



# PGL-SpringBoot

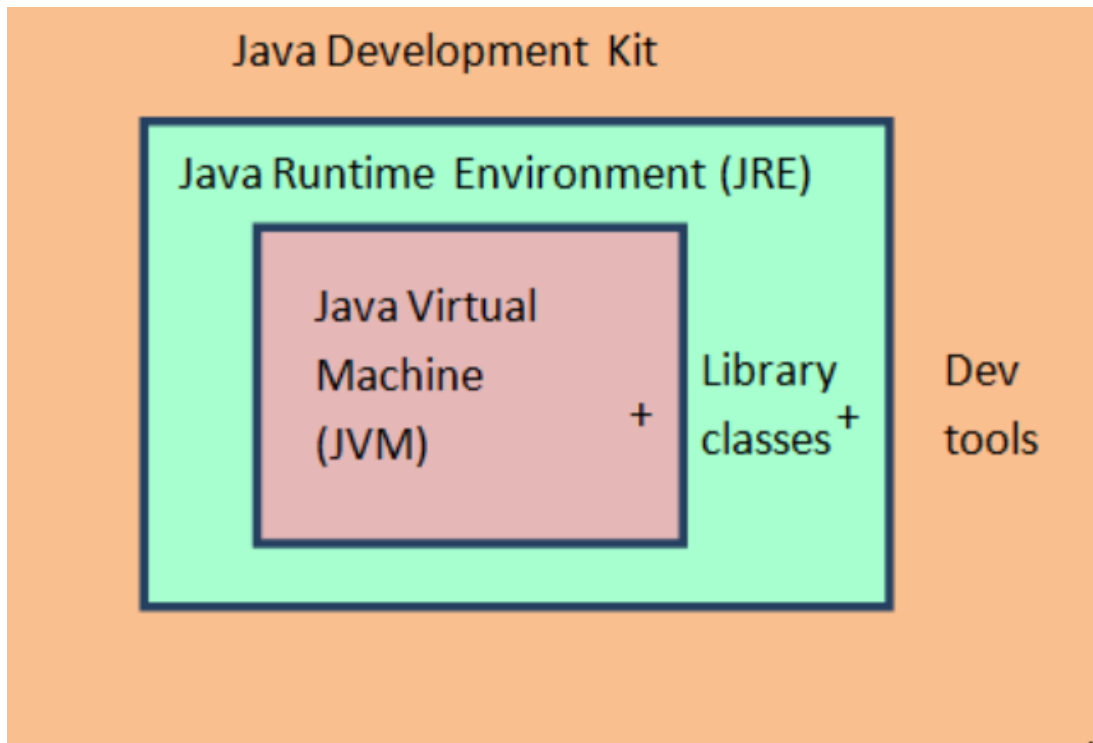
▼ 상태	진행 중
🕒 작성일시	@2022년 1월 1일 오후 1:07

## JDK의 개념

- JAVA = High Level 언어 (인간 친화적 언어)
- JAVA 도 컴퓨터가 이해하기 쉬운 언어로 바뀌어야 함
- 변경과정에서 알아두어야 할 것 : JVM - JRE - JDK

**\*\*JVM \*\*: Java Virtual Machine**

- JAVA 기계어를 컴퓨터가 이해할 수 있도록 JAVA Byte Code로 변경한다.
  - **JRE** : Java Runtime Environment
    - JVM 이 바뀐 Byte code 들 중에서 컴퓨터가 알아듣지 못하는 아이가 존재 가능
    - JVM은 언어를 Bytecode로 바꾸는 역할만 하지, 라이브러리들을 다루고 변환하는 능력은 없기 때문! (ex) 다른 라이브러리, 다른 외부에서 불러져 온 아이들
    - 따라서 JRE라는 개념 속에 JVM이 존재하게 되는 것
    - JRE 에는 JVM과 JVM이 사용하는 라이브러리 등을 포함
  - **JDK** : Java Development Kit
    - JAVA로 개발 시에 필요한 것들을 모아둔 키트
    - JAVA 코드를 JAVA ByteCode로 만드는 과정 = 컴파일 (기계화 알아들을 수 있도록 변환)
    - JDK 개념 안에 JRE, JVM이 다 포함되어 있음
- 이미지출처



- 이때 JAVA ByteCode를 만드는데 JAVA만 사용할 필요는 없음 (ex : 코틀린)

## JDK 설치방법

1. <https://jdk.java.net/> 접속한다.

2. 버전 11 선택

- 하위 버전이더라도 상위호환 가능

# jdk.java.net

*Java Development Kit builds, from Oracle*

**Ready for use: JDK 17, JMC 8**

**Early access: JDK 19, JDK 18, Loom, Metropolis  
& Valhalla**

**Reference implementations: Java SE 17, 16, 15,  
11, 10, 9, 8, & 7**

### 3. 자신의 운영체제에 맞는 시스템 설치 (윈도우 / 리눅스)

## Java Platform, Standard Edition 11 Reference Implementations

The official Reference Implementation for Java SE 11 (JSR 384) is based solely on open-source code available from the [JDK 11 Project](#) in the [OpenJDK Community](#). This Reference Implementation applies to both the Final Release of JSR 384 (Sep 2018) and Maintenance Release 1 (Mar 2019).

The binaries are available under the [GNU General Public License version 2](#), with [Classpath Exception](#).

### ***These binaries are for reference use only!***

These binaries are provided for use by implementers of the Java SE 11 Platform Specification and are for reference purposes only. This Reference Implementation has been approved through the Java Community Process. Production-ready binaries under the GPL are available from [Oracle](#); and will be in most popular [Linux distributions](#).

### **RI Binary (build 11+28) under the GNU General Public License version 2**

- [Linux/x64 Java Development Kit \(sha256\)](#) 178.9 MB
- [Windows/x64 Java Development Kit \(sha256\)](#) 178.7 MB

### **RI Source Code**

The source code of the RI binaries is available under the [GPLv2](#) in a single [zip file \(sha256\)](#) 178.1 MB.

## 4. 설치 완료 후

내 PC > 바탕 화면 > IntelliJ IDEA Community Edition 2021.3			
이름	수정한 날짜	유형	크기
bin	2021-12-22 오후 4:02	파일 폴더	
jbr	2021-12-22 오후 4:02	파일 폴더	
lib	2021-12-22 오후 4:02	파일 폴더	
license	2021-12-22 오후 4:01	파일 폴더	
plugins	2021-12-22 오후 4:02	파일 폴더	
redist	2021-12-22 오후 4:02	파일 폴더	
build.txt	2021-11-27 오후 4:25	텍스트 문서	1KB
LICENSE.txt	2021-11-27 오후 4:25	텍스트 문서	12KB
NOTICE.txt	2021-11-27 오후 4:25	텍스트 문서	1KB
product-info.json	2021-11-27 오후 4:29	JSON 원본 파일	1KB

## Code Editor vs IDE

- 언어마다 문법이나 어휘가 조금씩 다르다는 특징이 존재
- JDK : JAVA 코드를 실행하기 위한 프로그램
- Code Editor, IDE : JAVA 코드를 효율적으로 작성하기 위한 도구 (문법적인 부분 점검)

## 1) Code Editor

- 대표적인 코드 에디터 : 비주얼 스튜디오 코드, Vim 등
- 프로그래밍 언어에 따라 문법 검증 및 편집 편의성(코드 작성 편리하게) 제공

## 2) IDE

- 대표적인 IDE : Visual Studio, Xcode, IntelliJ 등
- 사용하는 언어 및 프레임워크에 따라서 다양한 기능을 제공, 단순 문법, 편집 기능 제공하는 것 외에 프레임워크에 적합한 편리한 기능을 제공

## IntelliJ IDEA Community Edition (IDE)

- 언어를 사용해서 실행을 하는 등의 여러가지 기능 존재 => 그만큼 코드가 매우 무겁다
- 기본에는 HTML, CSS, JS 등의 도움은 제공 X => VSC로 도우자

# Docker

## 1. Docker 기초

- 컨테이너 기반의 가상화 기술
- 컨테이너는 안에 내용물은 상관 x, 배에 올리기 위해서는 컨테이너의 형태로 되어 있어야
- OS 위에 컨테이너를 올려서 어플리케이션을 활용
- 이전 가상화 기술 (이전 버전은 Hypervisor, 컨테이너 안에 guest os 포함되어있음)

## 왜 docker을 사용할까?

- Spring Boot로 프로그램을 제작할 때 관계형 데이터베이스인 MySQL, redis (캐시 기능), RabbitMQ 등 이런애들은 OS 별로 설정이 다 다르다
- 웹 어플리케이션 개발에 필요한 다양한 소프트웨어의 설치를 간소화

## 2. Docker Desktop 설치

### 2-1 ) DOCKER 설치 페이지로 들어가기

# Get Started with Docker

We have a complete container solution for you – no matter who you are and where you are on your containerization journey.



## Docker Desktop

Developer productivity tools and a local Kubernetes environment.

Download for Windows



## Docker Hub

Cloud-based application registry and development team collaboration services.

- 도커 설치 페이지
- 자신의 운영체제에 맞는 것을 골라서 설치해주기
- 아래는 설치 중인 화면

## Docker Desktop 4.4.3

### Installing...

Deploying component: Install required Windows components for WSL 2  
Deploying component: Add user to docker-users group  
Deploying component: Create docker-users group  
Installing components  
Unpacking file: System.Xml.XPath.XDocument.dll  
Unpacking file: System.Xml.XPath.dll  
Unpacking file: System.Xml.XmlSerializer.dll  
Unpacking file: System.Xml.XmlDocument.dll  
Unpacking file: System.Xml.XDocument.dll  
Unpacking file: System.Xml.ReaderWriter.dll  
Unpacking file: System.Web.Http.Owin.dll  
Unpacking file: System.Web.Http.dll  
Unpacking file: System.ValueTuple.dll  
Unpacking file: System.Threading.Timer.dll

- 윈도우 사용자 중에서

Docker Desktop - Install WSL 2 kernel update



#### WSL 2 installation is incomplete.



The WSL 2 Linux kernel is now installed using a separate MSI update package. Please click the link and follow the instructions to install the kernel update: <https://aka.ms/wsl2kernel>.

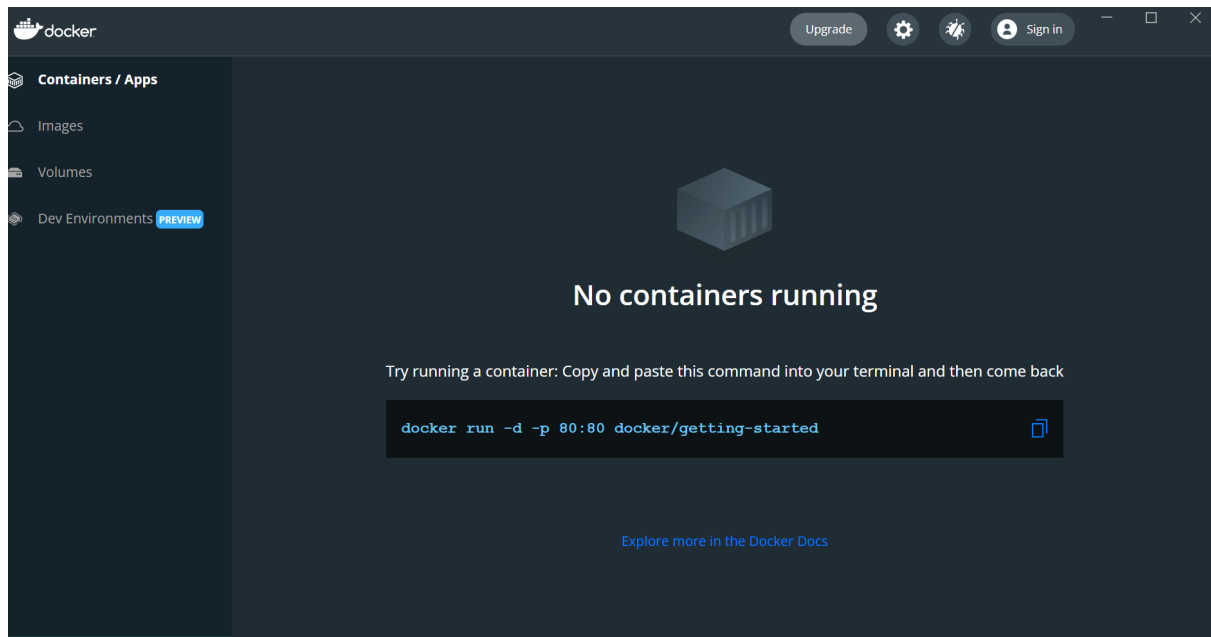
Press Restart after installing the Linux kernel.

Restart

Cancel

위와 같은 에러가 나는 경우도 많이 목격할텐데  
나는 아래의 블로그 글을 참고해서 해결했다.

[윈도우 도커 설치 에러 해결법](#)



### 3. Docker 테스트

- 잘 설치됐는지를 파악하기 위해서 cmd 창을 열고 `docker --help`, `docker --version` 이라고 쳤을 때, 관련 명령어들이 잘 나온다면 설치가 잘된 것이다.



```

C:\Users\DONGYUN>docker --version
Docker version 20.10.12, build e91ed57

C:\Users\DONGYUN>docker --help

Usage:  docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Options:
  --config string      Location of client config files (default
                        "C:\Users\DONGYUN\docker")
  -c, --context string  Name of the context to use to connect to the
                        daemon (overrides DOCKER_HOST env var and
                        default context set with "docker context use")
  -D, --debug           Enable debug mode
  -H, --host list       Daemon socket(s) to connect to
  -l, --log-level string Set the logging level
                        ("debug"|"info"|"warn"|"error"|"fatal")
                        (default "info")
  --tls                Use TLS; implied by --tlsverify
  --tlscacert string    Trust certs signed only by this CA (default
                        "C:\Users\DONGYUN\docker\ca.pem")
  --tlscert string      Path to TLS certificate file (default
                        "C:\Users\DONGYUN\docker\cert.pem")
  --tlskey string       Path to TLS key file (default
                        "C:\Users\DONGYUN\docker\key.pem")
  --tlsverify           Use TLS and verify the remote
  -v, --version         Print version information and quit

Management Commands:
  builder      Manage builds
  buildx*      Docker Buildx (Docker Inc., v0.7.1)
  compose*     Docker Compose (Docker Inc., v2.2.3)
  config       Manage Docker configs
  container    Manage containers
  context       Manage contexts

```

- 간단한 이미지를 받고 사용하기

```
docker run -d -p 80:80 docker/getting-started
```

- `docker` : Docker Daemon에 요청 보내기
- `run` : 컨테이너 실행
- `d` : Detached - 컨테이너를 background에서 실행하는 옵션
- `p` : Port, 컨테이너의 포트를 물리 서버의 포트와 연결
- `docker/getting-started` : 컨테이너로 실행할 이미지의 이름

=> 이후 127.0.0.1:80 으로 들어가주면 8000에 도커에 대한 설명이 있는 사이트가 나타날 것

=> docker rm -f ~ 위에 실행하고 나온 문자열들

- docker에서 새로운 어플리케이션을 도커로 실행하는 명령어

```
docker run -d -p 80:80 docker/getting-started
```

## Spring Boot

- Spring Boot 는 웹어플리케이션 프레임워크이다.

## HTML, CSS ,JS

- 인터넷의 주축
  1. 브라우저에 표시될 내용을 기술하는 언어
  2. 표현되는 형식을 정의하는 언어
  3. 브라우저에 동작을 제공하는 언어=> 인터넷이 사용하는 기술들

## Frontend , Backend

1. Frontend : 브라우저에 보이는 것을 중심으로 작업
2. Backend : 상황에 맞는 로직과 데이터를 다루는 개발자  
-앞단으로부터 요청이 오면 이에 적절한 데이터를 응답으로 돌려주는 역할 (내용 채워주는 역할)

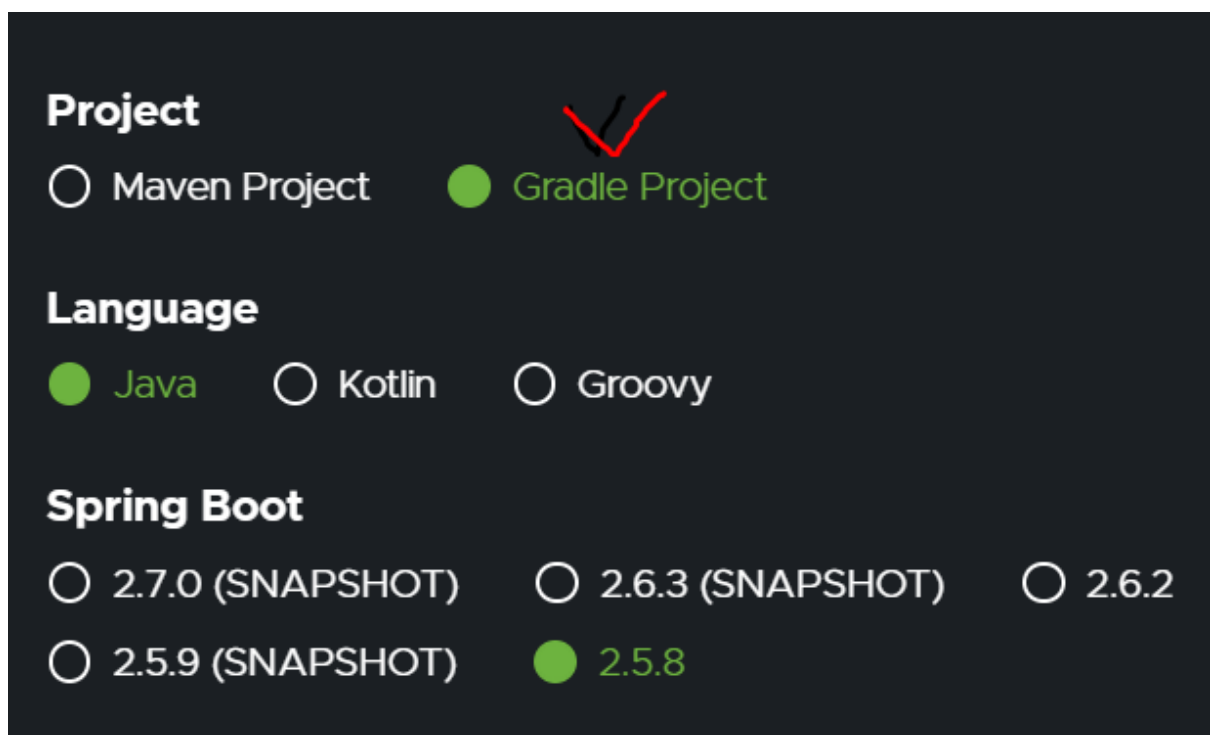
## Client , Server

- 인터넷 : 원격으로 연결된 수많은 컴퓨터들이 이루는 네트워크
- 우리 페이지에서 요청을 보낼 시에 인터넷을 통해서 서버컴퓨터, 혹은 다른 컴퓨터들에게 요청 보내지기 가능
- 이때 HTTP 는 어떤 형태로 요청을 보내고, 응답을 받을 지에 대한 규칙을 성립해 놓은 것
- 스프링은 어떤 역할?  
=> 서버 프로세스에 해당

- 서버 프로세스 : 인터넷에서 요청을 받고 적절한 응답 돌려줄 프로그램을 애로 제작
- 스프링 부트를 공부한다는 것
  - => HTTP 요청 등을 받아오면
  - => 요청에 따른 응답을 돌려주는
  - => Web Application을 만드는 framework(특정한 목적을 위한 도구, 라이브러리의 모음)

## Spring (Boot) initializer

Spring Boot initializer



The image shows a dark-themed form for creating a Spring Boot project. It has three sections: 'Project', 'Language', and 'Spring Boot'. In the 'Project' section, 'Maven Project' is unselected and 'Gradle Project' is selected with a green circle and a red checkmark. In the 'Language' section, 'Java' is selected with a green circle, while 'Kotlin' and 'Groovy' are unselected. In the 'Spring Boot' section, version '2.5.8' is selected with a green circle, while '2.7.0 (SNAPSHOT)', '2.6.3 (SNAPSHOT)', '2.6.2', and '2.5.9 (SNAPSHOT)' are unselected.

- gradle 프로젝트에 체크해주기
- initializer에 들어가서 프로젝트 이름 생성 시, 도메인 이름의 역순으로

## Project Metadata

**Group**  
com.example

---

**Artifact**  
code name

---

**Name**  
project name

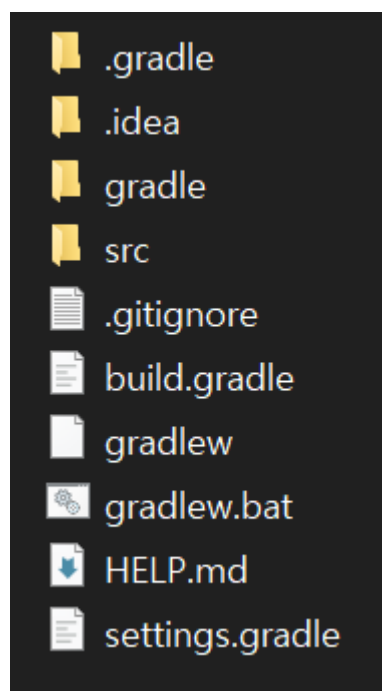
---

**Description**  
explanation

---

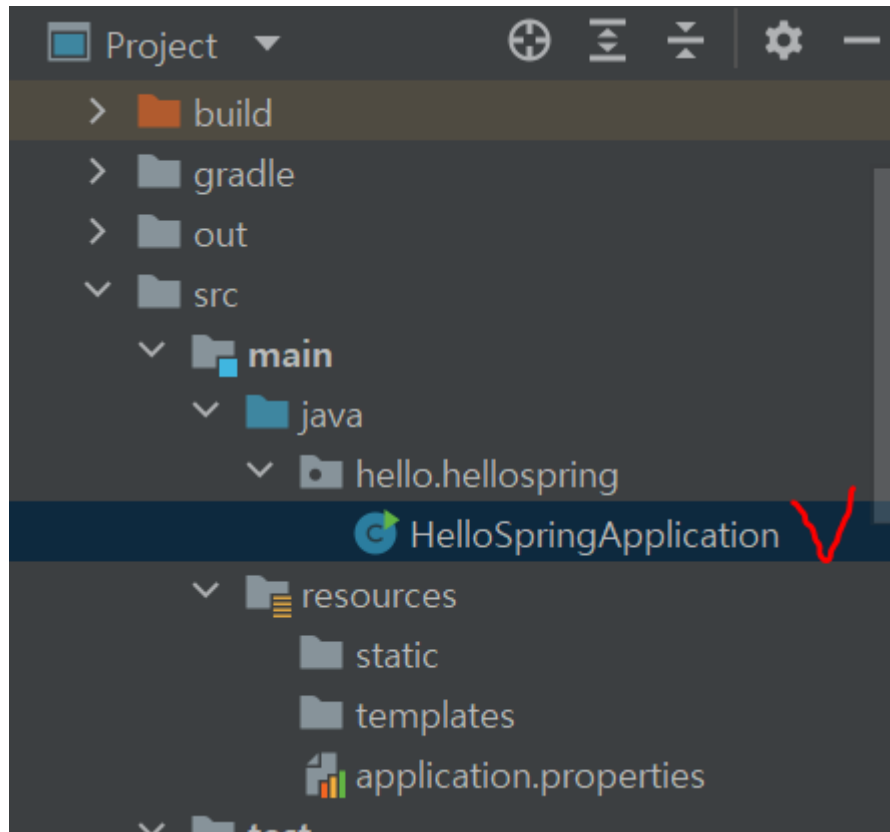
GENERATEEXPLORESHARE...

- generate 완료 후 파일 확인 시 아래와 같음

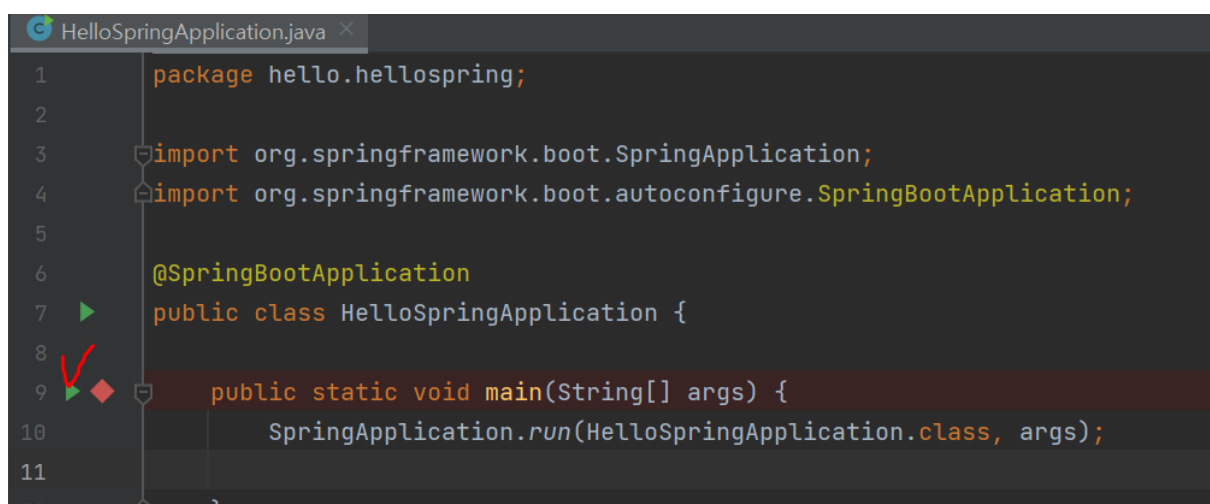


- intelliJ 접속해서 new project -> 우리가 다운받은 파일 열기

## Application 실행

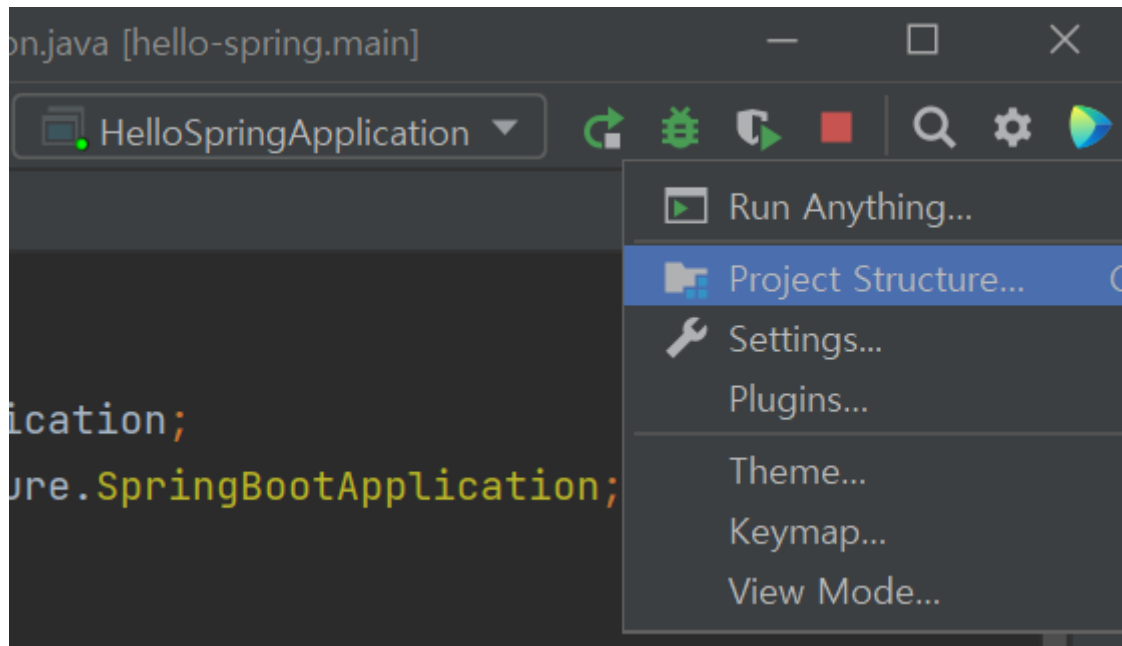


=> src > main > java > 내가설정한그룹이름 > 프로젝트이름

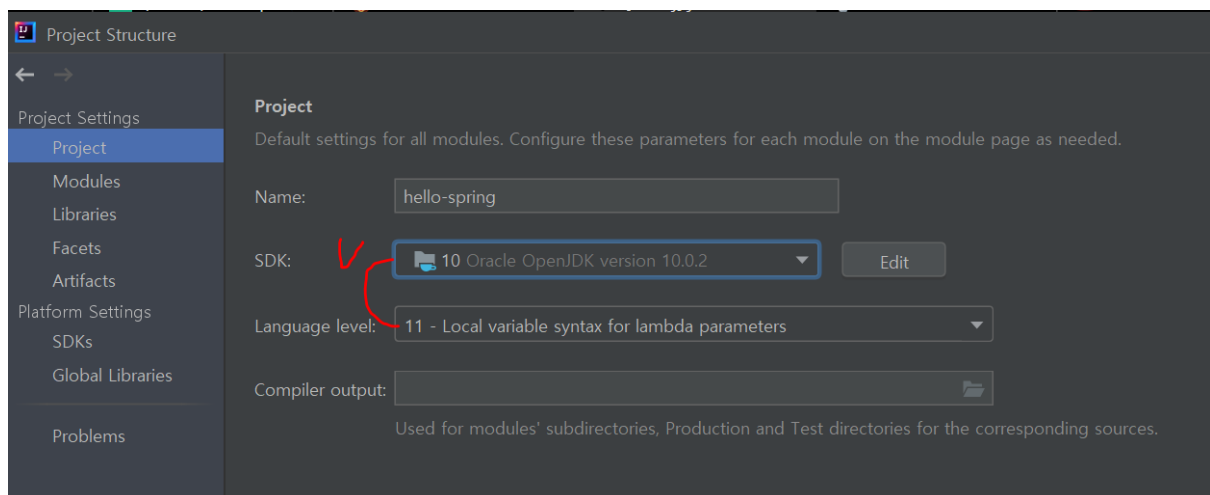


=> 초록색 세모를 누르면 잘 실행됨

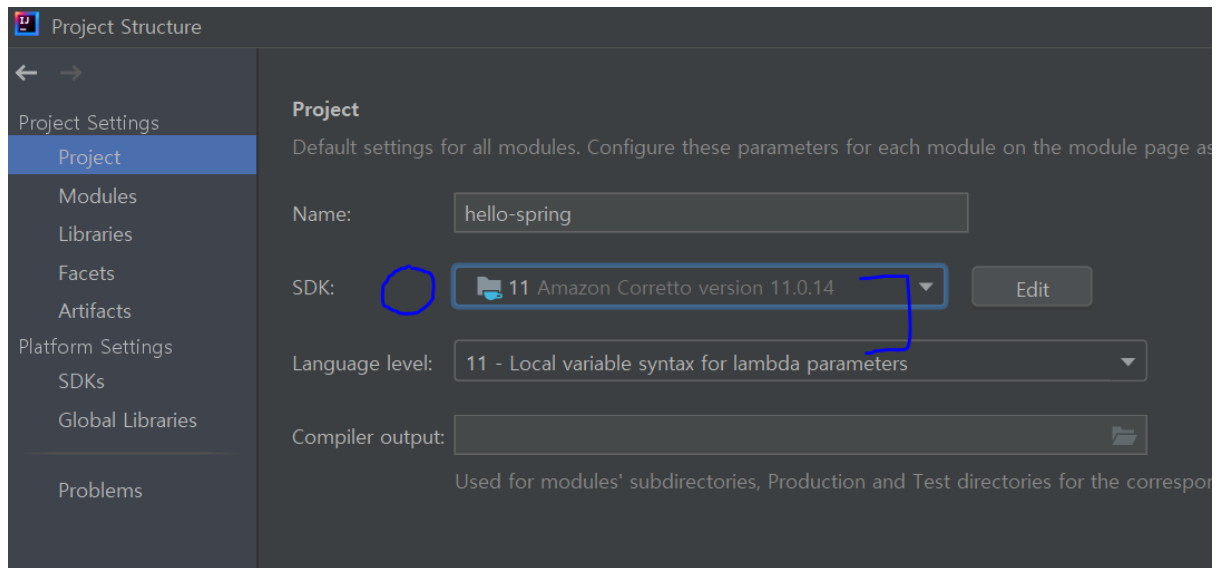




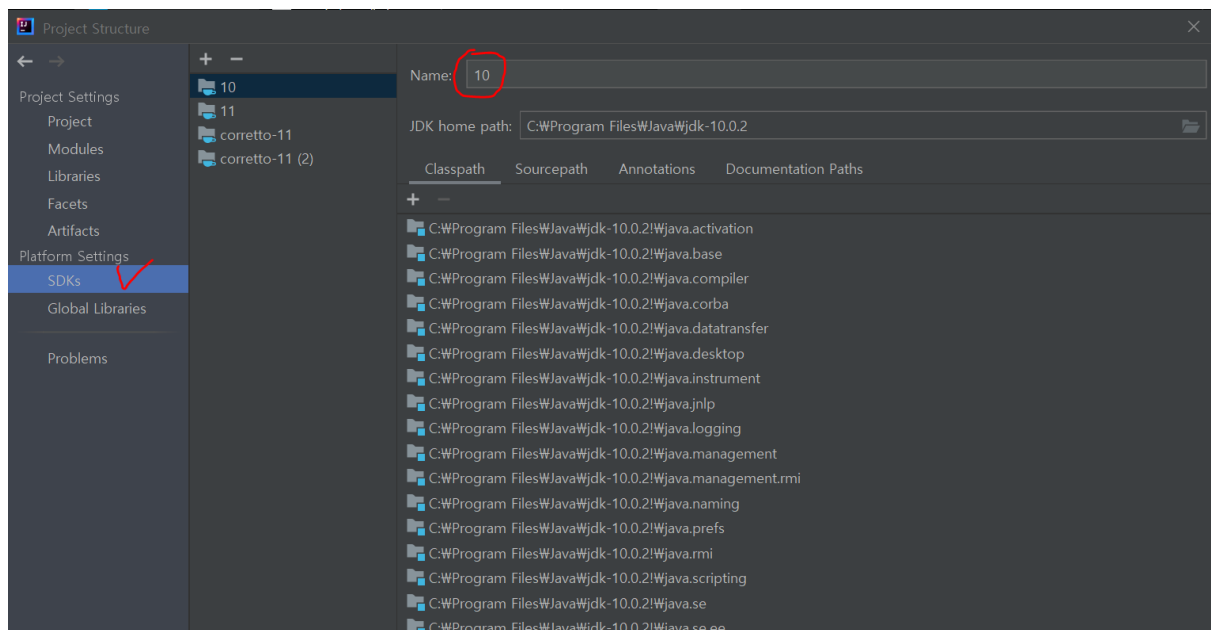
1. 나같은 경우에는 체크해놓은 부분을 보면 알 수 있듯이 sdk버전(10)과 language level(11)이 서로 다른 상황이었음
  - 따라서 sdk 버전을 변경해주면 된다



1. 둘이 버전이 동일하도록 맞춰주기 & 밑에 ok 누르기



## 1. 그 다음엔 밑에 sdk 부분 클릭



- sdk 가 10으로 여전히 맞춰진 상태, 11로 변경하자  
만약 11 다운로드 받아두지 않았다면 밑에 밑에 보이는 Download JDK 를 통해서 11버전 다운받으면 됨  
(name 직접 작성 & path도 11로 선택)



