따른 근접(within) 정보 추가

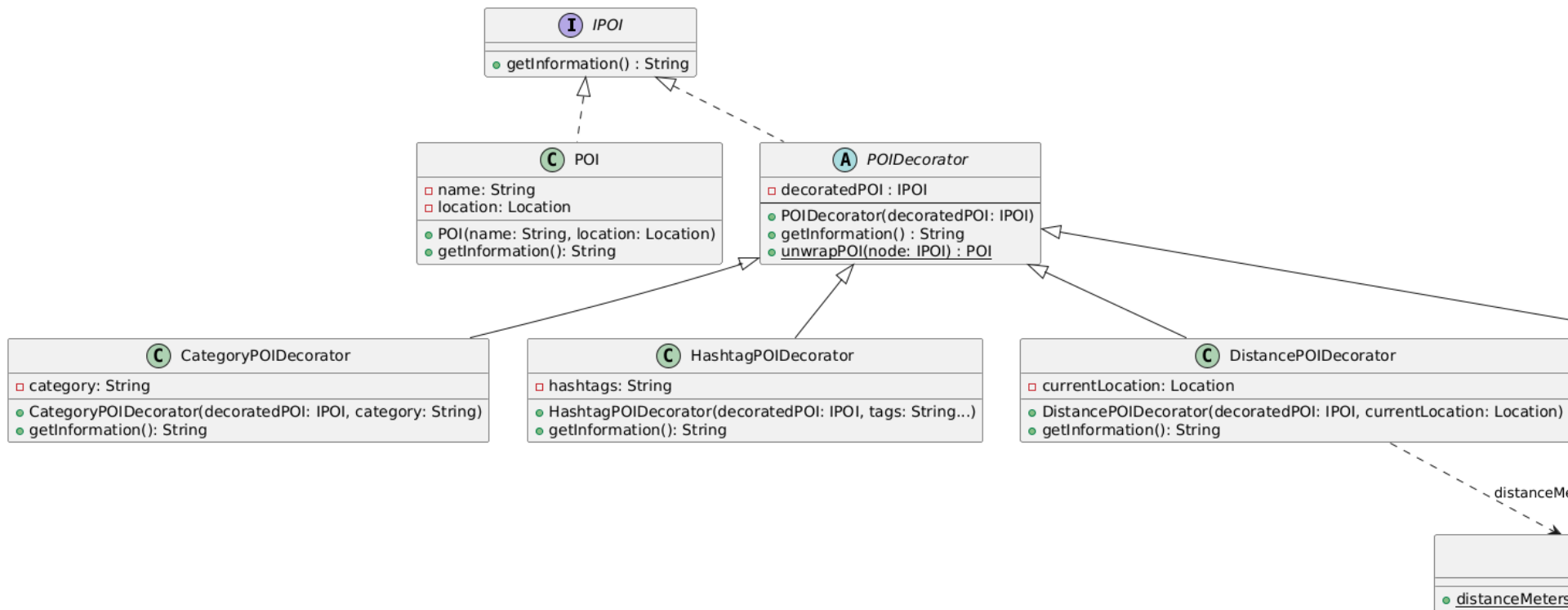
Lab4

▣ Class Diagram



Lab4

□ Class Diagram



Lab4

□ POIGsonFileLoader 클래스

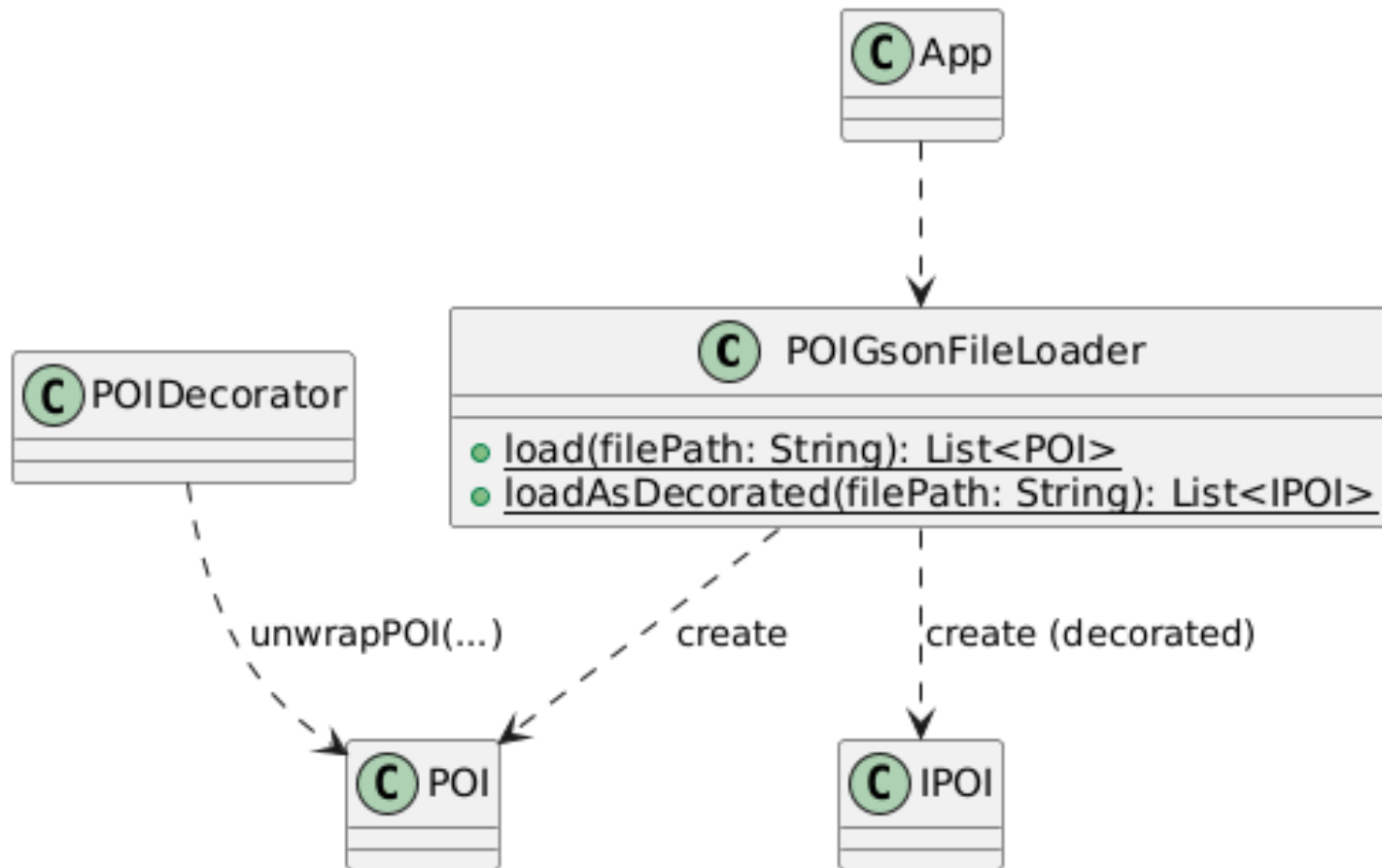
- **List<POI> load(String filePath)** 정적메소드는 기존 POI 리스트를 반환해준다.
- **List<POI> loadAsDecorated(String filePath)** 정적메소드는 기존 POI (name, Location)에 category와 hashtags 추가해서 IPOI 리스트를 반환해준다.

□ GeoUtil 클래스

- **double distanceMeters(Location a, Location b)** 정적메소드는 DistancePOIDecorator와 ProximityPOIDecorator 등에서 사용

Lab4

▣ Class Diagram



Lab4

```
public class App {  
    public static void main(String[] args) throws Exception {  
        // list of poi (NewPOI.json)  
        List<POI> pois = POIGsonFileLoader.load("NewPOI.json");  
        pois.forEach(System.out::println);  
  
        // category & hashtag decorator (used in  
        POIGsonFileLoader.loadAsDecorated method)  
        List<IPOI> decoratedBase =  
        POIGsonFileLoader.loadAsDecorated("NewPOI.json");  
        System.out.println("== Base decorated (from JSON) ==");  
        decoratedBase.forEach(d -> System.out.println(" - " +  
        d.getInformation()));
```

Lab4

```
Location[] path = new Location[] {
    new Location(37.5779, 126.9760),
    new Location(37.5784, 126.9766),
    new Location(37.5789, 126.9772),
    new Location(37.5795, 126.9787),
    new Location(37.5797, 126.9794),
    new Location(37.5800, 126.9800)
};
System.out.println("\n== Along path ==");
for (Location cur : path) {
    System.out.printf("\n-- step @ (%.5f, %.5f)%n",
cur.getLat(), cur.getLon());
    for (IPOI baseChain : decoratedBase) {
        // distance & proximity 데코레이터 적용해서 출력
    }
}
}
```

Submit to e-learning

- ▣ Lab4 과제에 yourcode 를 추가 (yourcode 없을시 10점에
서 -1점 감점)
- ▣ **Java25-2-HW4-YourID-YourName.zip** 과제(보고서에
반드시 yourcode 설명 포함)를 e러닝에 제출 (**due by
10/14**).