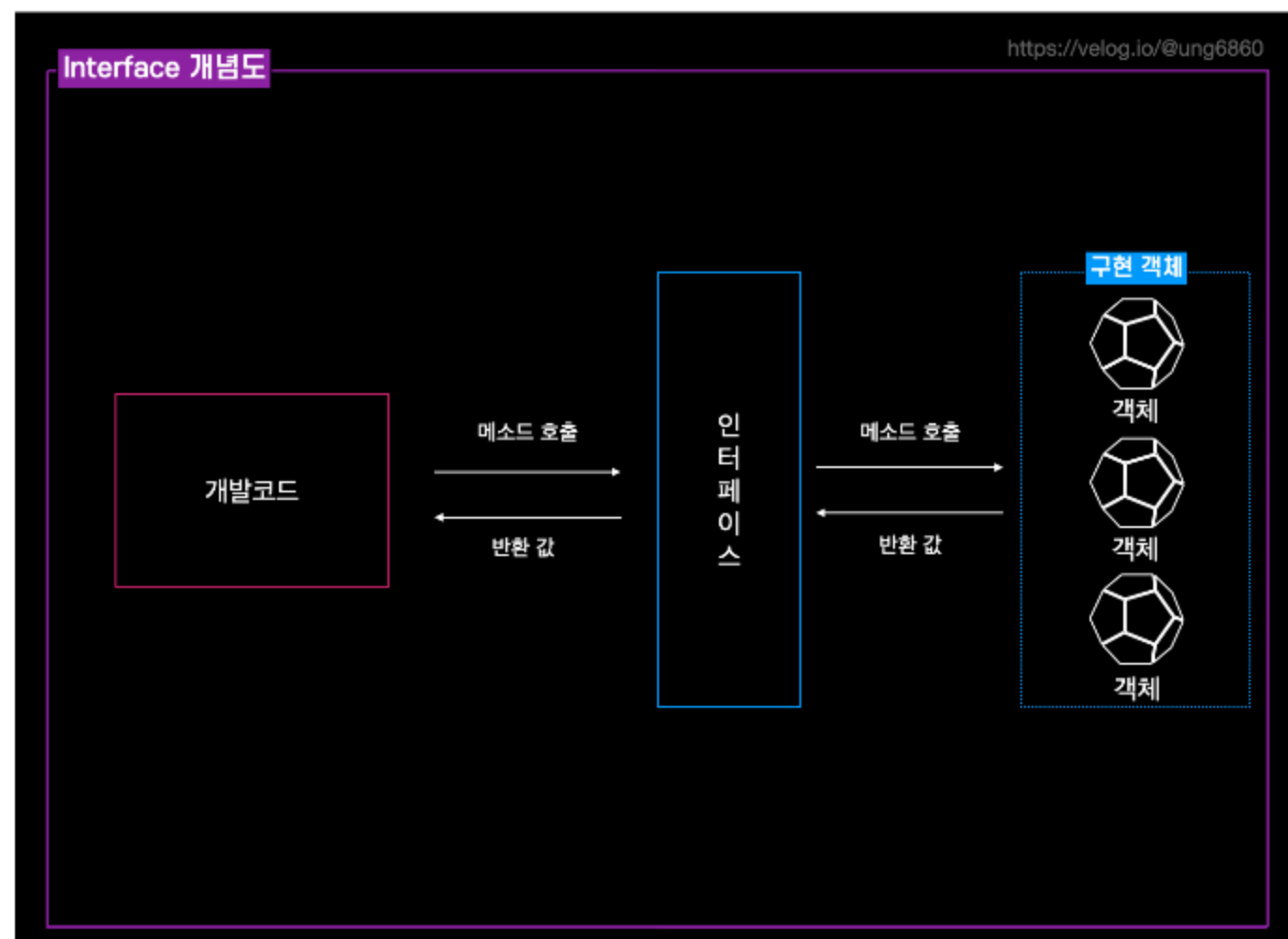


42주차 발표_민서연

인터페이스

01 인터페이스

다른 클래스를 작성할 경우 기본이 되는 틀을 제공하면서 다른 클래스 사이의
중간 매개 역할 까지 담당하는 추상 클래스



```
//인터페이스 선언
public interface Animal{
    //추상 메소드 선언
    public void move();
}
```

```
//Dog 클래스 선언
public class Dog implements Animal{
    //Animal의 추상 메소드 구현
    @Override
    public void move(){
        System.out.println("개가 움직입니다.");
    }
}

//Dog 클래스 선언
public class Cat implements Animal{
    //Animal의 추상 메소드 구현
    @Override
    public void move(){
        System.out.println("고양이가 움직입니다.");
    }
}
```

01 인터페이스

필요성

클래스는 특정 클래스의 상태나 동작의 범위에서 크게 벗어나지 않거나 거의 유사한 기능을 수행하는 경우

➡ 상속 사용

수행해야하는 동작은 유사하지만 다양한 성격의 클래스가 필요할 경우 각 클래스들을 묶어주어야 하는 경우

➡ 인터페이스 사용

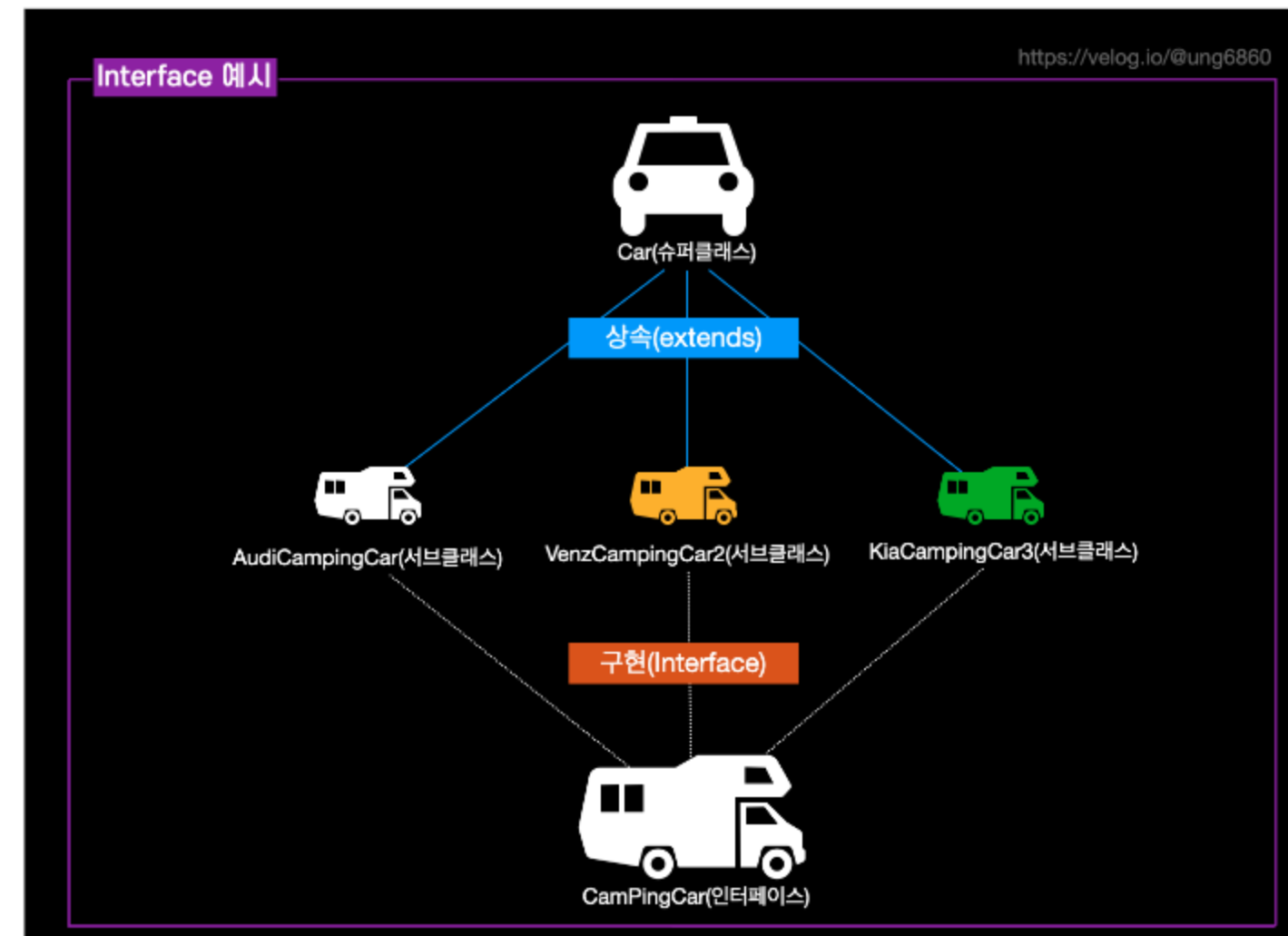
01 인터페이스

필요성

대규모 프로젝트 개발 시 일관되고 정형화 된 개발을 할 수 있는 표준화 기능이다.

클래스의 작성과 인터페이스 구현을 동시에 할 수 있으며 더 유연한 코드작성과 개발 시간이 단축 된다.

클래스와 클래스 간의 관계를 인터페이스로 연결하여 클래스마다 독립적인 프로그래밍 기능을 한다.



01 인터페이스

기본 문법

상수와 추상 메서드 보유 가능

필드

: 모든 필드를 public static final로 선언하며 모든 필드는 상수

추상 메서드

: abstract 키워드는 생략 가능.

선언 할 때 interface 키워드를 붙여서 만들고 객체를 구현할 때 implements 키워드를 통해 객체들을 구현하는 용도로 사용한다.

02 인터페이스와 추상클래스

추상 클래스

- 추상 클래스는 하위 클래스들의 공통점을 모아 추상화하여 만든 클래스다.
- 추상 클래스는 다중 상속이 불가능하고 단일 상속만 허용한다.
- 추상 메서드 외에 일반 클래스와 같이 일반적인 필드, 메서드, 생성자를 가질 수 있다.
- 추상화를 하면서 중복되는 클래스 멤버들을 통합 및 확장 할 수 있다.

02 인터페이스와 추상클래스

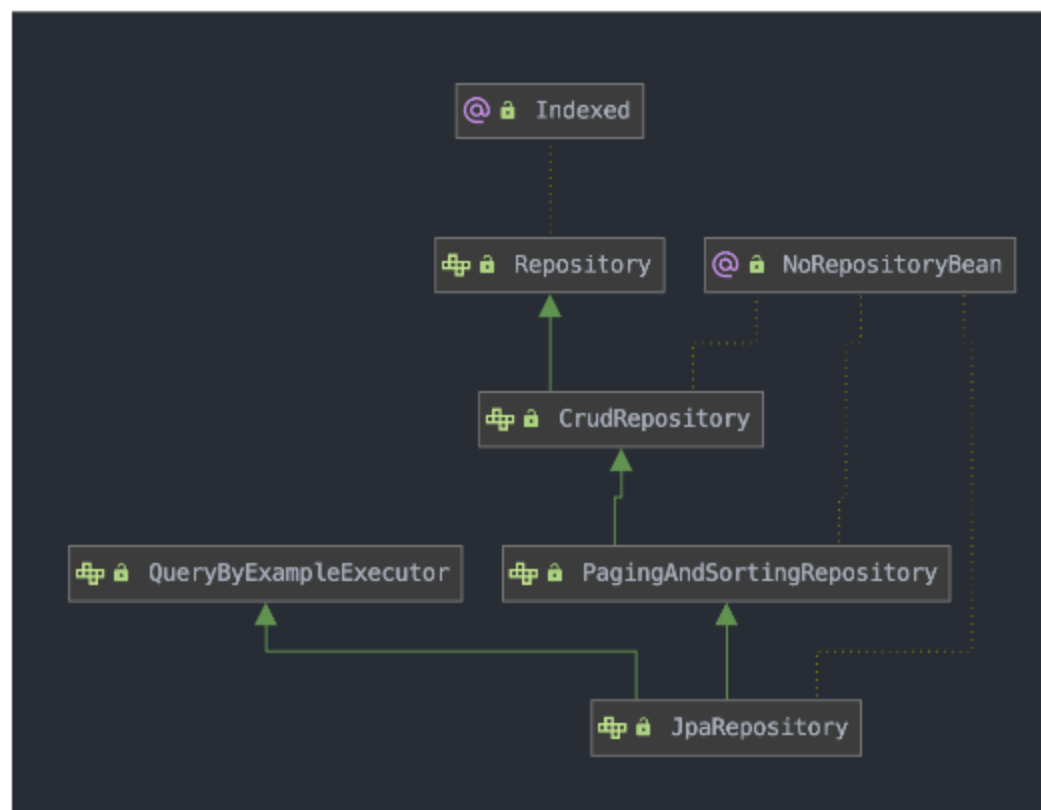
	추상 클래스	인터페이스
사용 키워드	abstract	interface
사용 가능 변수	제한 없음	static final (상수)
사용 가능 접근 제어자	제한 없음 (public, private, protected, default)	public
사용 가능 메소드	제한 없음	abstract method, default method, static method, private method
상속 키워드	extends	implements
다중 상속 가능 여부	불가능	가능 (클래스에 다중 구현, 인터페이스 끼리 다중 상속)
공통점	1. 추상 메소드를 가지고 있어야 한다. 2. 인스턴스화 할 수 없다 (new 생성자 사용 X) 3. 인터페이스 혹은 추상 클래스를 상속받아 구현한 구현체의 인스턴스를 사용해야 한다. 4. 인터페이스와 추상클래스를 구현, 상속한 클래스는 추상 메소드를 반드시 구현하여야 한다.	

03 JPA 인터페이스

Spring Data JPA는 간단한 CRUD 기능을 공통으로 처리하는 인터페이스를 제공한다.

```
public interface UserRepository extends JpaRepository<User, Long>
```

조회, 입력, 수정, 삭제 등과 같은 단순 반복 작업들을 Spring Data JPA 구현체인 Hibernate가 애플리케이션 실행 시점에 동적으로 자주 사용되는 쿼리 집합을 만들어 Repository 인터페이스를 구현해준다.



다양한 기능이 정의된 인터페이스를 상속 받고 있기 때문에 단순 CRUD 외에 Paging, Sorting과 같은 기능을 수행할 수 있다.

03 Resource 인터페이스

java.net.URL을 추상화한 인터페이스

스프링 내부에서 가장 많이 사용하는 인터페이스이며 IoC 컨테이너가 생성 될 때 설정 정보를 담는 파일들을 가져올 때도 사용한다.

```
public interface Resource extends InputStreamSource {  
    // 리소스가 있는지 있는지 확인  
    boolean exists();  
  
    // Stream이 열렸는지 체크  
    boolean isOpen();  
  
    // 해당 리소스의 url을 가져온다  
    URL getURL() throws IOException;  
  
    // 해당 리소스의 파일을 가져온다  
    File getFile() throws IOException;  
  
    // 상대 경로를 통해서 다른 리소스를 가져옴  
    Resource createRelative(String relativePath) throws IOException;  
  
    // 파일 이름을 가져온다  
    String getFilename();  
  
    // 설명을 가져온다  
    String getDescription();  
}
```

03 Resource 인터페이스

인터페이스 구현체

UrlResource

URL을 기준으로 리소스를 읽어들인다.

http, https, ftp, file, jar 프로토콜을 명시해주고 해당 리소스의 위치를 알려주는 URL 방식을 통해서 리소스의 위치를 알려주는 방식이다.

```
Resource resource = new UrlResource("file:{절대경로}");
```

ClassPathResource

지원하는 접두어가 classpath: 일때 클래스 패스를 기준으로 리소스를 읽어 들인다.

Class Loader를 통해서 ClassPath에서 리소스를 찾는 방식이다.

```
ClassPathResource resource = new ClassPathResource("dao.xml");
```

03 Resource 인터페이스

인터페이스 구현체

ServletContextResource

웹어플리케이션 루트에서 상대 경로로 리소스를 찾는다.

FileSystemResource

파일 시스템을 기준으로 읽는다.

```
Resource resource = new FileSystemResource({절대경로});
```

04 출처

<https://velog.io/@ung6860/JAVA%EC%9D%B8%ED%84%B0%ED%8E%98%EC%9D%B4%EC%8A%A4Interface%EC%97%90-%EB%8C%80%ED%95%B4-%EC%95%8C%EC%95%84%EB%B3%B4%EC%9E%90-fc81k7rr>

<https://coding-factory.tistory.com/867>

<https://inpa.tistory.com/entry/JAVA-%E2%98%95-%EC%9D%B8%ED%84%B0%ED%8E%98%EC%9D%B4%EC%8A%A4Interface%EC%9D%98-%EC%A0%95%EC%84%9D-%ED%83%84%ED%83%84%ED%95%98%EA%B2%8C-%EA%B0%9C%EB%85%90-%EC%A0%95%EB%A6%AC>

<https://velog.io/@hoyun7443/JPA%EC%9D%98-%EA%B3%B5%ED%86%B5-%EC%9D%B8%ED%84%B0%ED%8E%98%EC%9D%B4%EC%8A%A4-JpaRepository%EC%9D%98-%EA%B8%B0%EB%8A%A5%EA%B3%BC-%EA%B5%AC%EC%A1%B0>

<https://engkimbs.tistory.com/entry/Spring-%EC%8A%A4%ED%94%84%EB%A7%81-Resource-%EC%B6%94%EC%83%81%ED%99%94-Spring-Resource-Abstraction>

<https://always-develop.tistory.com/37>