Eclipse를 이용한 기본 소스코드 작성

Project 생성

Project

Eclipse를 이용한 코드 작성

1. **Project 생성**
   - 하나의 큰 작업(하나의 작품)으로 여러 개의 패키지가 존재하는 공간

2. **Package 생성**
   - 프로젝트 안에 존재하는 하나의 폴더(Class들이 들어있음)

3. **Class 생성**
   - 실제 소스코드가 작성되어 있는 메소드들이 모여 있는 파일

4. **Method 생성**
   - 하나의 기능을 담당하는 공간으로 실제 소스코드가 작성되어 있음

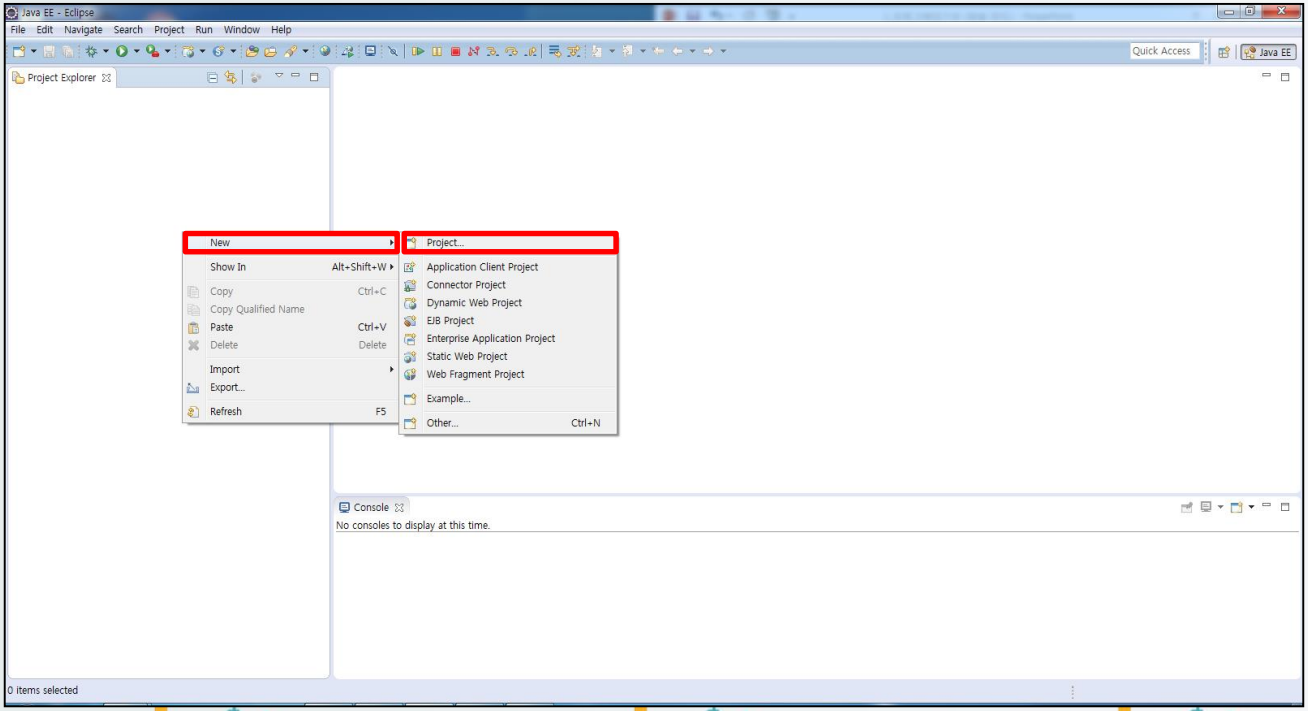
5. **Method 내부에 소스코드 작성**

6. **컴파일 및 실행 (단축키 : Ctrl + F11)**

Project

Project 생성

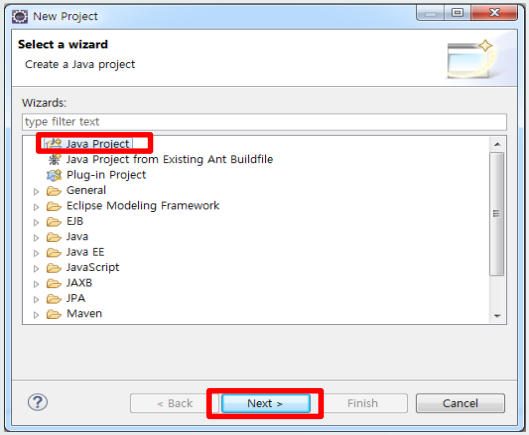
마우스 오른쪽 클릭 → New → Project

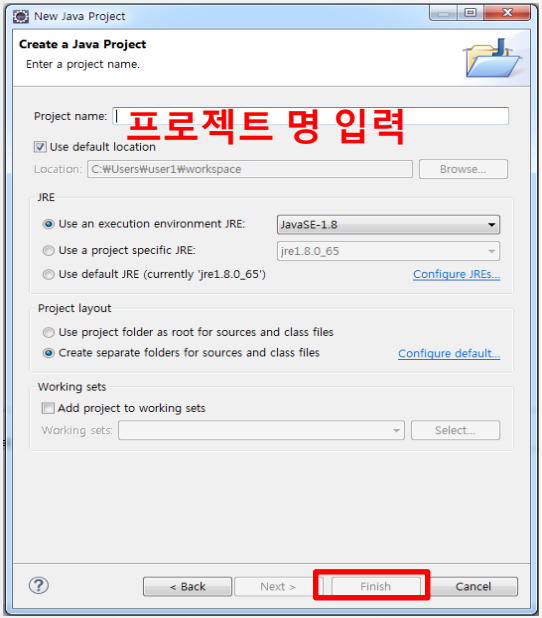


Project

Project 생성

Java Project 클릭 → 프로젝트명 입력 → Finish

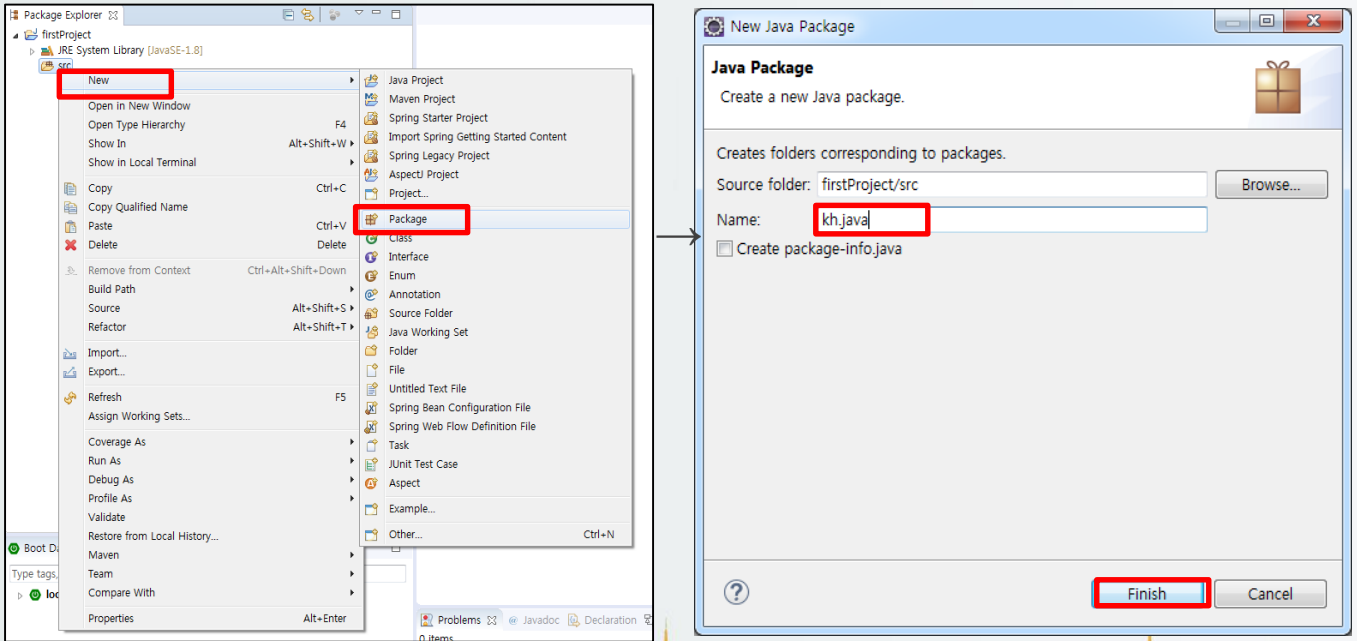




Project

Package 생성

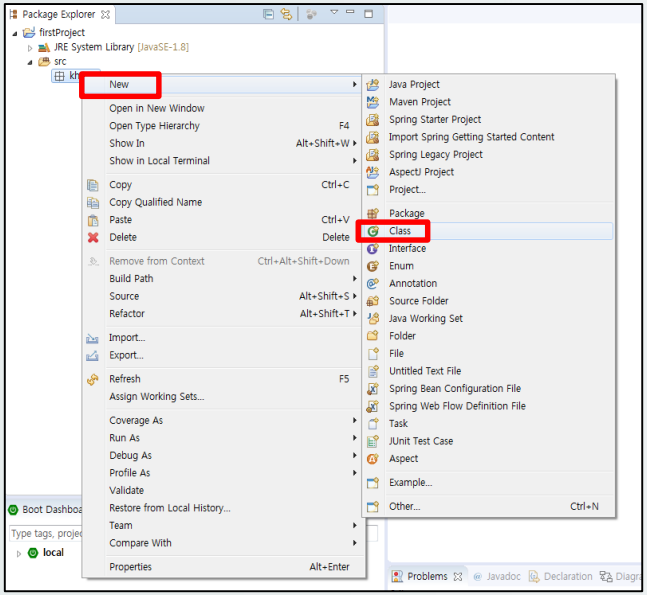
src 폴더 우 클릭 → New → Package → 패키지명 입력 → Finish

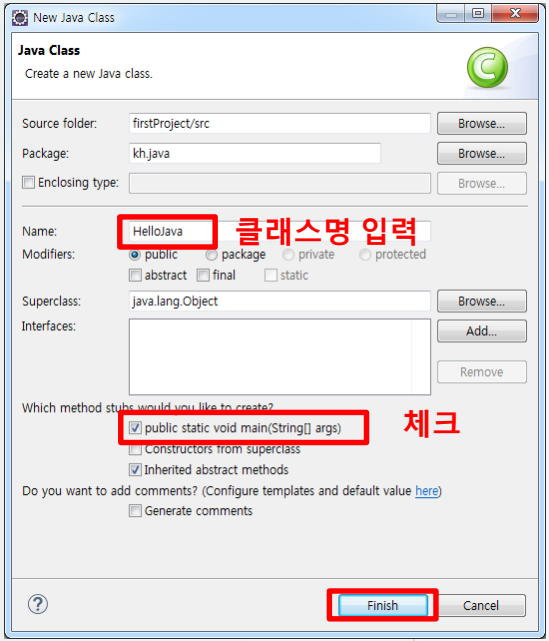


Project

Class 생성

Package 우 클릭 → New → Class → Class명 입력 → public static void main 체크 → Finish

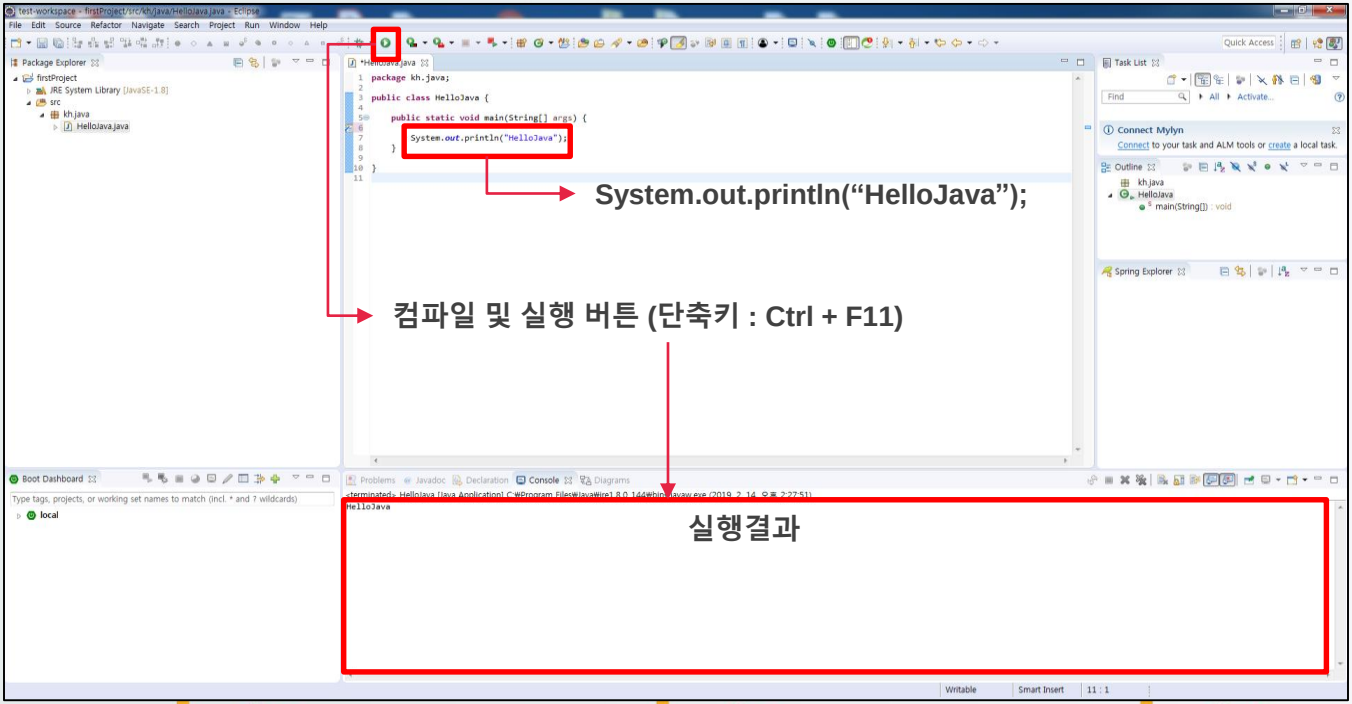




Project

소스코드 작성 후 컴파일 및 실행

Main Method 내부에 출력코드 작성



작성규칙

Project

소스코드 구조

- 소스코드는 메소드의 블록 안에 작성(블록은 중괄호 { } 내부를 의미)
- 클래스 안에 작성 된 public static void main(String[] args)는 메인 메소드 또는 실행 메소드라고 부르며, 하나의 프로젝트 당 한 개만 만들어서 사용
  → 클래스 생성시 체크한 체크박스가 메인 메소드를 자동으로 만들어주는 것을 체크한 것
  → 추가 클래스를 생성하는 경우에는 체크박스를 체크하지 않고 생성
- System.out.println("HelloJava");
  → "" 쌍 따옴표 내부의 내용을 출력하는 명령어

HelloJava.java

```
1  package kh.java;
2
3  public class HelloJava {
4
5      public static void main(String[] args) {
6
7          System.out.println("HelloJava");
8      }
9
10 }
11
12
13
14
15
```

패키지 이름

클래스 이름

메소드 이름

소스코드

Project

표기법

1. 카멜 표기법 : 낙타의 혹이 튀어 나온 것처럼 소문자로 시작하여 단어와 단어가 만날 때 뒷 단어의 첫 글자를 대문자로 작성하는 방식 → phoneNumber

2. 파스칼 표기법 : 카멜표기법과 유사하지만 첫 글자까지 대문자로 작성 → PhoneNumber

- 프로젝트 이름, 메소드 이름은 카멜 표기법 사용
  → 프로젝트 이름 : firstProject / 메인메소드 이름 : main

- 패키지 명은 모두 소문자로 작성하고 단어와 단어 사이를 '.' 으로 구분(2~3단계 권고)
  → 패키지 이름 : kh.java

- 클래스 이름은 파스칼 표기법 사용
  → 클래스 이름 : HelloJava

※ 표기법은 실제로 지키지 않더라도 사용이 가능하다. 하지만 표기법은 개발자들 간의 암묵적인 규약(컨벤션)이며, 이러한 규약을 통해서 새로운 코드를 접하거나 오류가 발생한 경우 분석 및 파악이 용이하다. 처음 시작할 때 약간 불편하더라도 습관을 들일 수 있도록 하는 것이 좋다.

Project

들여쓰기

- 새 블록마다 들여쓰기를 하며, 작성중인 코드가 어느 블록에 속해 있는지 구분하기 위해 사용

1. GNU
   - 블록을 아래에서 들여쓰기 해서 작성
   - 블록의 표시가 분명하여 구조가 잘 보임
   - 들여쓰기를 많이 해서 수평으로 많은 코드 작성 불가

```
if(true)
        {
                System.out.println(“HelloJava”);
        }
```

2. K&R
   - 여는 블록을 같은 행에 배치
   - 코드 줄 수를 절약하여 한눈에 많은 코드를 볼 수 있고 수평으로 많은 코드 작성 가능
   - 일반적으로 가장 많이 사용됨

```
if(true){
        System.out.println(“HelloJava”);
}
```

3. BSD
   - GNU의 블록표시가 분명한 장점과, K&R의 수평으로 많은 코드를 작성가능한 장점을 가져와 결합한 스타일

```
if(true)
{
        System.out.println(“HelloJava”);
}
```

Project

주석

- 컴파일 시 소스코드를 분석하지 않는 부분
- 소스코드의 내용을 설명하기 위한 문구를 넣을 때 사용

1. 한 줄 주석(//)
   - //기준으로 오른쪽을 컴파일 하지 않음
   - System.out.println(“안녕하세요”); → 안녕하세요 메시지 출력
   - //System.out.println(“안녕하세요”); → 출력 명령어가 주석 오른쪽에 있으므로 출력되지 않음

2. 범위 주석(/* */)
   - /* : 주석시작 */ : 주석 끝
   - 주석 시작과 끝 사이가 모두 주석처리 됨
   - /*
     System.out.println(“안녕하세요”);
     System.out.println(“이건”);
     System.out.println(“범위주석입니다!”);
     */

   주석 시작과 끝사이의 3줄 모두 주석 처리되어 출력되지 않음

Project

실행클래스와 기능제공 클래스

1. 실행클래스
   - 메인 메소드(실행 메소드)를 가지고 있는 클래스
   - 메인 메소드가 프로젝트당 1개이기 때문에 실행클래스도 프로젝트당 1개 존재
   - 실행클래스의 메인 메소드에서는 기능제공 클래스에 작성한 기능을 실행하는 용도로만 사용하며, 기능을 구현하지 않는다.

2. 기능제공 클래스
   - 실제 기능이 구현되어 있는 메소드를 가지고있는 클래스
   - 기능용 메소드는 제한이 없기 때문에 여러 개의 클래스 및 메소드를 작성해도 상관없음
   - 일반적으로 비슷한 기능들을 하나의 기능제공 클래스로 작성
   - 유사한 기능제공 클래스들은 같은 패키지 내부 존재

※ Hello라는 메시지를 출력하는 프로그램을 작성하는 경우 기능제공 클래스에 Hello메세지를 출력하는 메소드를 작성하고, 실행클래스의 메인 메소드에서 기능제공클래스의 작성된 메소드를 호출하여 사용

Project

기능제공 클래스

- 기능제공용 클래스를 생성하고 Hello메세지를 출력하는 메소드 및 소스코드 작성

```
1  package kh.java.func;
2
3  public class PrintMsg {
4
5      public void printHelloMsg() {
6
7          System.out.println("Hello!!!!!");
8
9      }
10 }
```

패키지 명

클래스 명

메소드 명

소스코드

메소드 블록

클래스 블록

Project

실행 클래스

- Hello 메시지를 출력하는 메소드를 실행하는 클래스
- 다른 패키지에 있는 클래스를 실행하기 위해서는 import를 통해서 해당 클래스의 위치를 명시

```
1  package kh.java.run;
2
3  import kh.java.func.PrintMsg;
4
5  public class Start {
6
7      public static void main(String[] args) {
8          PrintMsg pm = new PrintMsg();
9          pm.printHelloMsg();
10
11     }
12
13 }
```

기능제공 클래스의 위치 명시
(패키지경로 포함)

- 다름 클래스의 메소드를 호출하여 실행하는 방법
  클래스이름 약어 = new 클래스이름();
  약어.메소드이름();

실습

Project

실습문제

문제 1

- 프로젝트 명 : printInfomationPjt

- 실행용 클래스
  1. 패키지명 : kr.or.iei.start
  2. 클래스명 : Run
  3. 내용 :
  Information 클래스의 printInfo메소드 실행

- 기능제공 클래스
  1. 패키지명 : kr.or.iei.vo
  2. 클래스명 : Information
  3. 내용 :
  자기소개 출력(본인의 이름, 나이, 생년월일)

문제 2

- 프로젝트 명 : shoppingPjt

- 실행용 클래스
  1. 패키지명 : com.khmarket.run
  2. 클래스명 : Start
  3. 내용 :
  Goods 클래스의 printList메소드 실행

- 기능제공 클래스
  1. 패키지명 : com.khmarket.vo
  2. 클래스명 : Goods
  3. 내용 : 쇼핑몰 판매물건 출력
  (예 : 맥북프로 2020-16인치 / 300만원)