객체지향 프로그래밍

Object-Oriented Programming

목차

소프트웨어의 이해

소프트웨어의 정의

- 프로그램: 프로그래밍한 원시 코드(source code)
- 소프트웨어
  - 프로그램(코드)을 비롯해 개발 과정에서 생성되는 모든 산출물과 각 단계에서 만들어지는 문서와 사용자 매뉴얼 등
    - (자료 구조, 데이터베이스 구조, 테스트 결과 등)

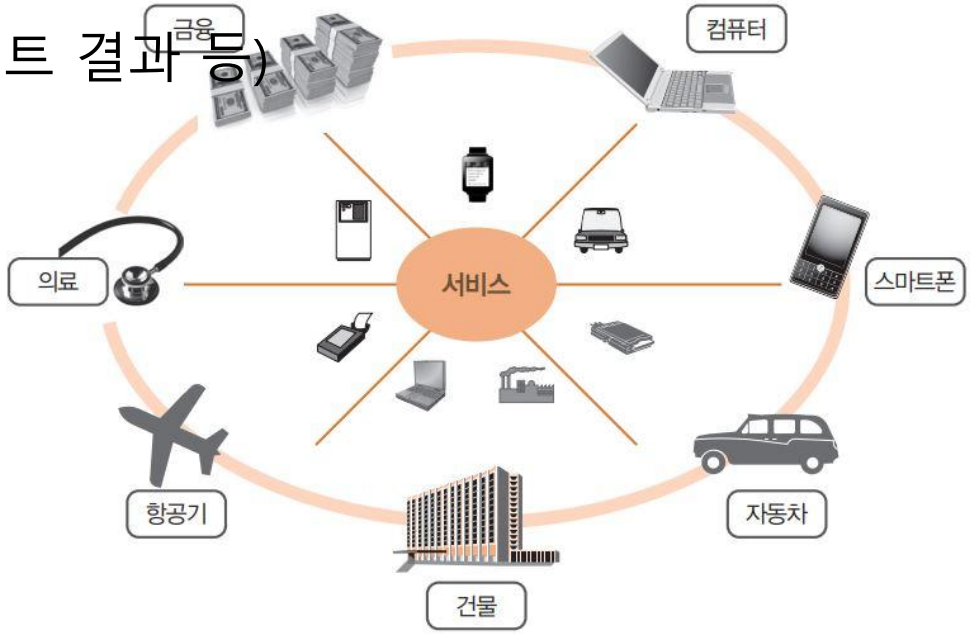


그림 1-1 소프트웨어가 사용되는 곳

소프트웨어의 이해

소프트웨어의 특징

- 제조가 아닌 개발
  - 제조: 개인 능력에 따라 차이가 크므로 능력에 따른 결과물의 차이는 크지 않음
  - 소프트웨어 개발 과정은 제조와 달리 개인 능력에 따라 차이가 큼

- 소모가 아닌 품질 저하
  - 하드웨어: 오래 사용하면 부품이 닳고 기능도 떨어짐
  - 소프트웨어: 닳지 않으며 또한 시간이 지나도 고장 빈도가 높지 않음, 사용 시작 단계부터 사용자의 요구가 계속 발생

소프트웨어의 이해

- 소프트웨어 개발 생명주기 (Software Development Life Cycle, **SDLC**)
  - 소프트웨어를 만들기 위해 계획 단계에서 유지보수 단계에 이르기까지 일어나는 일련의 과정
  - (계획, 분석, 설계, 구현, 테스트, 유지보수)

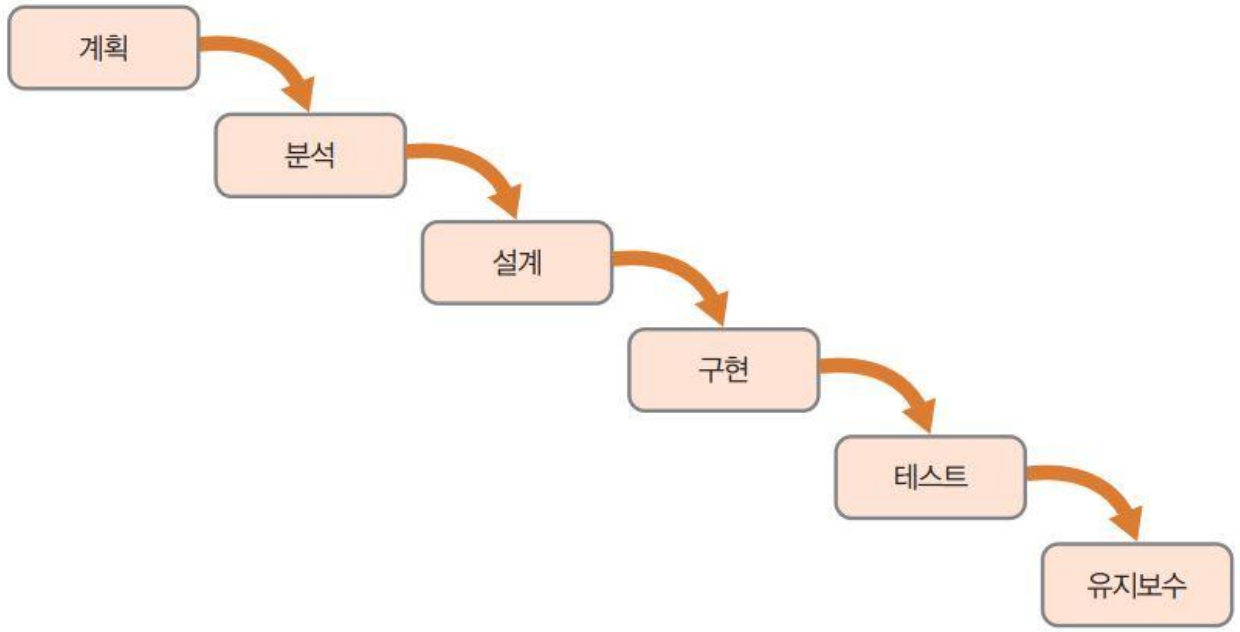


그림 1-8 소프트웨어 개발 생명주기

프로그램과 프로그래밍

프로그램(Computer program)

- 컴퓨터에서 실행될 때 특정 작업을 수행하는 일련의 명령어들의 모음
- 0과 1로 구성된 이진 형식의 파일로 저장 (기계어)

프로그래밍(Computer programming)

- 일련의 절차와 규칙을 **프로그래밍 언어**를 이용해 프로그램으로 구현하는 기술

프로그램과 프로그래밍

프로그래밍 언어(Programming language)

- 소프트웨어를 작성하기 위한 형식언어
- **저급(low-level) 언어**일 수록 기계어에 가깝고,
  고급(high-level) 언어일 수록 사람이 사용하는 언어에 가깝다.

```
INHEX   BSR     INCH        GET A CHAR
        CMP A   #'0         ZERO
        BMI     HEXERR      NOT HEX
        CMP A   #'9         NINE
        BLE     HEXRTS      GOOD HEX
        CMP A   #'A
        BMI     HEXERR      NOT HEX
        CMP A   #'F
        BGT     HEXERR
        SUB A   #7          FIX A-F
```

저급 언어(Assembly)

```c
int a = 10;
int b = 20;
int c;
 c = a + b;
 printf("%d", c);
```

고급 언어(C)

프로그램과 프로그래밍

객체 지향 프로그래밍 (Object-Oriented Programming, OOP)

- 컴퓨터 프로그램을 명령어의 목록으로 보는 시각에서 벗어나
  여러 개의 독립된 단위, 즉 '객체들의 모임'으로 파악하고자 하는 패러다임
- 추상화, 클래스, 상속 등의 고급 문법이 맞물려 기능

객체 지향 프로그래밍 언어

- Java
- C++
- Python
- Kotlin
- C#

프로그램과 프로그래밍

컴파일 (compile)

- Java, C 등 사람이 작성한 언어를 컴퓨터가 실행할 수 있는 기계어로 변환하는 과정
- **컴파일러(compiler)**라는 소프트웨어가 이 과정을 수행

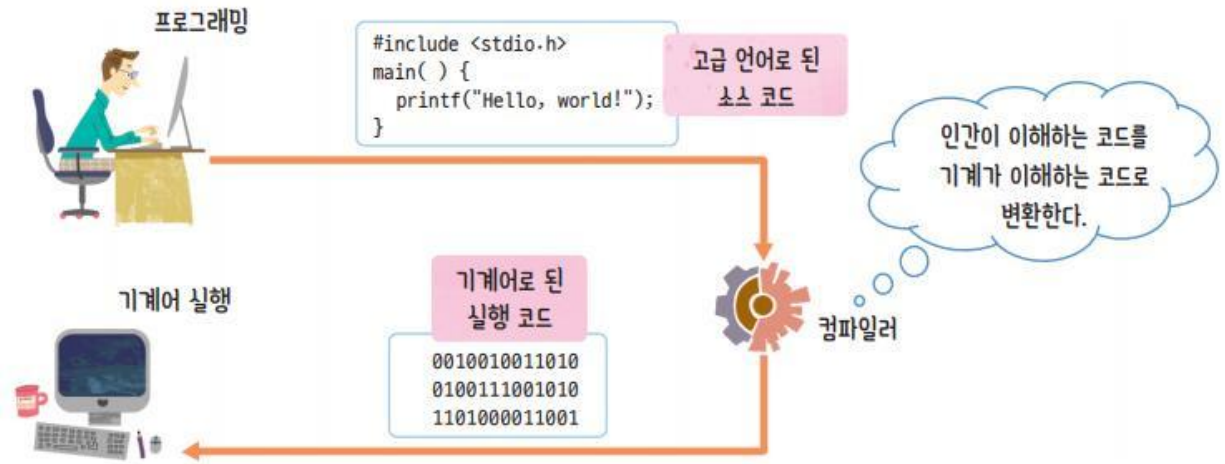
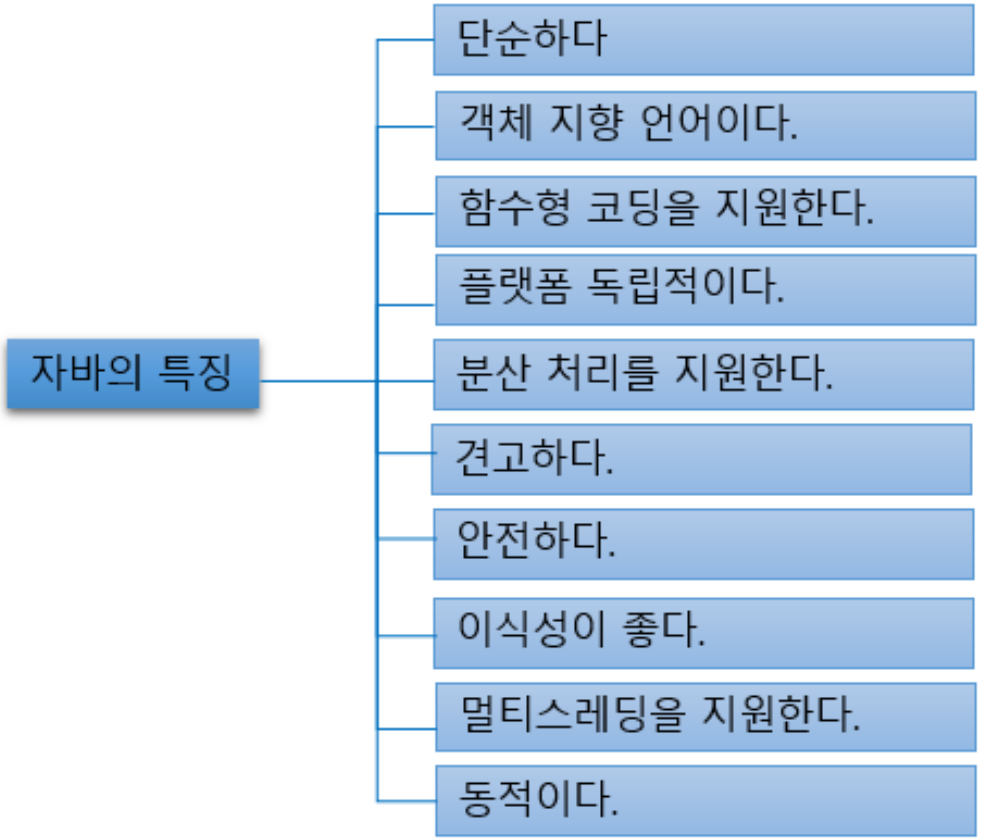


그림 1-1 고급 프로그래밍 언어의 실행 과정

Java 소개

Java: Sun Microsystems에 의해 개발된 **객체 지향 프로그래밍 언어**



Java 소개

Java 버전

- 1996년 Java 1.0 발표,
  2025년 9월 현재 최신 버전은 Java 24

- 어떤 버전 환경에서든, 하위 버전에서 작성된 프로그램을 실행할 수 있도록 하는 **하위 호환성(backward compatibility)**가 보장됨

Java 소개

활용 분야

- 모바일 어플리케이션
- 웹 어플리케이션 서버(WAS)
- CRM, ERP, SCM 등 기업용 애플리케이션
- 빅데이터, 클라우드, 소셜, 모바일, 사물인터넷 등 혁신 기술

Java 소개

Java 가상 머신(Java Virtual Machine, JVM)

- **바이트 코드**로 컴파일된 프로그램을 실행할 수 있도록 하는 가상 머신
- 기존 프로그래밍 언어의 **플랫폼 종속성 문제를 해결**
- 자바의 가장 큰 장점은 **JVM(자바 가상 머신)** 위에서 실행된다는 점
  - 하드웨어나 운영체제가 달라도, JVM만 있으면 같은 코드가 실행 가능
- 모바일 어플리케이션
  - 다양한 제조사와 운영체제가 공존하던 **초기 모바일 환경**(예: 노키아, 모토로라, 삼성, LG 등)에서 특히 강점

Java 가상 머신

기존 프로그래밍 언어의 플랫폼 종속성

- **하드웨어에 따른 종속성**
- 하드웨어 아키텍처(CPU)에 따라 사용하는 기계어 종류가 다름

- **운영체제에 따른 종속성**
- 운영체제마다 메모리 관리 방식이 다름
- 실행 파일 형식이 다름
- 응용 프로그래밍 인터페이스(API) 형식이 다름

→ 하드웨어, 운영체제 종류 별로 프로그램을 따로 작성

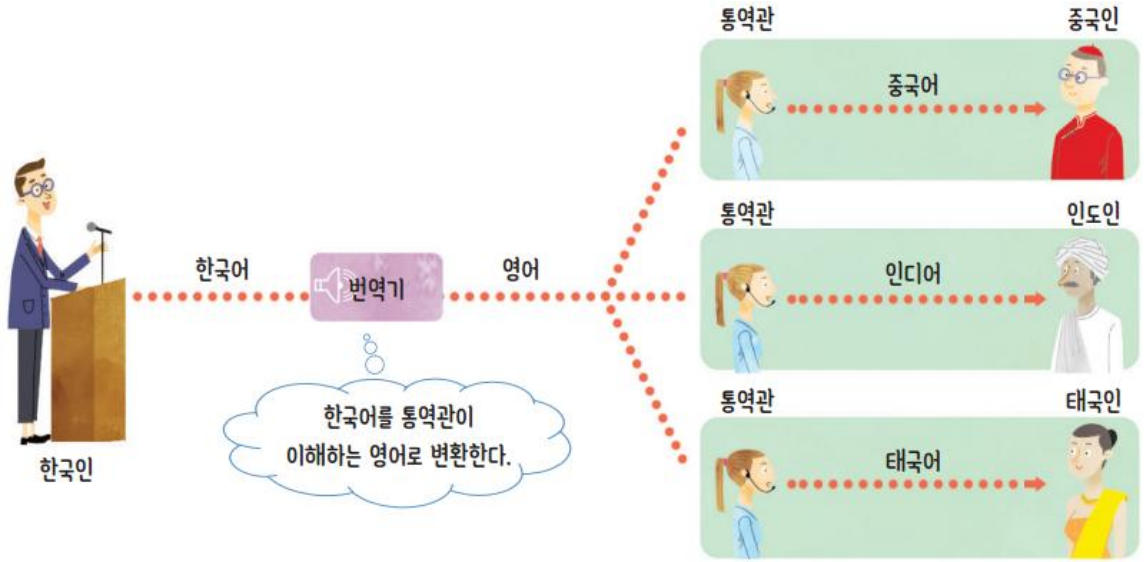
Java 가상 머신

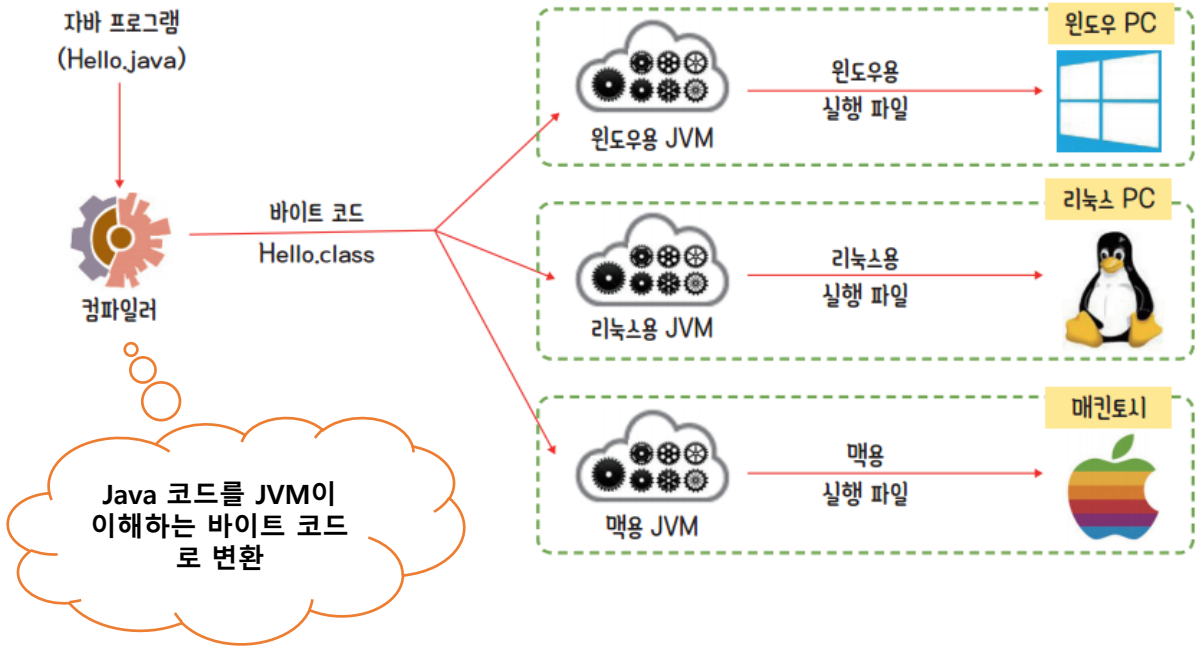
Java 바이트 코드(Java byte code)

- Java 가상 머신의 명령 집합
- Java 소스 코드 컴파일 시 결과물
- 다양한 플랫폼에서 일관되게 실행될 수 있도록 설계됨

Java 가상 머신

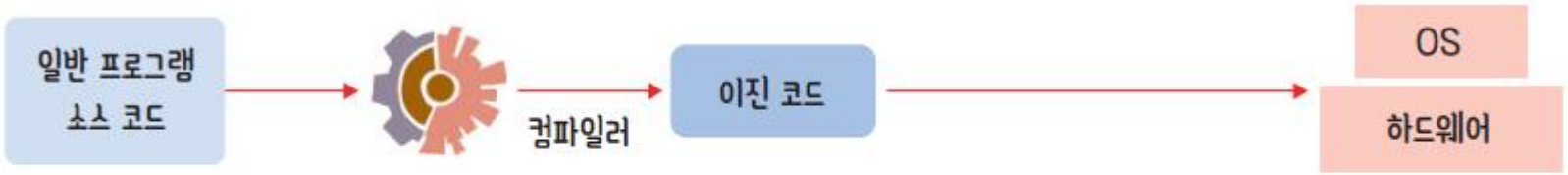
Java 가상 머신은 통역관 역할





Java 프로그램은 한번 컴파일되면 어디서나 실행 가능

Java 가상 머신



플랫폼 종속

플랫폼 독립

Java 개발 환경 구축

Java 개발 환경

자바 개발 키트(JDK)

- 컴파일러
- 디버거
- JVM

통합 개발 환경(IDE)

여러 개 파일로 구성된

프로젝트를 효율적으로 관리

Java 개발 환경 구축

JDK와 JRE

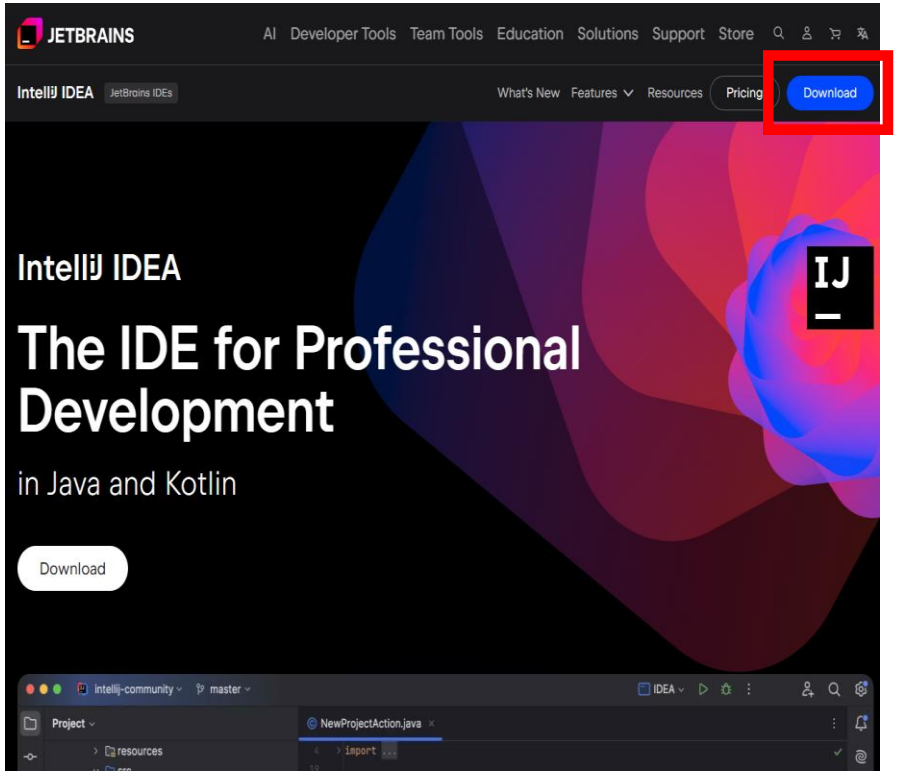
- **JDK(Java Development Kit)**
- Java 프로그램 작성을 위한 컴파일러, 디버거

- **JRE(Java Runtime Environment)**
- Java 프로그램(바이트 코드)를 실행하기 위한 JVM과 API 등

Java 개발 환경 구축

IntelliJ IDEA 설치

1. 설치 페이지 접속
    - IntelliJ 공식 홈페이지 접속 (https://www.jetbrains.com/idea/)
    - **Download** 버튼 클릭

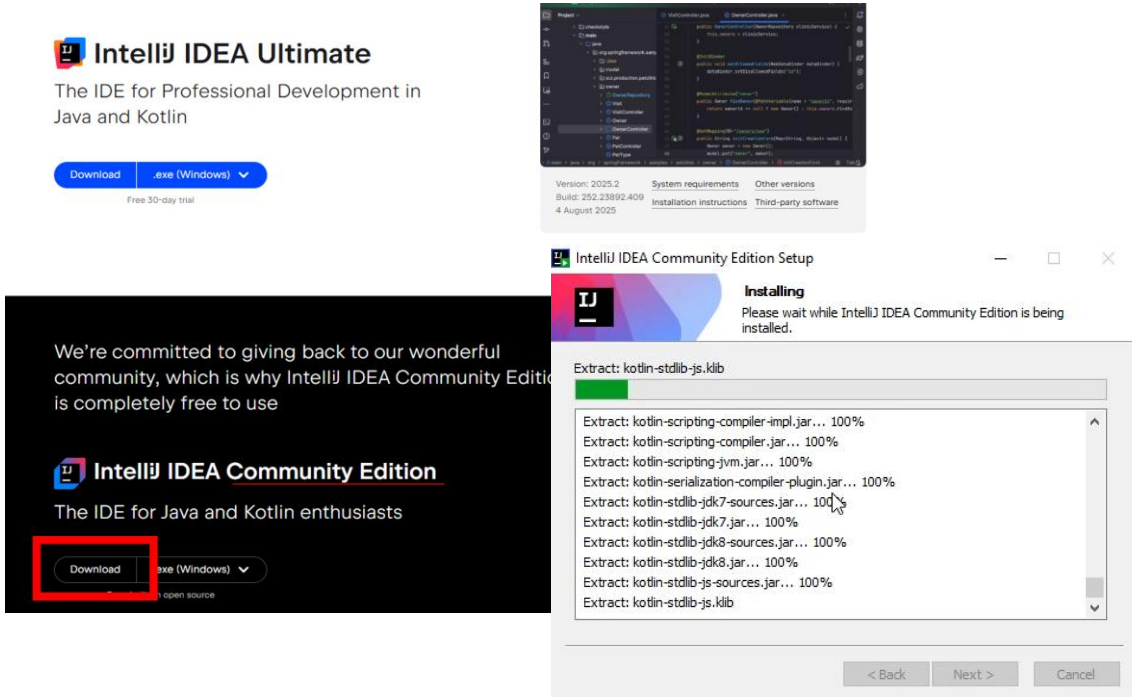


Java 개발 환경 구축

IntelliJ IDEA 설치

2. 설치 프로그램 다운로드 및 설치 진행

- **Community Edition** 설치 파일 다운로드
- 설치 파일 실행
- 기본 옵션으로 프로그램 설치

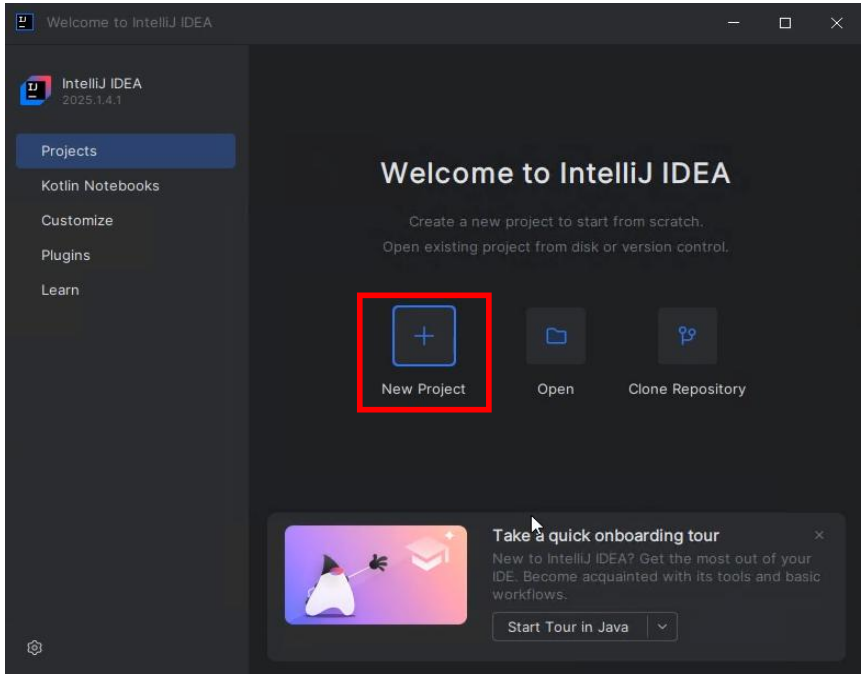


Java 개발 환경 구축

JDK 설치

1. IntelliJ New Project 창 열기

- IntelliJ 실행
- New Project 클릭

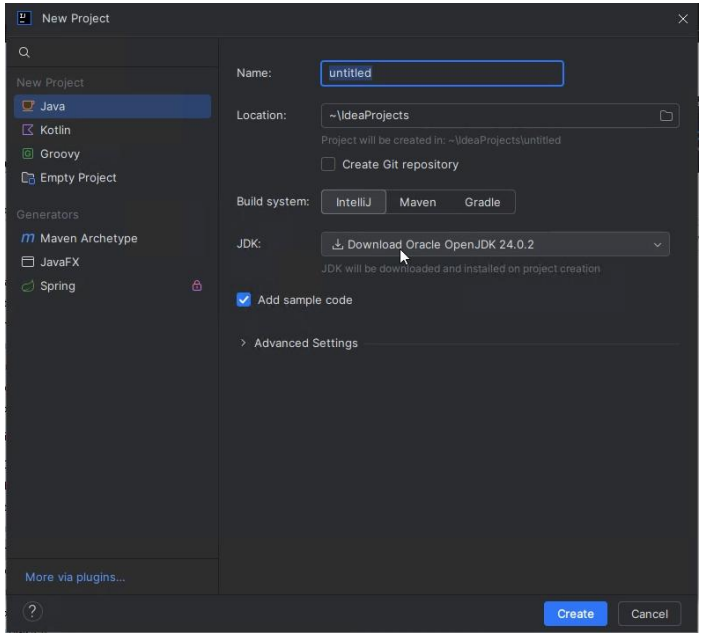


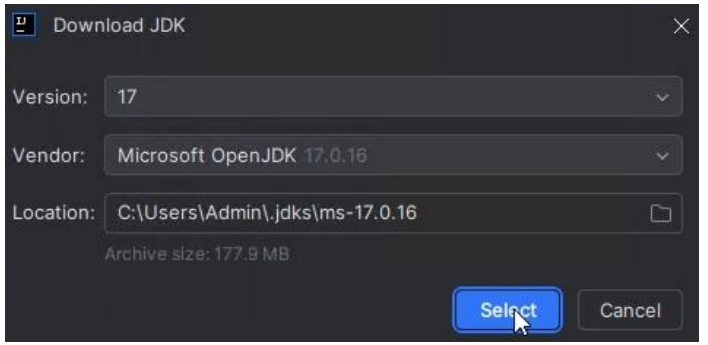
Java 개발 환경 구축

JDK 설치

2. JDK 옵션 설정

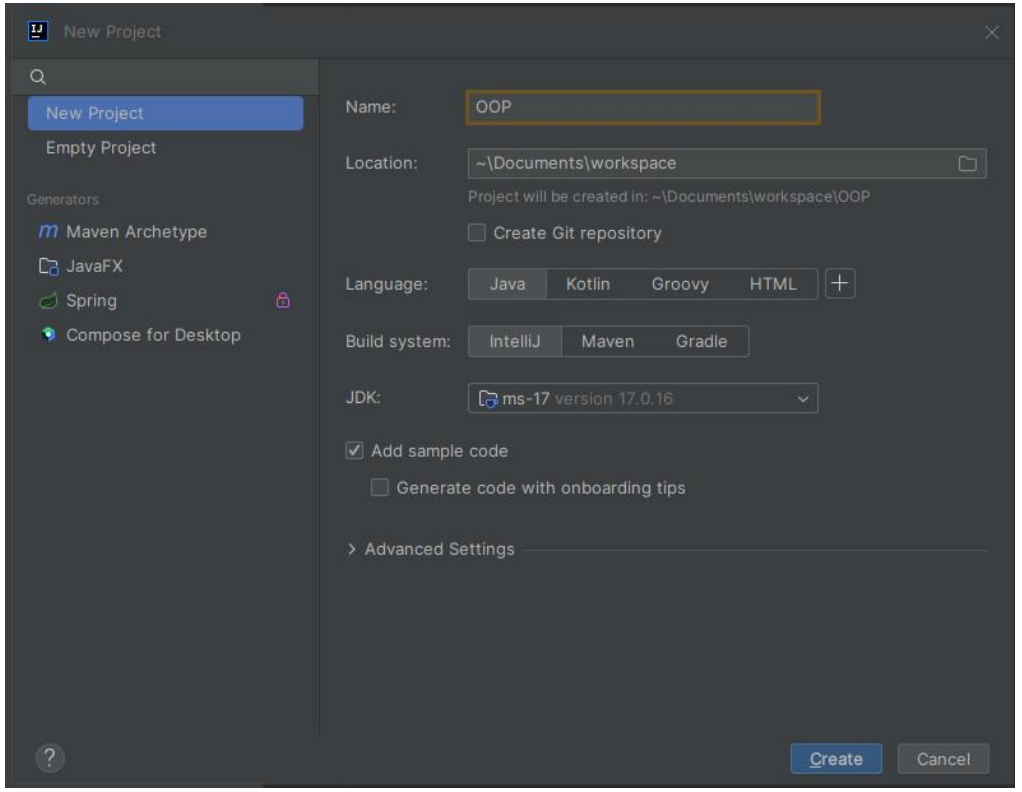
- JDK – Donwload JDK… 선택
- Version: 17 설정
- Select 버튼 클릭





Hello 프로그램 작성과 실행

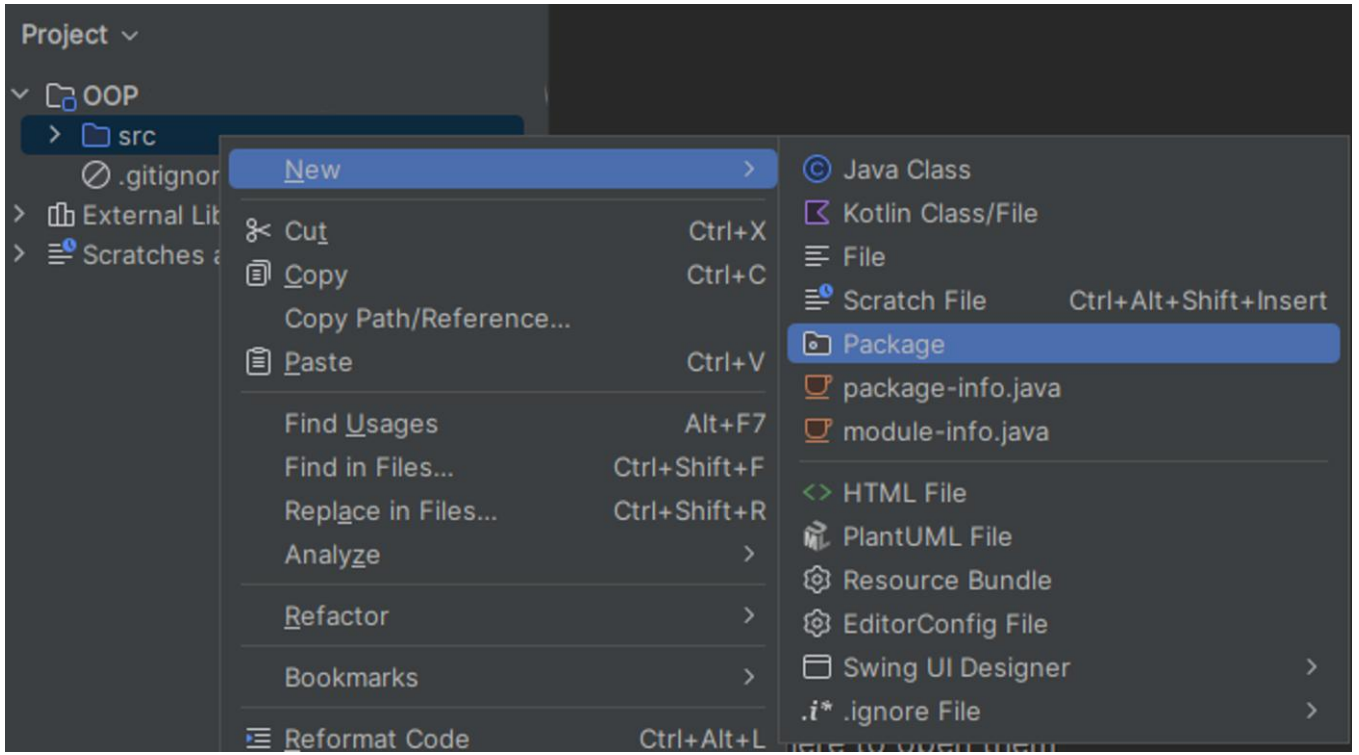
1. 새 프로젝트 생성
   - Intellij – New Project 창 열기
   - Project name: OOP
   - JDK: 17 version



Hello 프로그램 작성과 실행

2. 패키지(폴더) 생성

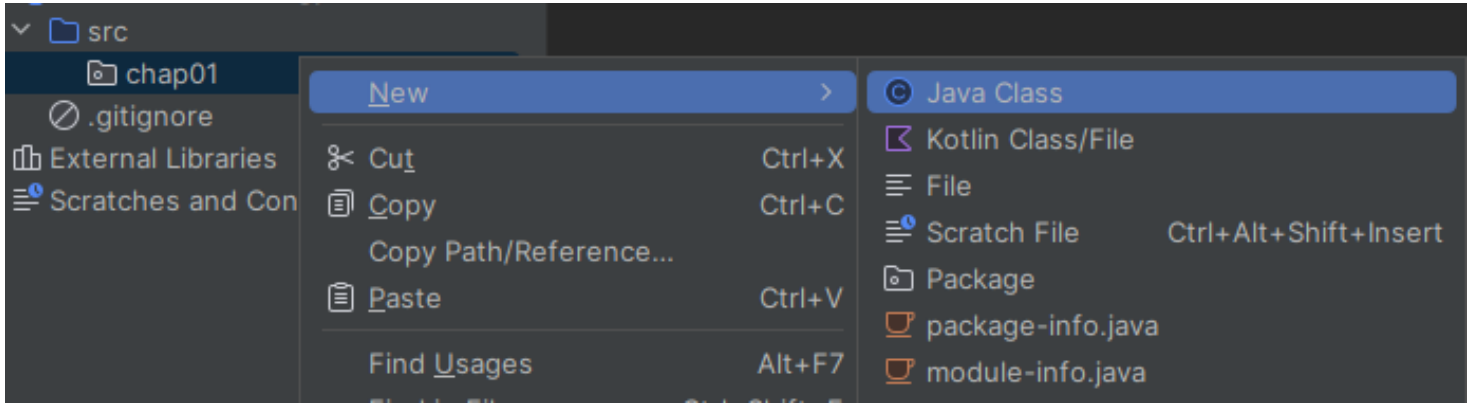
- src 폴더 우클릭
- New > package
- chap01 입력



Hello 프로그램 작성과 실행

3. 클래스 생성

- chap01 패키지 우클릭
- New > Java Class
- Hello 입력



Hello 프로그램 작성과 실행

4. Hello 클래스 작성

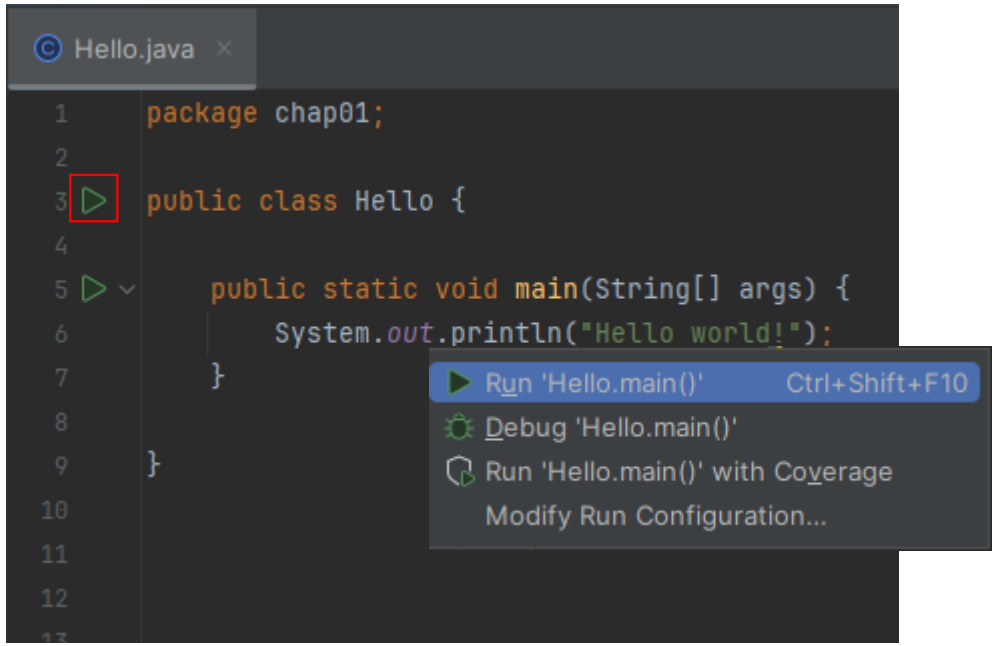
```java
package chap01;

public class Hello {

    public static void main(String[] args) {
        System.out.println("Hello world!");
    }

}
```

Hello 프로그램 작성과 실행



단축키 활용

- Hello 클래스 파일 편집 창이 열린 상태로 Shift + F10 키 입력

컴퓨터 시스템- 소프트웨어와 하드웨어와의 관계