

## 제 2강

# 출력메서드와 입력메서드

교재:내용 없음  
PPT 수업

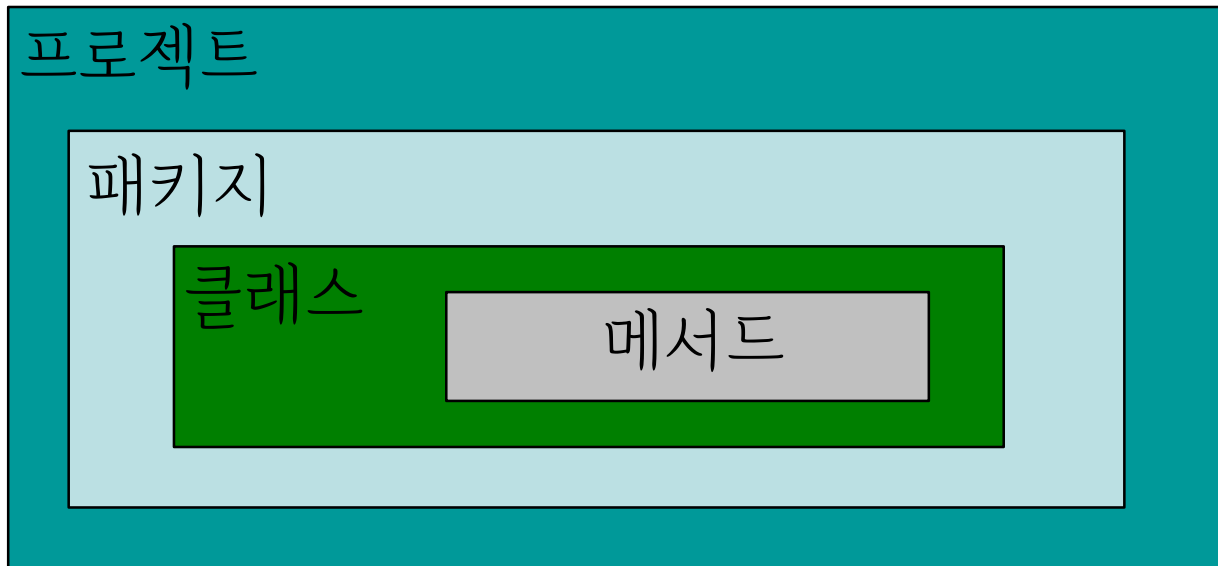
# 목차

1. 프로젝트 만들기
2. 주석
3. 출력메서드
4. 입력메서드

# 1. 프로젝트 만들기

JAVA 프로젝트 생성 순서?

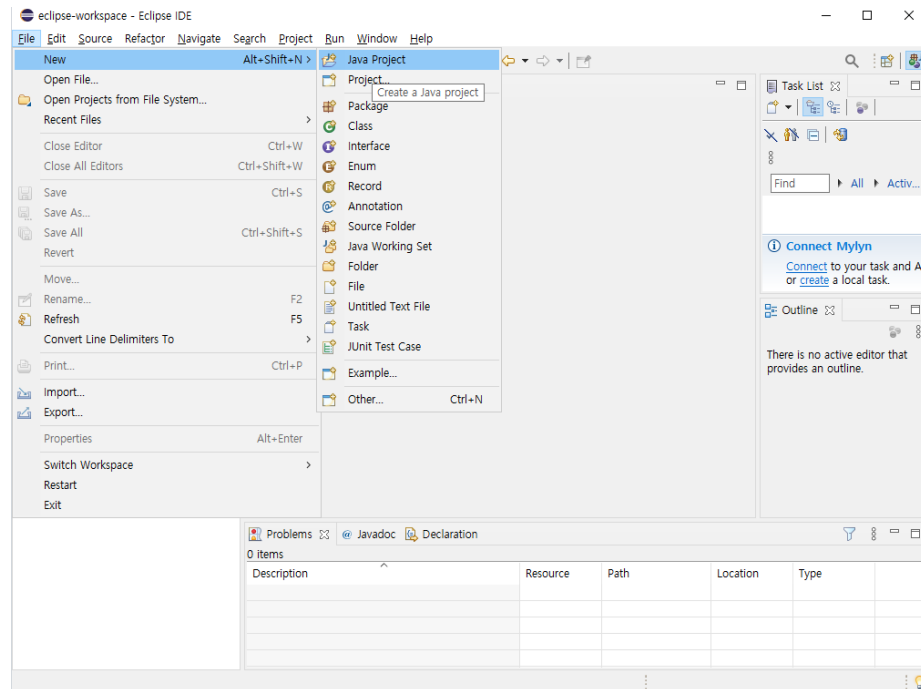
: 프로젝트-패키지-클래스-메서드



# 1. 프로젝트 만들기

JAVA 프로젝트 생성 순서?

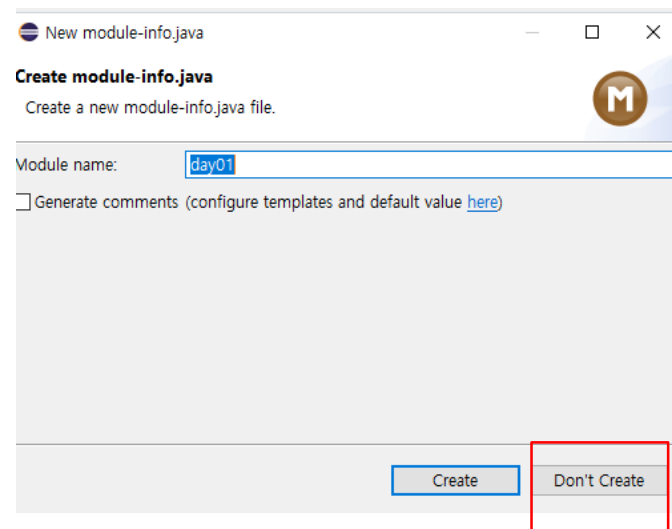
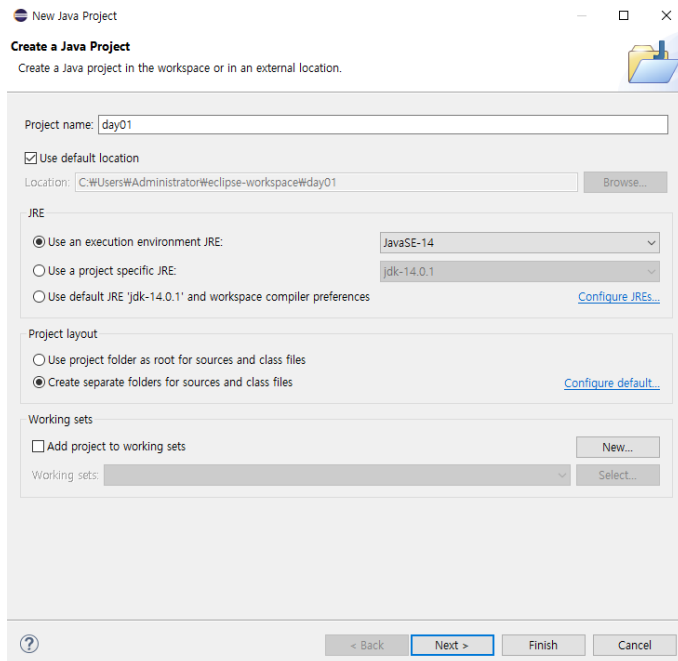
File -> New -> Java Project 클릭



# 1. 프로젝트 만들기

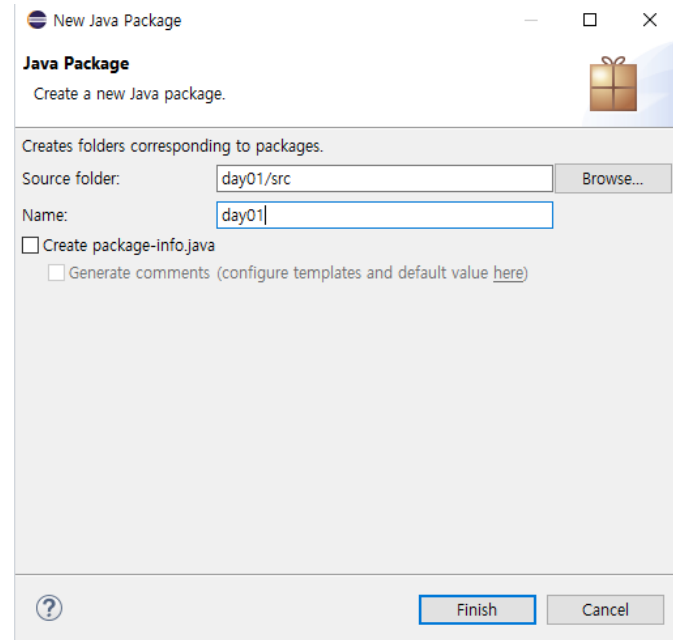
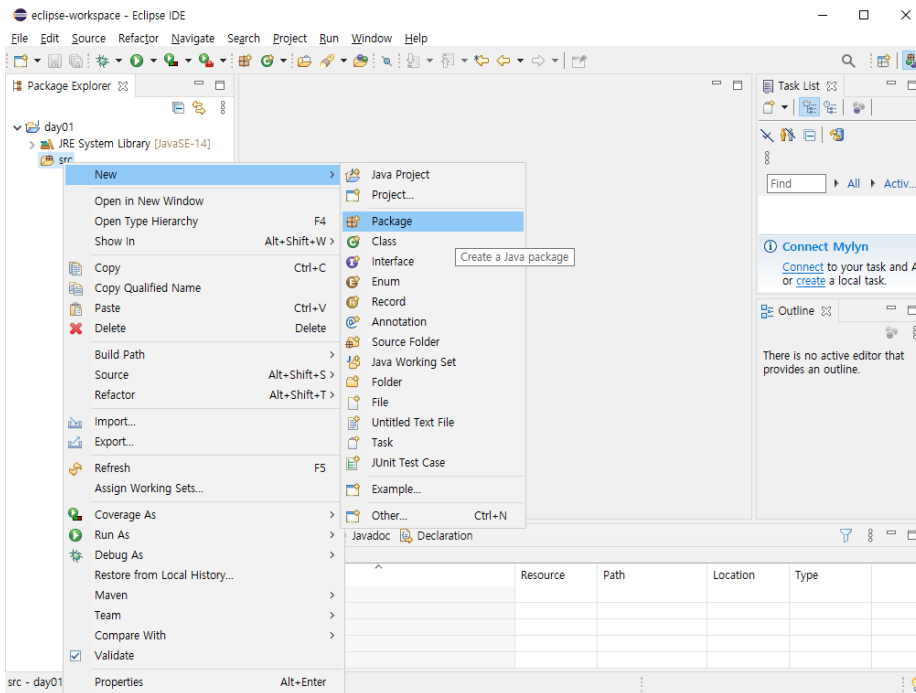
JAVA 프로젝트 생성 순서?

프로젝트명은 day01, 모듈은 생성하지 않고 확인



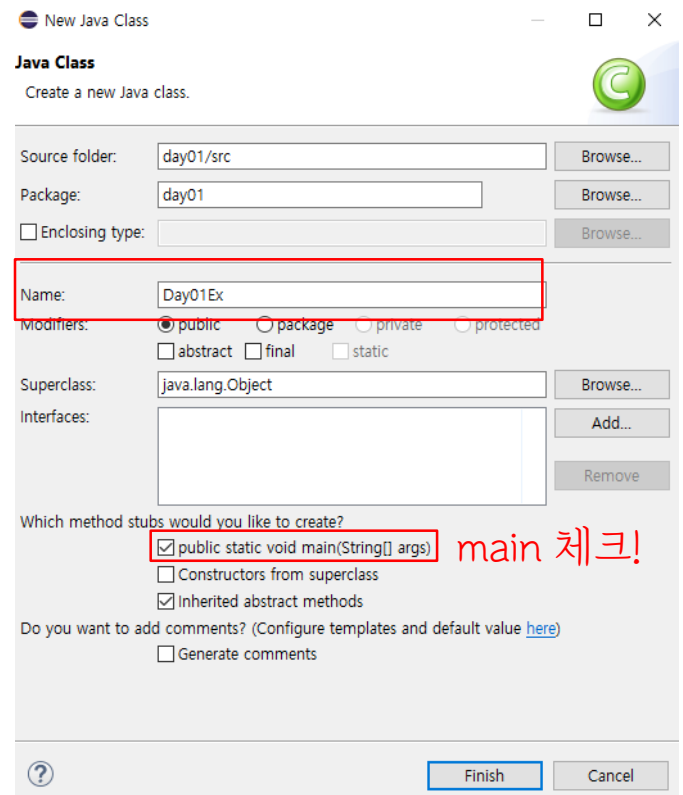
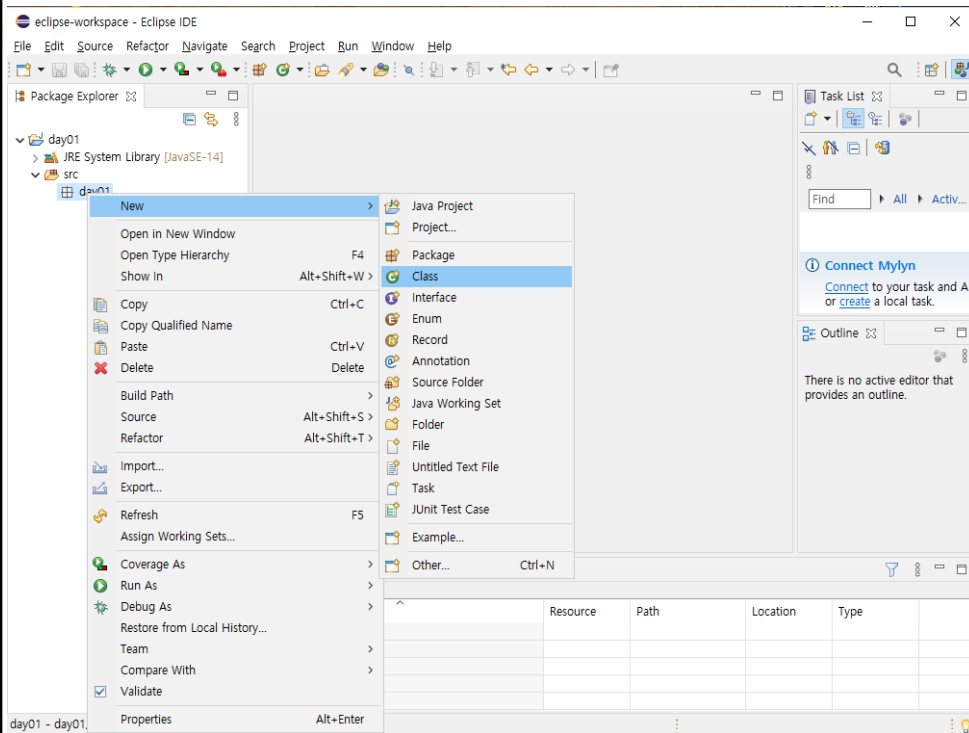
# 1. 프로젝트 만들기

day01 프로젝트 클릭 - src 폴더위 마우스 우클릭  
- New - Package 클릭 -> Finish 클릭



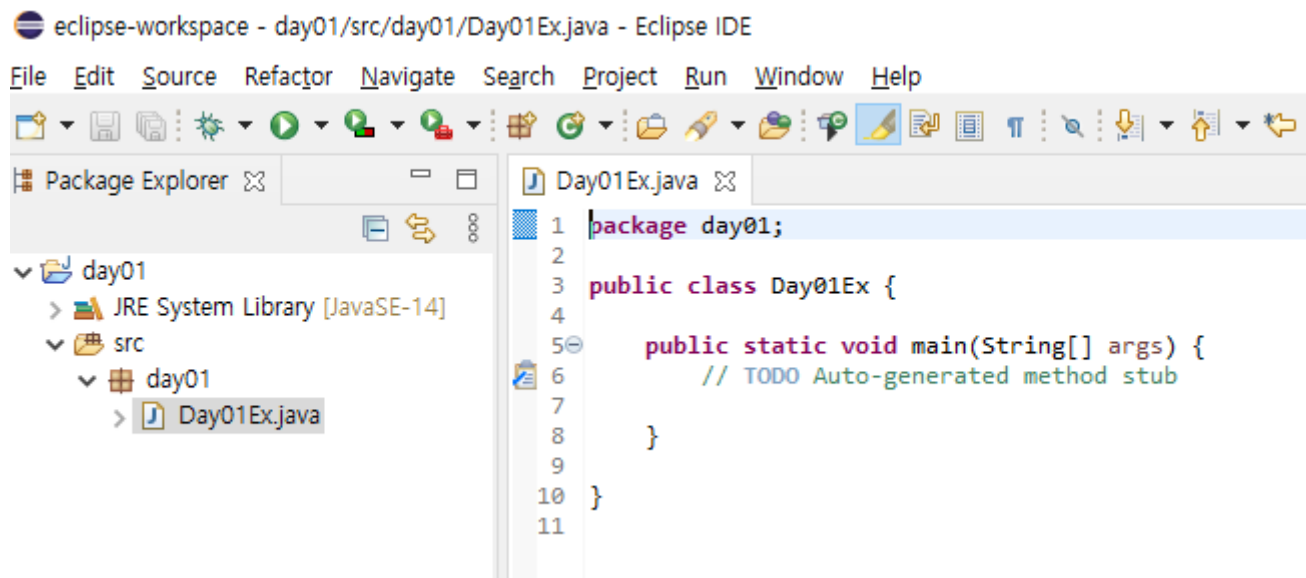
# 1. 프로젝트 만들기

day01 패키지 위 마우스 우클릭-New-Class 클릭  
Name에 Day01Ex 클릭 후 main 체크!



# 1. 프로젝트 만들기

준비완료된 화면!

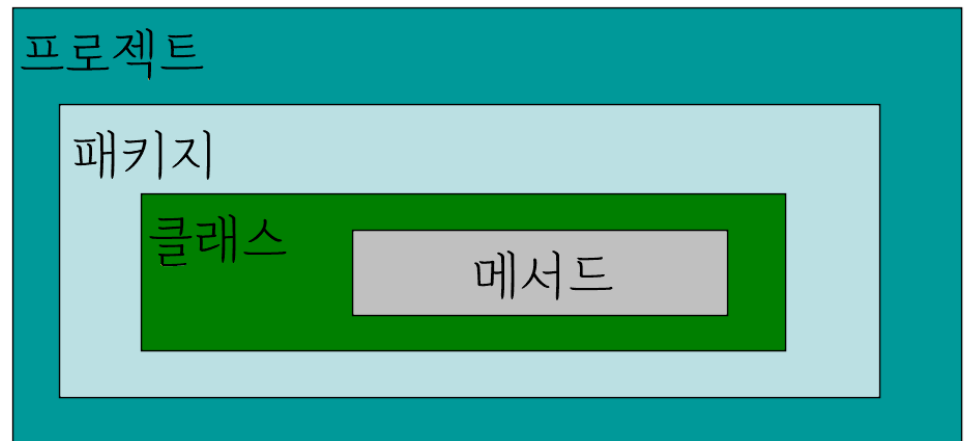
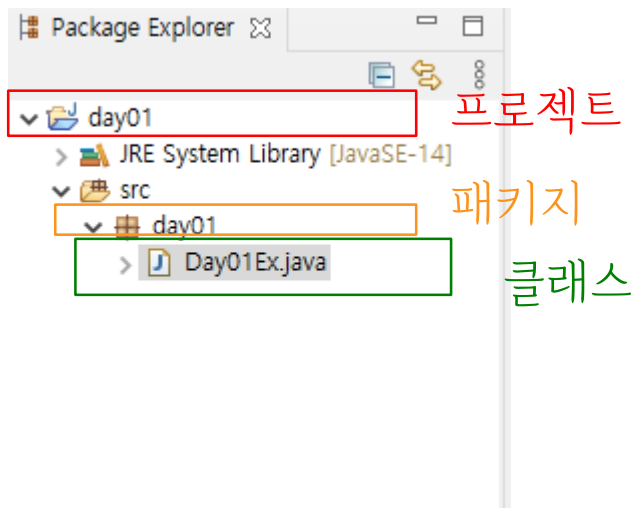




# 1. 프로젝트 만들기

프로그램 설명

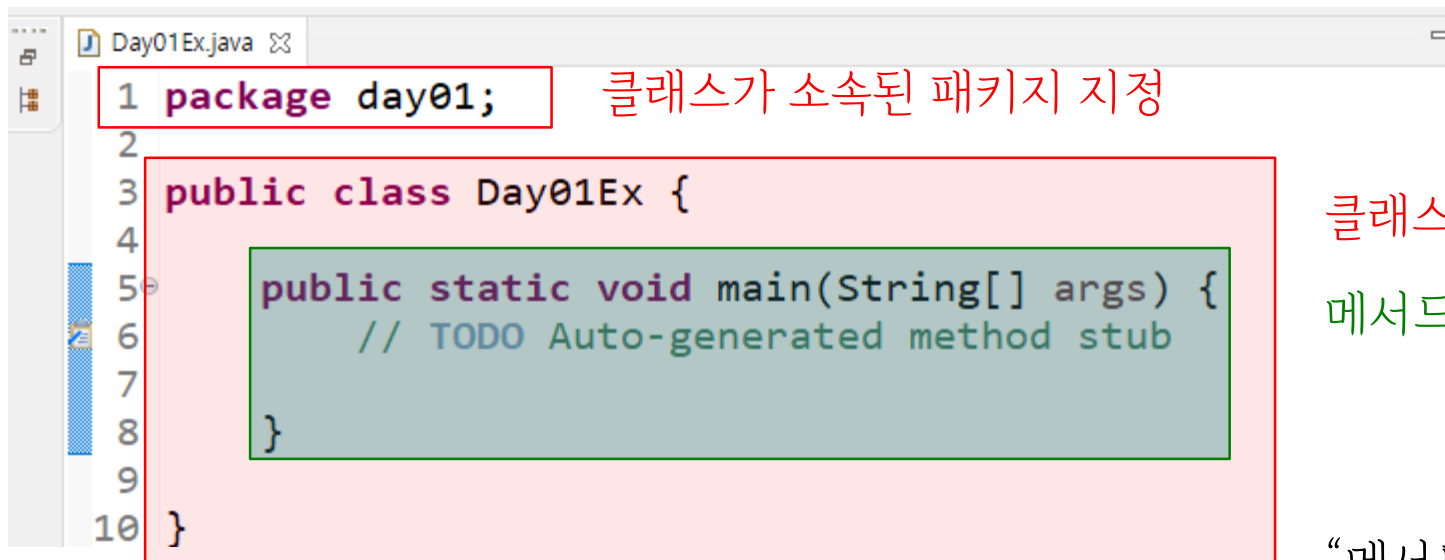
Package Explorer: 프로젝트 구성 확인



# 1. 프로젝트 만들기

프로그램 설명

Day01Ex.java: 코드 작성할 파일



```
1 package day01;
2
3 public class Day01Ex {
4     public static void main(String[] args) {
5         // TODO Auto-generated method stub
6     }
7 }
8
9
10 }
```

The screenshot shows a code editor window titled 'Day01Ex.java'. The code is as follows:

```
1 package day01;
2
3 public class Day01Ex {
4     public static void main(String[] args) {
5         // TODO Auto-generated method stub
6     }
7 }
8
9
10 }
```

Annotations in the image:

- A red box highlights line 1: `package day01;` with the text "클래스가 소속된 패키지 지정" (Specify the package the class belongs to).
- A large red box highlights the entire class definition from line 3 to line 10: `public class Day01Ex { ... }` with the text "클래스 영역" (Class area).
- A green box highlights the method definition from line 5 to line 6: `public static void main(String[] args) { ... }` with the text "메서드 영역" (Method area).

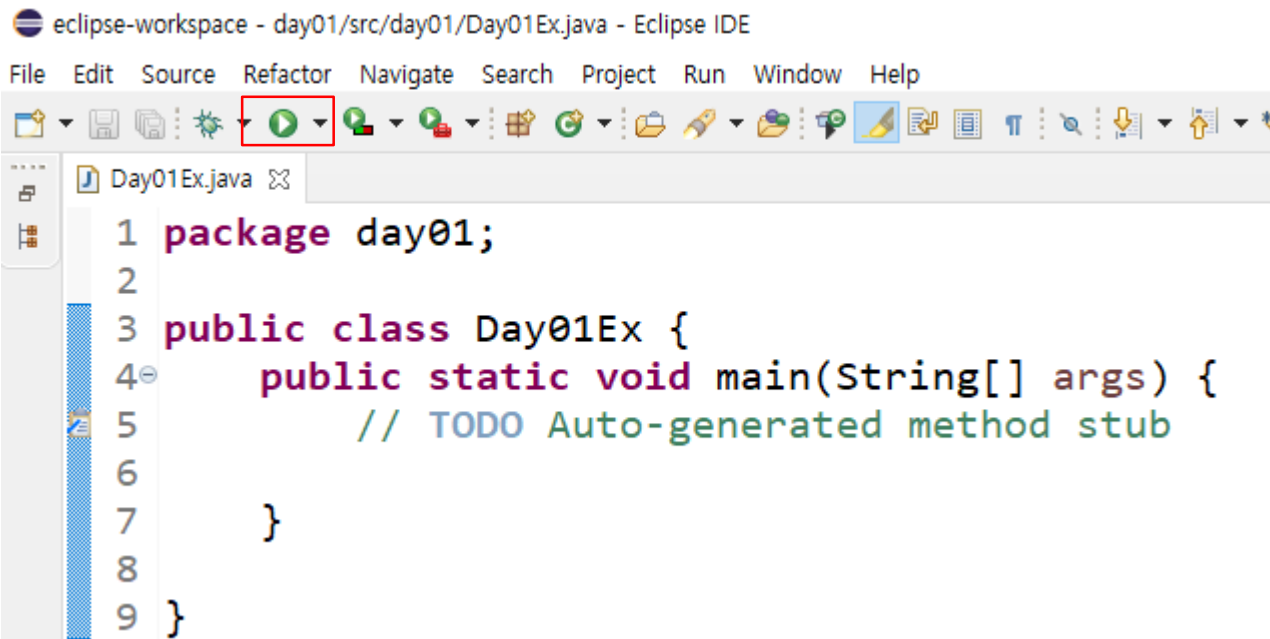
클래스 영역

메서드 영역

“메서드 영역 내에서  
명령을 내릴 예정!”

# 1. 프로젝트 만들기

프로그램 실행: Ctrl+F11 누르기 혹은 아래 아이콘 누르기



현재 main 메서드 안에 명령이 없기 때문에 결과가 안보임(실행은 된다)

## 2. 주석

### 1) 주석이란?

: 프로그램에 아무런 영향이 없이 작성할 수 있는 문장

### 2) 사용 목적?

- 코드에 대한 설명
- 미완성된 코드 보류

## 2. 주석

### 3) 주석 사용 방법

- 한줄 주석: // 주석내용

단축키) Ctrl+/

\* 한번더 단축키 입력 시, 주석 해제

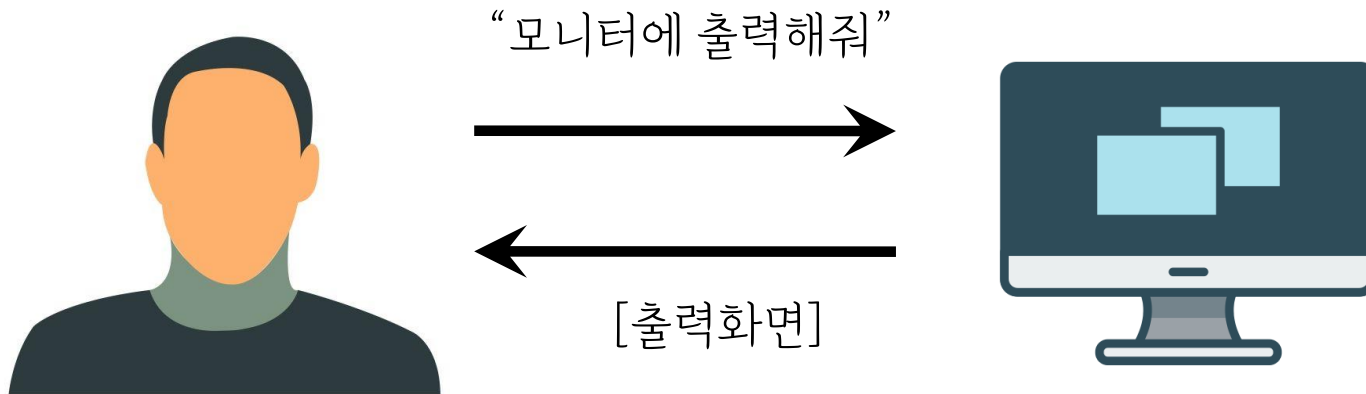
- 여러줄 주석: /\* 주석내용 \*/

단축키) 블럭처리 후 Ctrl+Shift+ /

### 3. 출력메서드

#### 1) 출력 메서드란?

: 개발자가 컴퓨터에게 모니터에 출력해달라고 내릴 수 있는 명령(기능)



### 3. 출력메서드

#### 2) 출력 메서드 사용방법

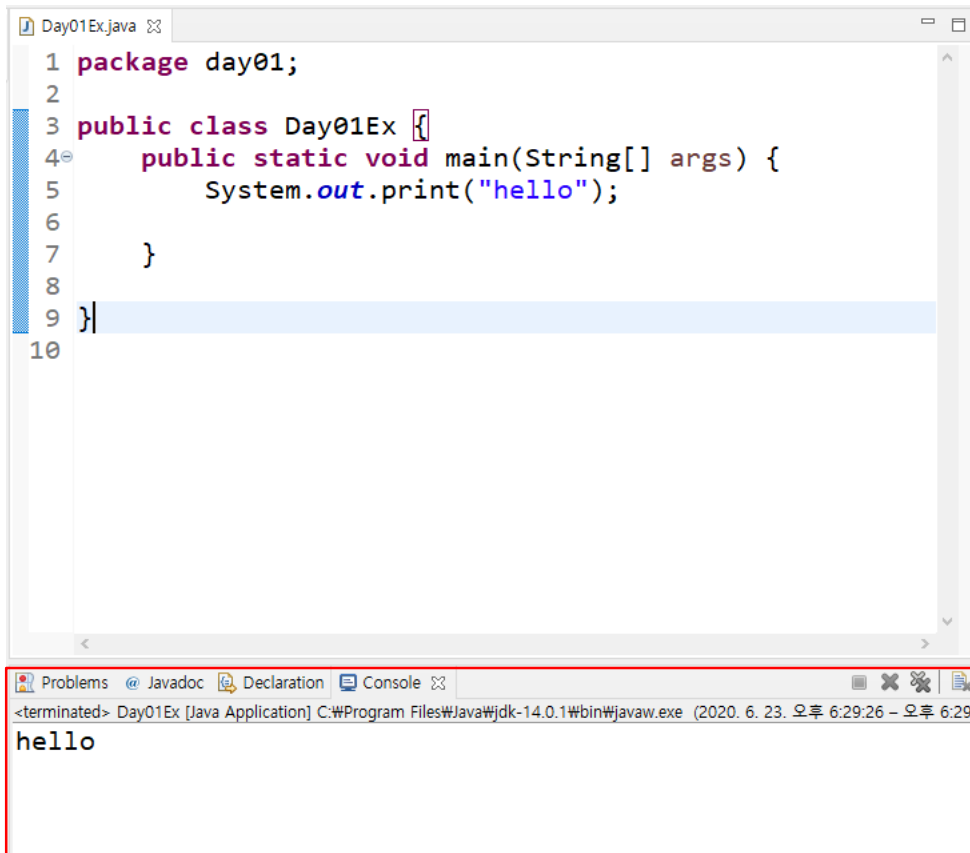
```
System.out.print(" 출력할 내용" );
```

```
1 package day01;  
2  
3 public class Day01Ex {  
4     public static void main(String[] args) {  
5         System.out.print("hello");  
6     }  
7  
8 }
```

명령에 끝에는  
항상 ; 을 붙인다!

# 3. 출력메서드

코드 결과 실행(Ctrl+F11)



The screenshot shows an IDE window with a Java file named 'Day01Ex.java'. The code is as follows:

```
1 package day01;  
2  
3 public class Day01Ex {  
4     public static void main(String[] args) {  
5         System.out.print("hello");  
6     }  
7 }  
8  
9 }  
10
```

Below the code editor, the 'Console' tab is active, showing the output of the program:

```
<terminated> Day01Ex [Java Application] C:\Program Files\Java\jdk-14.0.1\bin\javaw.exe (2020. 6. 23. 오후 6:29:26 - 오후 6:29:26)  
hello
```

결과 확인!



## 3. 출력메서드

### 3) 출력메서드의 종류

(1) `System.out.print(출력값);`

(2) `System.out.println(출력값);`

(3) `System.out.printf(" 형식지정자",출력값);`

# 3. 출력메서드

## 3) 출력메서드의 종류

(1) System.out.print(출력값);

: 줄바꿈 없이 출력값을 그냥 출력해라

```
1 package day01;
2
3 public class Day01Ex {
4     public static void main(String[] args) {
5         System.out.print("문자열값");
6         System.out.print('A'); //문자값
7         System.out.print(100); //정수값
8         System.out.print(3.14); //실수값
9     }
10 }
11
```

Problems Javadoc Declaration Console

<terminated> Day01Ex [Java Application] C:\Program Files\J...  
문자열값A1003.14

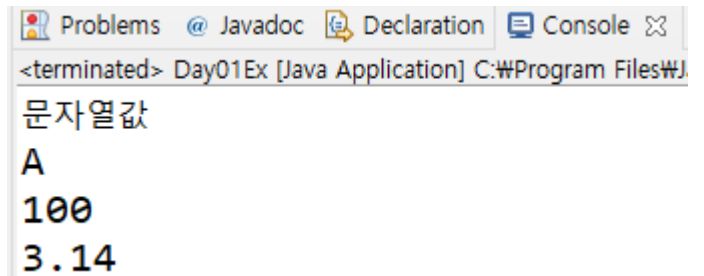
# 3. 출력메서드

## 3) 출력메서드의 종류

(2) System.out.println(출력값);

: 출력값을 출력한 후 줄을 바꿔라

```
1 package day01;
2
3 public class Day01Ex {
4     public static void main(String[] args) {
5         System.out.println("문자열값");
6         System.out.println('A'); // 문자값
7         System.out.println(100); // 정수값
8         System.out.println(3.14); // 실수값
9     }
10 }
11
```



The screenshot shows the IDE's console window with tabs for Problems, Javadoc, Declaration, and Console. The Console tab is active, displaying the output of the program: "문자열값", "A", "100", and "3.14", each on a new line. The window title is "<terminated> Day01Ex [Java Application] C:\Program Files\WJ..."

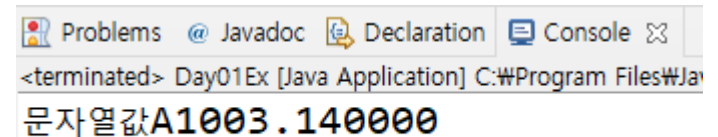
```
<terminated> Day01Ex [Java Application] C:\Program Files\WJ...
문자열값
A
100
3.14
```

# 3. 출력메서드

## 3) 출력메서드의 종류

(3) `System.out.printf("형식지정자",출력값);`  
: 출력값을 내가 지정한 형식으로 출력해라

```
1 package day01;  
2  
3 public class Day01Ex {  
4     public static void main(String[] args) {  
5         System.out.printf("%s", "문자열값");  
6         System.out.printf("%c", 'A'); //문자값  
7         System.out.printf("%d", 100); //정수값  
8         System.out.printf("%f", 3.14); //실수값  
9     }  
10 }
```



The screenshot shows an IDE window with tabs for Problems, Javadoc, Declaration, and Console. The Console tab is active, displaying the output of the program: "문자열값A1003.140000". The text is split across two lines: "문자열값" on the first line and "A1003.140000" on the second line.

# 3. 출력메서드

## 3) 출력메서드의 종류

(3) `System.out.printf(“형식지정자”,출력값);`  
: 출력값을 내가 지정한 형식으로 출력해라

[형식지정자]

%d	정수형
%f	실수형
%c	문자형
%s	문자열형

# 3. 출력메서드

## 3) 출력메서드의 종류

(3) System.out.printf("형식지정자",출력값);  
-> 정수값을 문자형 형식지정자로 출력하면?

```
1 package day01;
2
3 public class Day01Ex {
4     public static void main(String[] args)
5     {
6         System.out.printf("%s", "문자열값");
7         System.out.printf("%c", 65); // 문자값 (아스키코드값)
8         System.out.printf("%d", 100); // 정수값
9         System.out.printf("%f", 3.14); // 실수값
10    }
```

Problems @ Javadoc Declaration Console

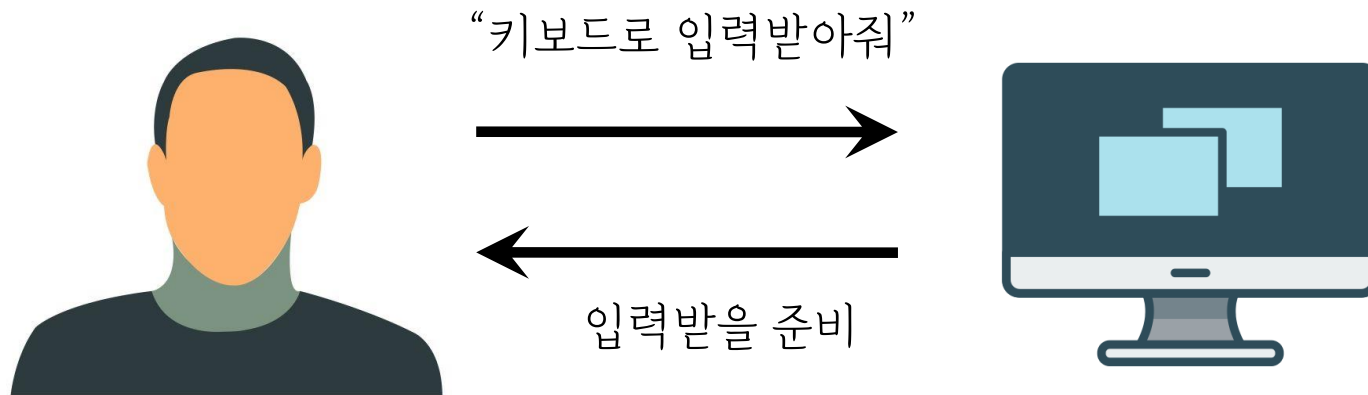
<terminated> Day01Ex [Java Application] C:\Program Files\Ja

문자열값A1003.140000

## 4. 입력메서드

### 1) 입력메서드란?

: 키보드로 입력받을 준비를 해라~



# 4. 입력메서드

## 2) 입력메서드 사용방법

sc.next형식( );

\* 필요한 준비

1) **import** java.util.Scanner;

2) Scanner **sc** = **new** Scanner(System.**in**);

```
Day01Ex.java
1 package day01;
2 import java.util.Scanner;
3 public class Day01Ex {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6         sc.next();
7     }
8 }
9
```



## 4. 입력메서드

### 2) 입력메서드의 종류

(1) `sc.next( );`

(2) `sc.nextInt();`

(3) `sc.nextFloat();`

(4) `sc.nextLine();`

## 4. 입력메서드

### 2) 입력메서드의 종류

(1) `sc.next( );`

: 문자열로 입력받아라

```
Day01Ex.java
1 package day01;
2 import java.util.Scanner;
3 public class Day01Ex {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6         System.out.println(sc.next());
7     }
8 }
9 }
```

Problems @ Javadoc Declaration Console

<terminated> Day01Ex [Java Application] C:\Program Files\Ja

안녕하세요

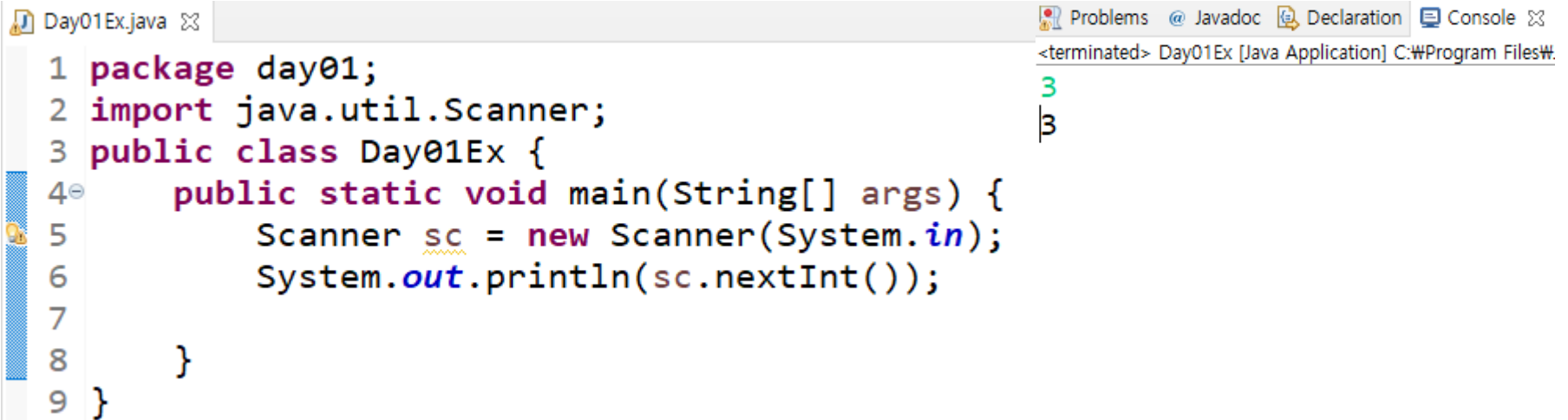
안녕하세요

## 4. 입력메서드

### 2) 입력메서드의 종류

(1) `sc.nextInt( );`

: 정수값으로 입력받아라



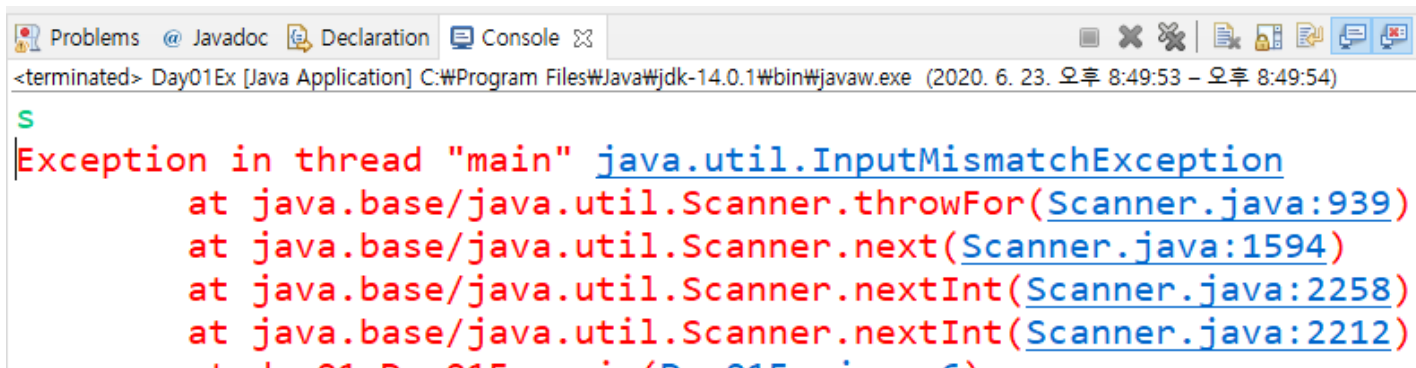
```
Day01Ex.java  Problems  Javadoc  Declaration  Console  <terminated> Day01Ex [Java Application] C:\Program Files\W...
1 package day01;
2 import java.util.Scanner;
3 public class Day01Ex {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6         System.out.println(sc.nextInt());
7     }
8 }
9 }
```

# 4. 입력메서드

## 2) 입력메서드의 종류

(1) `sc.nextInt( );`

: 정수값으로 입력받아라



The screenshot shows an IDE window with a console tab. The title bar indicates the application is 'Day01Ex [Java Application]' running 'C:\Program Files\Java\jdk-14.0.1\bin\javaw.exe' on '2020. 6. 23. 오후 8:49:53 - 오후 8:49:54'. The console output shows a green 'S' followed by a red exception message: 'Exception in thread "main" java.util.InputMismatchException'. The stack trace is as follows:

```
Exception in thread "main" java.util.InputMismatchException
    at java.base/java.util.Scanner.throwFor(Scanner.java:939)
    at java.base/java.util.Scanner.next(Scanner.java:1594)
    at java.base/java.util.Scanner.nextInt(Scanner.java:2258)
    at java.base/java.util.Scanner.nextInt(Scanner.java:2212)
```

정수값이 아닌 문자를 입력할 경우 오류!

## 4. 입력메서드

### 2) 입력메서드의 종류

(1) `sc.nextFloat( );`

: 실수값(float)으로 입력받아라

```
Day01Ex.java
1 package day01;
2 import java.util.Scanner;
3 public class Day01Ex {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6         System.out.println(sc.nextFloat());
7     }
8 }
9 }
```

```
Problems Javadoc Declaration Console
<terminated> Day01Ex [Java Application] C:\Program Files\J
3.14
3.14
```

```
Problems Javadoc Declaration Console
<terminated> Day01Ex [Java Application] C:\Program Files\J
3
3.0
```

## 4. 입력메서드

### 2) 입력메서드의 종류

(1) `sc.nextLine( );`

: 문자열로 **피어쓰기**까지 입력받아라

```
Day01Ex.java
1 package day01;
2 import java.util.Scanner;
3 public class Day01Ex {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6         System.out.println(sc.next());
7     }
8 }
9 }
```

```
Problems @ Javadoc Declaration Console
<terminated> Day01Ex [Java Application] C:\Program Files\J
hello world!
hello
```

# 4. 입력메서드

## 2) 입력메서드의 종류

(1) `sc.nextLine( );`

: 문자열로 **피어쓰기**까지 입력받아라

Day01Ex.java

```
1 package day01;
2 import java.util.Scanner;
3 public class Day01Ex {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6         System.out.println(sc.nextLine());
7     }
8 }
9 }
```

Problems @ Javadoc Declaration Console

<terminated> Day01Ex [Java Application] C:\Program Files\

```
hello world!
hello world!
```