

2025년 상반기 K-디지털 트레이닝

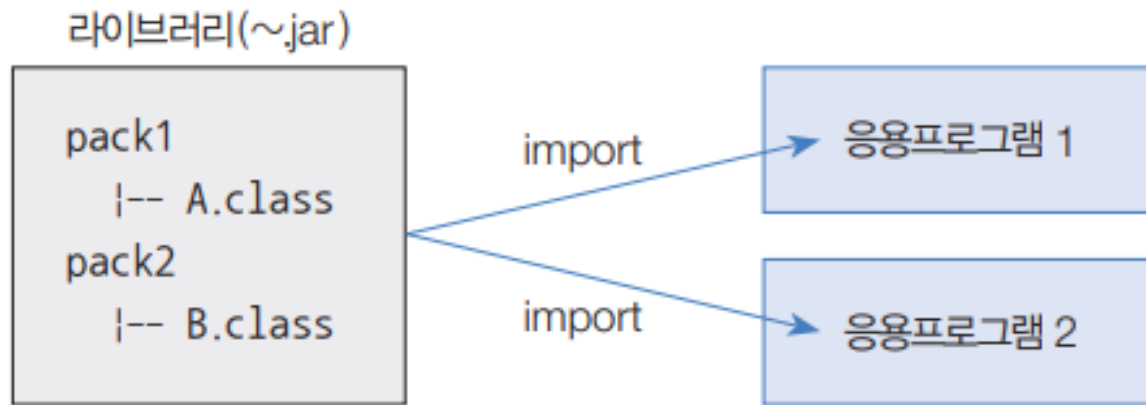
라이브러리와 모듈

[KB] IT's Your Life

1 라이브러리

✓ 라이브러리 추가하기

- 프로그램 개발 시 활용할 수 있는 클래스와 인터페이스들을 모아놓은 것
- 일반적으로 JAR 압축 파일(~.jar) 형태. 클래스와 인터페이스의 바이트코드 파일(~.class)들이 압축



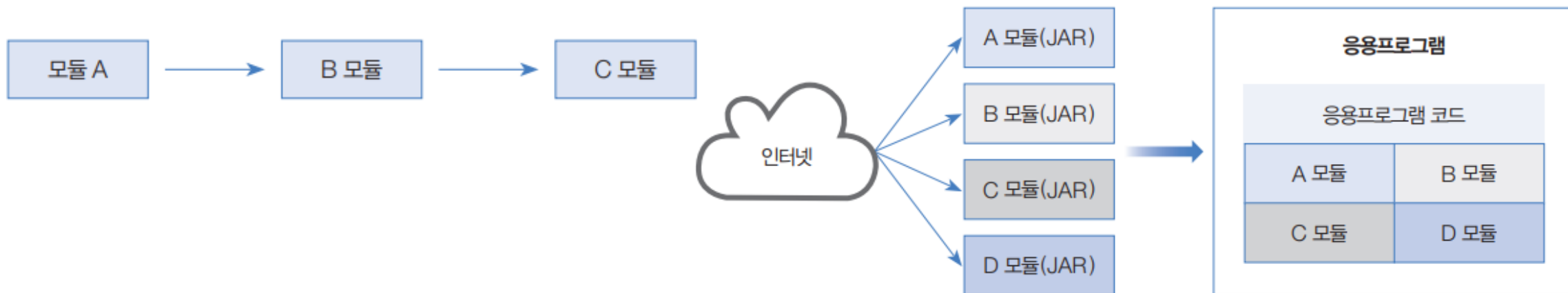
- 라이브러리 JAR 파일을 사용하려면 ClassPath(클래스를 찾기 위한 경로)에 추가
- 콘솔(명령 프롬프트 또는 터미널)에서 프로그램을 실행할 경우: java 명령어를 실행할 때 -classpath로 제공. 또는 CLASSPATH 환경 변수에 경로 추가
- 이클립스 프로젝트에서 실행할 경우: 프로젝트의 Build Path에 추가

✓ 모듈(라이브쇼핑 동영상 → 모듈별 개발)

- 패키지 관리 기능까지 포함된 라이브러리. Java 9부터 지원
- 모듈은 일부 패키지를 은닉하여 접근할 수 없게끔 할 수 있음



- 의존 모듈을 모듈 기술자(module-info.java)에 기술할 수 있어 모듈 간 의존 관계를 파악하기 쉬움
- 대규모 응용프로그램은 기능별로 모듈화해서 개발. 재사용성 및 유지보수에 유리



4 모듈 배포용 JAR 파일

✓ 모듈 배포용 JAR 파일 생성

라이브방송용 모듈
live.jar

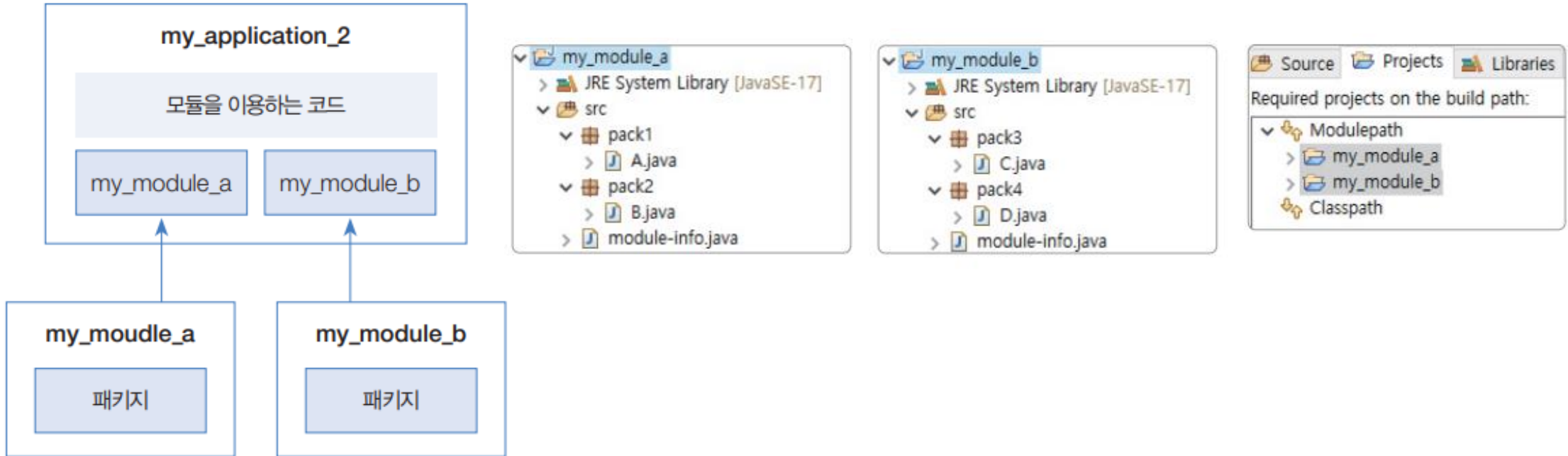
쇼핑몰용 모듈
shop.jar

관리자용 모듈
manager.jar

3 응용프로그램 모듈화

✓ 모듈화

- 모듈화: 응용프로그램을 기능별로 서브 프로젝트(모듈)로 쪼개어 다음 조합해서 개발



- 응용프로그램의 규모가 클수록 협업과 유지보수 측면에서 모듈화 유리
- 다른 응용프로그램서도 재사용 가능

4 모듈 배포용 JAR 파일

✓ 모듈 배포용 JAR 파일 생성

- 다른 모듈에서 쉽게 사용할 수 있게 바이트코드 파일(.class)로 구성된 배포용 JAR 파일을 모듈별로 따로 생성할 수 있음

```
>>> Main.java

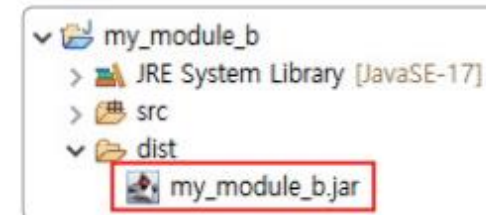
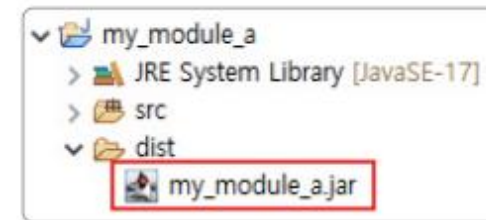
1  package app;
2
3  import pack1.A;
4  import pack2.B;
5  import pack3.C;
6
7  public class Main {
8      public static void main(String[] args) {
9          //my_module_a 패키지에 포함된 A 클래스 이용
10         A a = new A();
11         a.method();
12
13         //my_module_a 패키지에 포함된 B 클래스 이용
14         B b = new B();
15         b.method();
16
17         //my_module_b 패키지에 포함된 C 클래스 이용
18         C c = new C();
19         c.method();
20     }
21 }
```

my_module_a 모듈에서 가져옴

my_module_b 모듈에서 가져옴

my_module_a 모듈의 클래스 이용

my_module_b 모듈의 클래스 이용

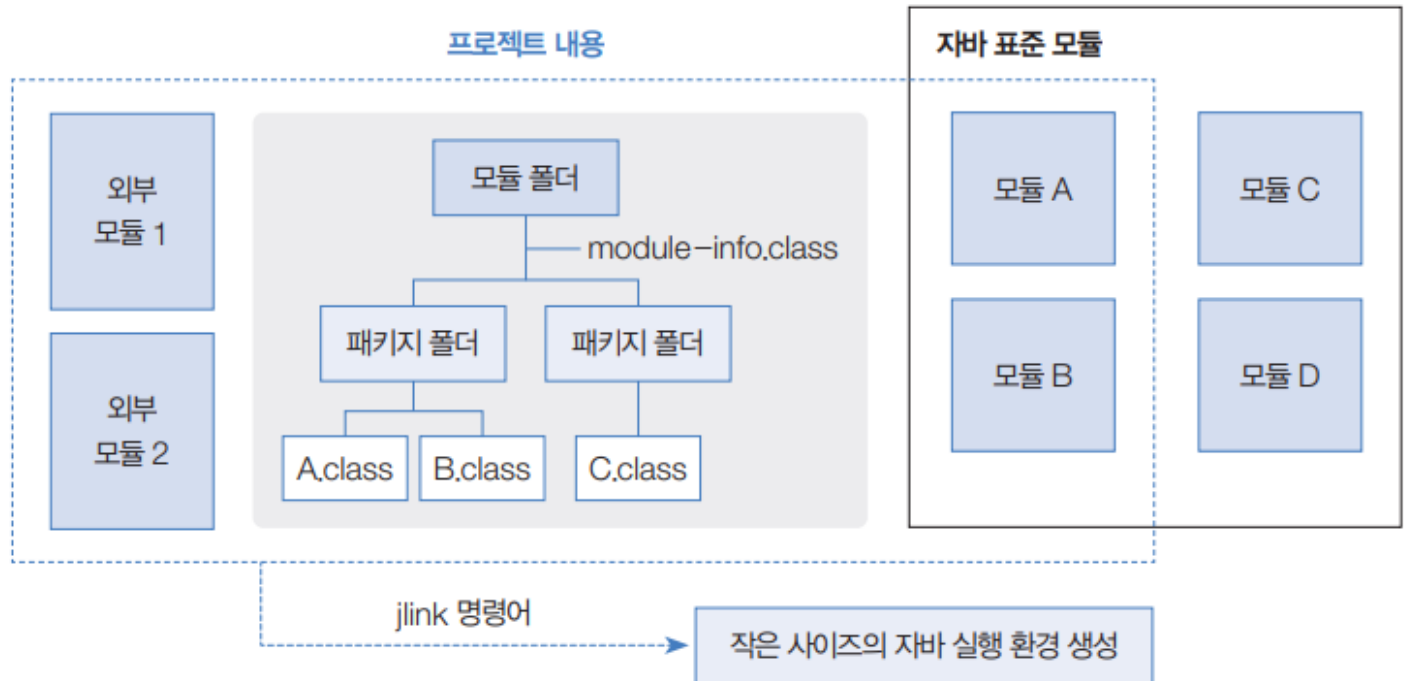
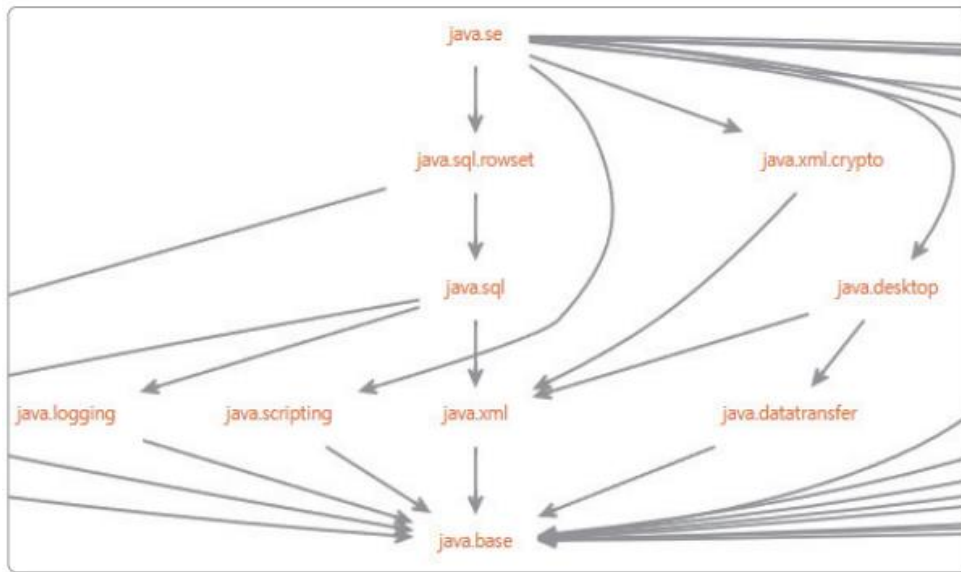


9 자바 표준 라이브러리

✓ 표준 라이브러리

- JDK가 제공하는 표준 라이브러리는 Java 9부터 모듈화됨
- 응용프로그램을 실행하는 데 필요한 모듈만으로 구성된 작은 사이즈의 자바 실행 환경(JRE)
- Java 17의 전체 모듈 그래프(화살표는 모듈간의 의존 관계를 표시)

<https://docs.oracle.com/en/java/javase/17/docs/api/java.se/module-summary.html>



File Edit View Navigate Code Refactor Build Run Tools Git Window Help

study [build] v

Project

- study C:\fullstack\java\study
 - .gradle
 - .idea
 - build
 - classes
 - java
 - main
 - inherit
 - inherit2
 - object
 - object2
 - test1
 - test2
 - generated
 - libs
 - study-1.0-SNAPSHOT.jar
 - tmp
 - gradle
 - out
 - production
 - classes
 - src
 - main
 - java
 - inherit
 - HomePhone
 - Phone
 - SmartPhone
 - SmartPhoneUse
 - inherit2

Gradle

- study
 - Tasks
 - build
 - assemble
 - build
 - buildDependents
 - buildNeeded
 - classes
 - clean
 - jar
 - testClasses
 - build setup
 - documentation
 - help

Run study [build] x

```
> Task :compileTestJava NO-SOURCE
> Task :processTestResources NO-SOURCE
> Task :testClasses UP-TO-DATE
> Task :test NO-SOURCE
> Task :check UP-TO-DATE
> Task :build

BUILD SUCCESSFUL in 18s
```

api-gateway

- common-service
 - gradle/wrapper
 - src
 - build.gradle
- elastic
 - .env
 - compose.yml
 - elasticsearch.Dockerfile
 - filebeat.Dockerfile
 - filebeat.yml
 - kibana.yml
 - logstash.conf
- gradle
 - lambda-image-resize
 - node_modules
 - index.js
 - package-lock.json
 - package.json
- reserve-service
 - gradle/wrapper
 - src