

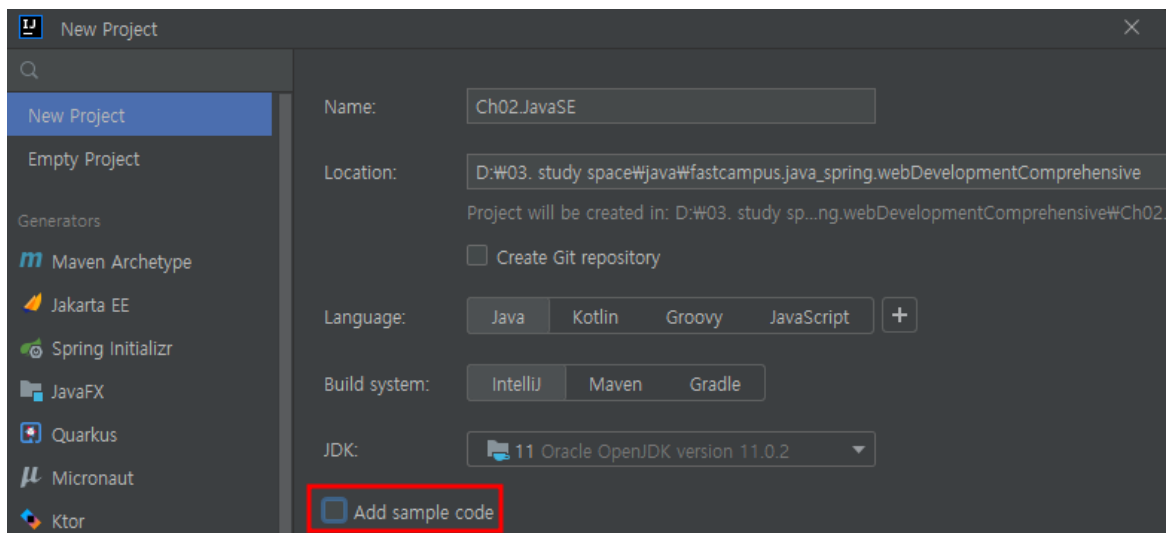
# Ch 02. Java SE Project

🕒 생성일	@2023년 12월 18일 오후 4:24
☰ 태그	

## ▼ Index

- [01. JavaSE Project 생성](#)
- [02. Program](#)
- [03. JavaSE Structure](#)
- [04. JVM & JAVA](#)

## ▼ 01. JavaSE Project 생성



## ▼ 02. Program

- 두 개의 정수를 더하여 출력하는 자바 프로그램을 만들어 보자.

```
public class Calculator {  
    public static void main(String[] args) {  
        int firNum, secNum, result;  
  
        firNum = 1;  
        secNum = 1;  
  
        result = firNum + secNum;  
    }  
}
```

```

        System.out.println(result);
    }
}

```

```

C:\Java\jdk-11.0.2\bin\java.exe "-javaagent:C:\Program F
2

Process finished with exit code 0

```

- 두 개의 정수를 사칙연산 처리하여 출력하는 프로그램

```

public class FourArithmetic {
    public static void main(String[] args) {
        int a = 12, b;
        b = 2;

        System.out.println("a + b = " + (a + b));
        System.out.println("a - b = " + (a - b));
        System.out.println("a * b = " + (a * b));
        System.out.println("a / b = " + (a / b));
    }
}

```

```

C:\Java\jdk-11.0.2\bin\java.exe "-javaagent:C:\Program F
a + b = 14
a - b = 10
a * b = 24
a / b = 6

Process finished with exit code 0

```

- 두 과목의 점수를 이용하여 총점과 평균을 출력하는 프로그램

```

public class Operator {
    public static void main(String[] args) {
        int kor = 77, eng;
        eng = 87;
    }
}

```

```

        System.out.println("총점 : " + (kor + eng));
        System.out.println("평균 : " + (kor + eng) / 2);
    }
}

```

C:\Java\jdk-11.0.2\bin\java.exe "-javaagent:C:\Program F

총점 : 164

평균 : 82

Process finished with exit code 0

## ▼ 03. JavaSE Structure

### ▼ Over View

실행가능한 파일은 .class로 만들어 진다(byte code).

소스파일은 .java로 저장한다.

JDK를 설치

컴파일러

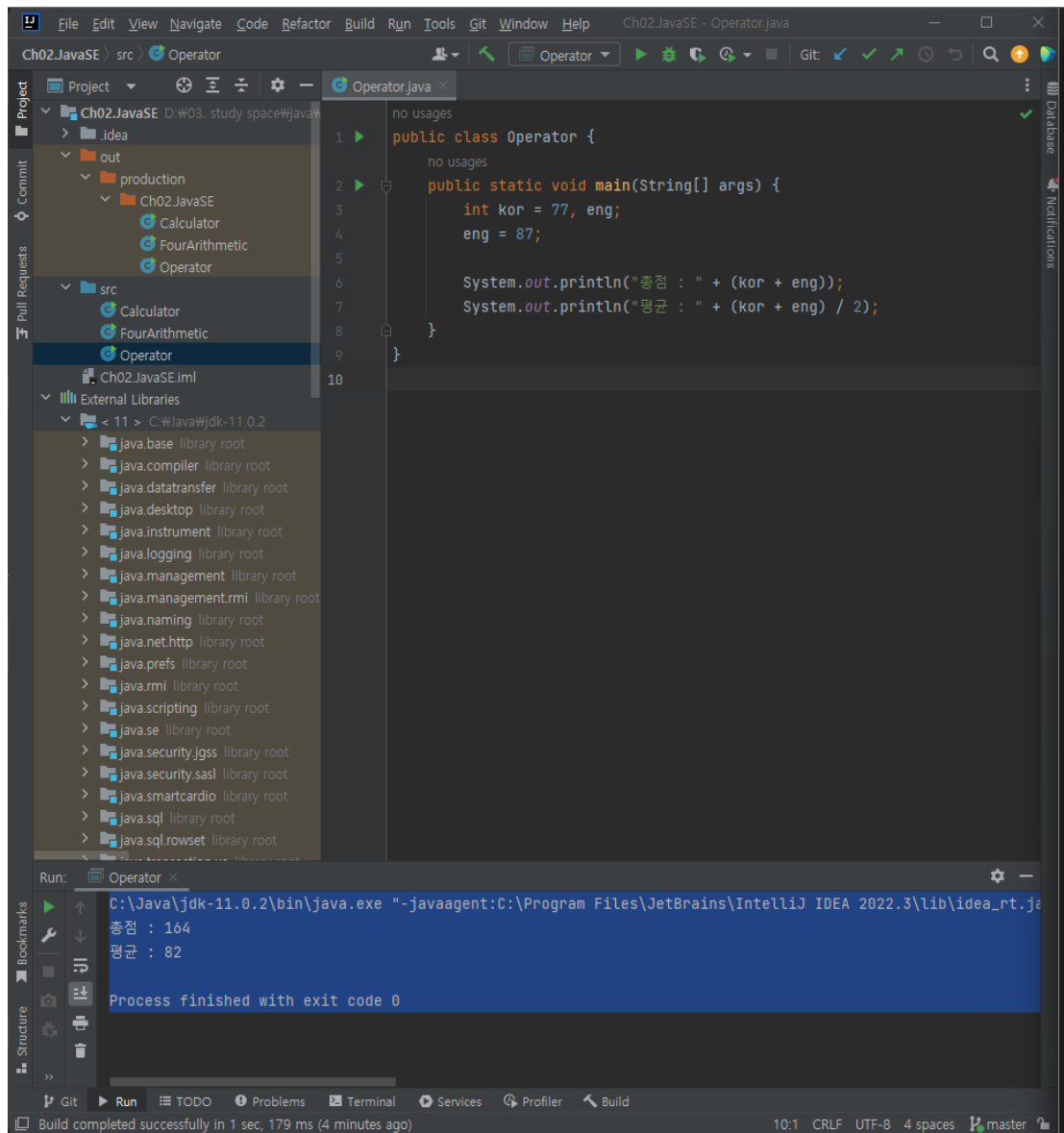
실행

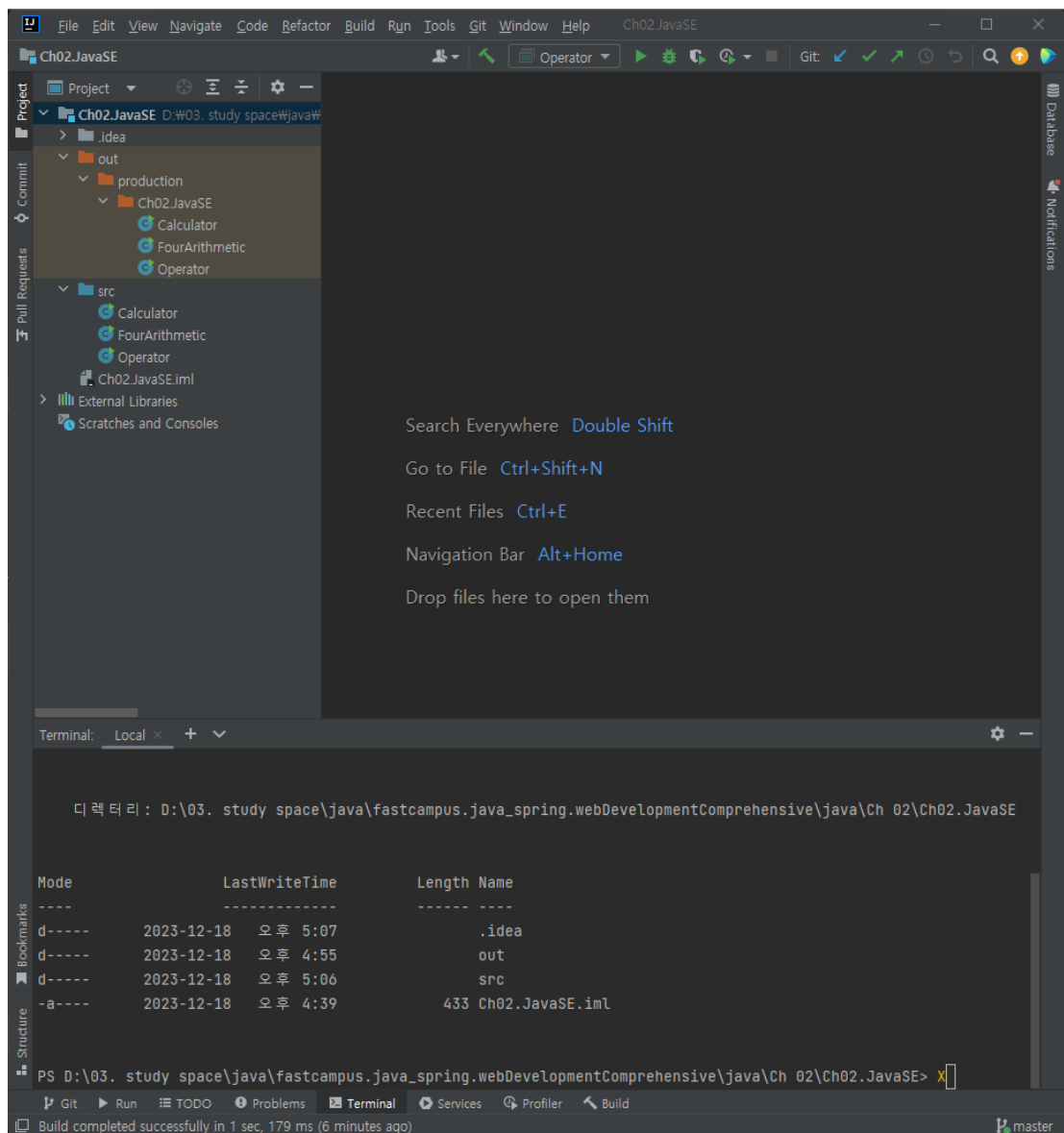
컴파일러

터미널 창을 이용해서 디렉토리를 보고 컴파일과 실행을 할 수 있다.

자바에서 제공하는 API(클래스들의 집합)를 사용하여 개발

Mode	LastWriteTime	Length	Name
d----			.idea
d----			out
d----			src
-a----		433	FastCampus.iml





## ▼ Terminal

```
PS D:\03. study space\java\fastcampus.java_spring.webDev
```

```
디렉터리: D:\03. study space\java\fastcampus.java_spr
```

```

Mode                LastWriteTime         Length Name
PS D:\03. study space\java\fastcampus.java_spring.webDev
.\src\Operator.java:7: error: unmappable character (0xEC
  
```

```

System.out.println("?뵐뵐? : " + (kor + eng) / 2);
                ^
.\src\Operator.java:7: error: unmappable character (0xA0)
System.out.println("?뵐뵐? : " + (kor + eng) / 2);
                ^

2 errors
PS D:\03. study space\java\fastcampus.java_spring.webDev
PS D:\03. study space\java\fastcampus.java_spring.webDev
총점 : 164
평균 : 82
PS D:\03. study space\java\fastcampus.java_spring.webDev

```

```

PS D:\03. study space\java\fastcampus.java_spring.webDevelopmentComprehensive\java\Ch 02\Ch02.JavaSE> dir

디렉터리: D:\03. study space\java\fastcampus.java_spring.webDevelopmentComprehensive\java\Ch 02\Ch02.JavaSE

Mode                LastWriteTime         Length Name
----                -
PS D:\03. study space\java\fastcampus.java_spring.webDevelopmentComprehensive\java\Ch 02\Ch02.JavaSE> javac.exe -d .\out\production\Ch02.JavaSE\ .\src\Operator.java
.\src\Operator.java:7: error: unmappable character (0xED) for encoding x-windows-949
    System.out.println("?뵐뵐? : " + (kor + eng) / 2);
                        ^
.\src\Operator.java:7: error: unmappable character (0xA0) for encoding x-windows-949
    System.out.println("?뵐뵐? : " + (kor + eng) / 2);
                        ^

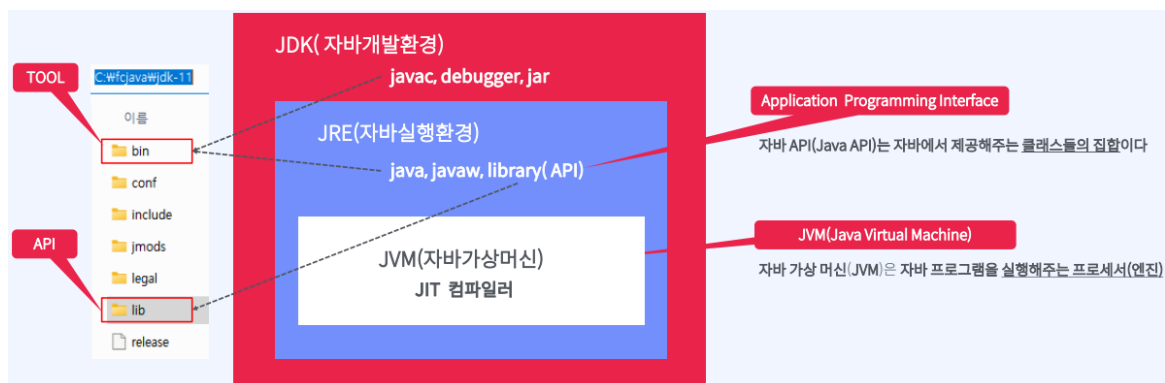
2 errors
PS D:\03. study space\java\fastcampus.java_spring.webDevelopmentComprehensive\java\Ch 02\Ch02.JavaSE> javac.exe -d .\out\production\Ch02.JavaSE\ .\src\Operator.java -encoding UTF-8
PS D:\03. study space\java\fastcampus.java_spring.webDevelopmentComprehensive\java\Ch 02\Ch02.JavaSE> java -classpath .\out\production\Ch02.JavaSE\ Operator
총점 : 164
평균 : 82

```

## ▼ 04. JVM & JAVA

### 01. JVM 이란

- Java 프로그램은 Platform(OS; Operation System) 독립적 개발 가능
- 'Managed Programming Language'로써 자동으로 메모리를 관리



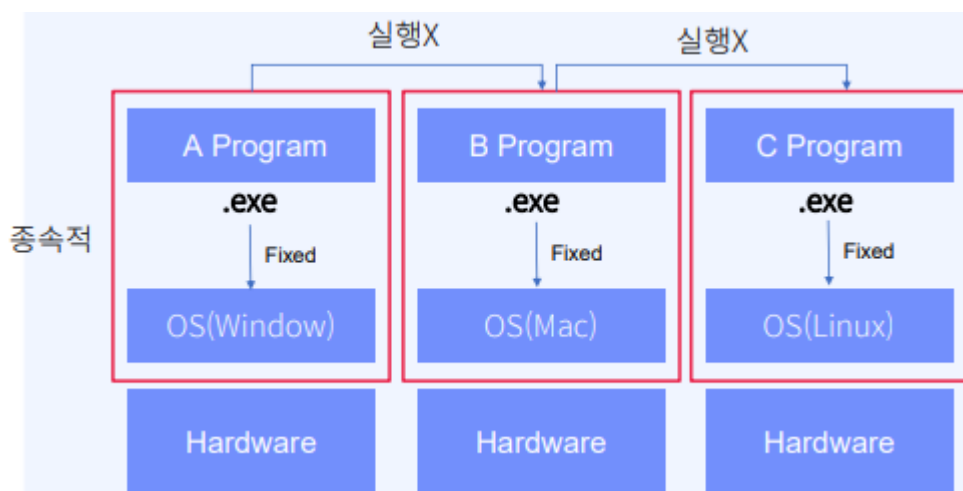


## ▼ Unmanaged Programming Language



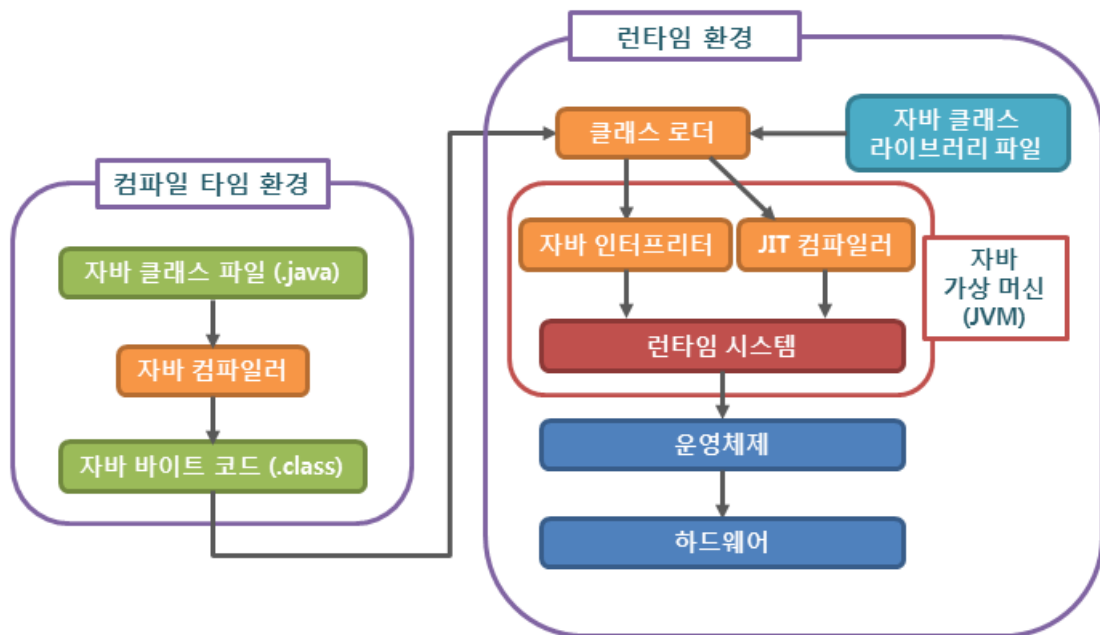
C++, MFC ...

- 메모리를 수동으로 관리해야 함으로 (메모리) 비 관리 프로그래밍 언어
- 각 OS에 맞춰서 코딩해야 함으로 각 OS API에 따라 코드가 다를 수 있으며,  
또한 각 OS에 맞춰 작성된 코드를 컴파일하여 OS에 종속된 실행 파일이 생성  
된다.  
⇒ 컴파일 언어



## 02. Java의 구동 방식

- 컴파일 과정



<https://junroot.github.io/programming/Java의-빌드와-배포/>

- Java Application 동작 방식

