

bantaehak

Chapter 02

변수

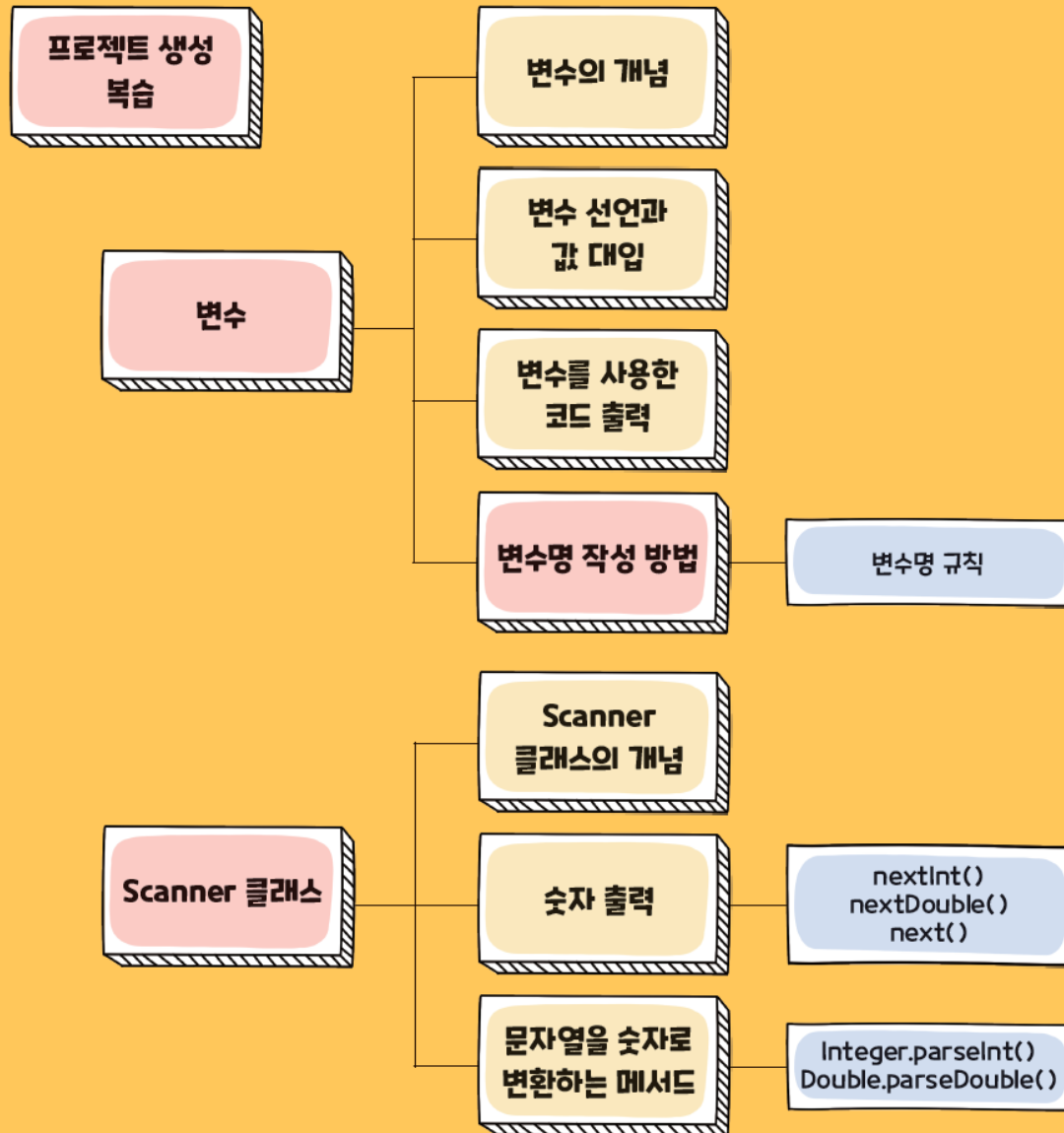


목차

1. 간단한 자바 프로젝트 생성 복습
2. 변수
3. 변수명 작성 방법
4. 값을 입력받는 Scanner 클래스

[실전 예제] 거북이가 나오는 프로그램

Preview



학습목표

- 간단한 계산기를 만들어봅니다.
- 변수의 개념과 사용법을 완전히 이해합니다.
- 키보드로 값을 입력받는 방법을 익힙니다.
- 계산기 응용 프로그램을 작성합니다.
- 터틀 그래픽을 작동하는 기본 방법을 알아봅니다.

Section 01

간단한 자바

프로젝트 생성 복습

1. 프로젝트 생성

■ 프로젝트 생성, 클래스 추가, 코드 작성 및 실행 과정

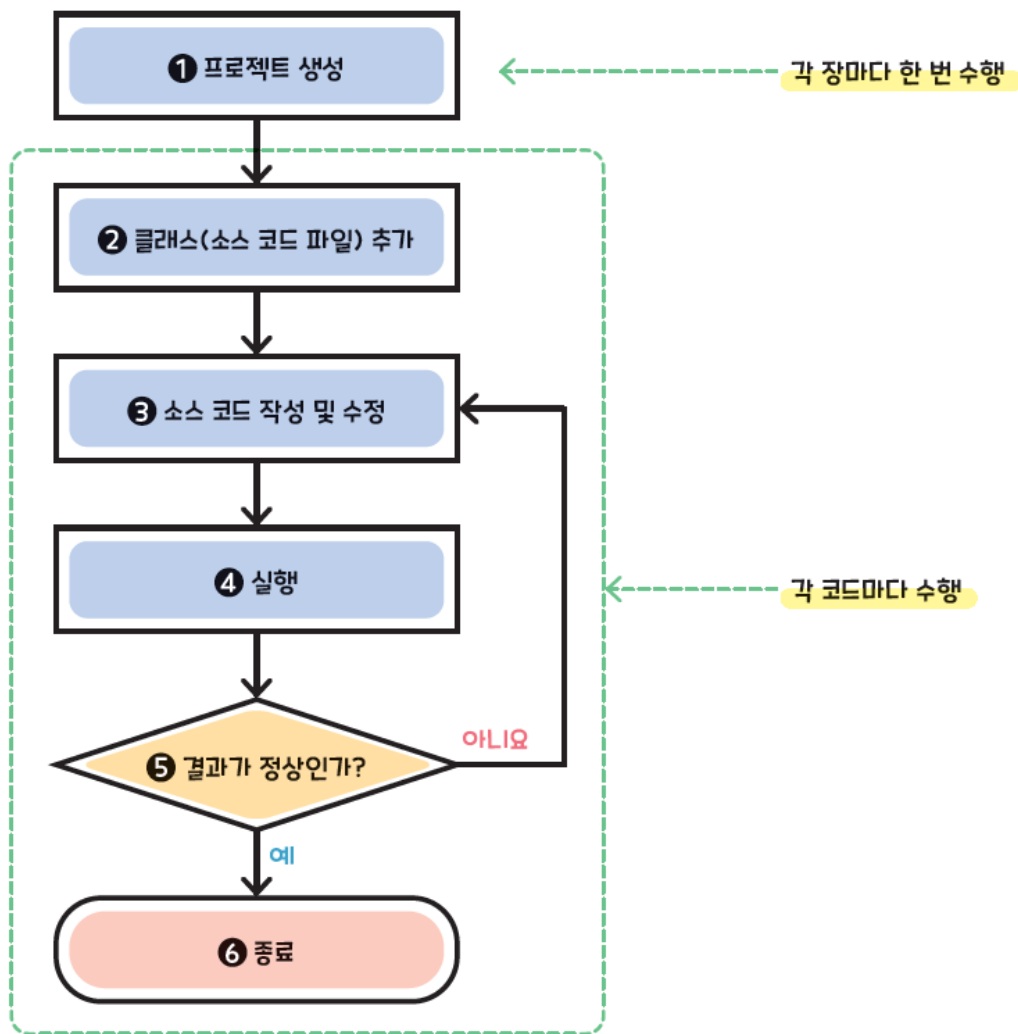


그림 2-1 프로젝트 생성 및 코드 작성 과정

1. 프로젝트 생성

1. 이클립스를 실행한 뒤 [Select a directory as workspace] 창에서 'Workspace'가 'C:\FirstJava'인지 확인하고 <Launch>를 클릭
2. 1장에서 실습한 프로젝트가 열려 있다면 프로젝트명에서 마우스 오른쪽 버튼을 클릭하고 [Close Project]를 선택하여 프로젝트를 닫기

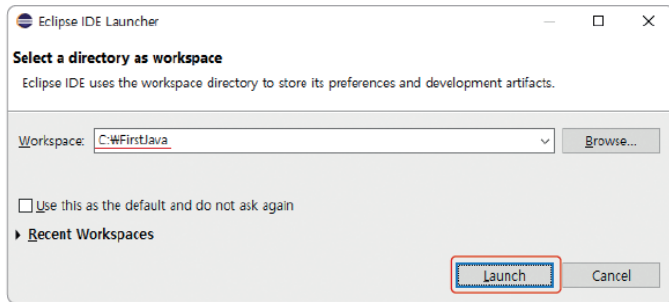


그림 2-2 워크스페이스 선택

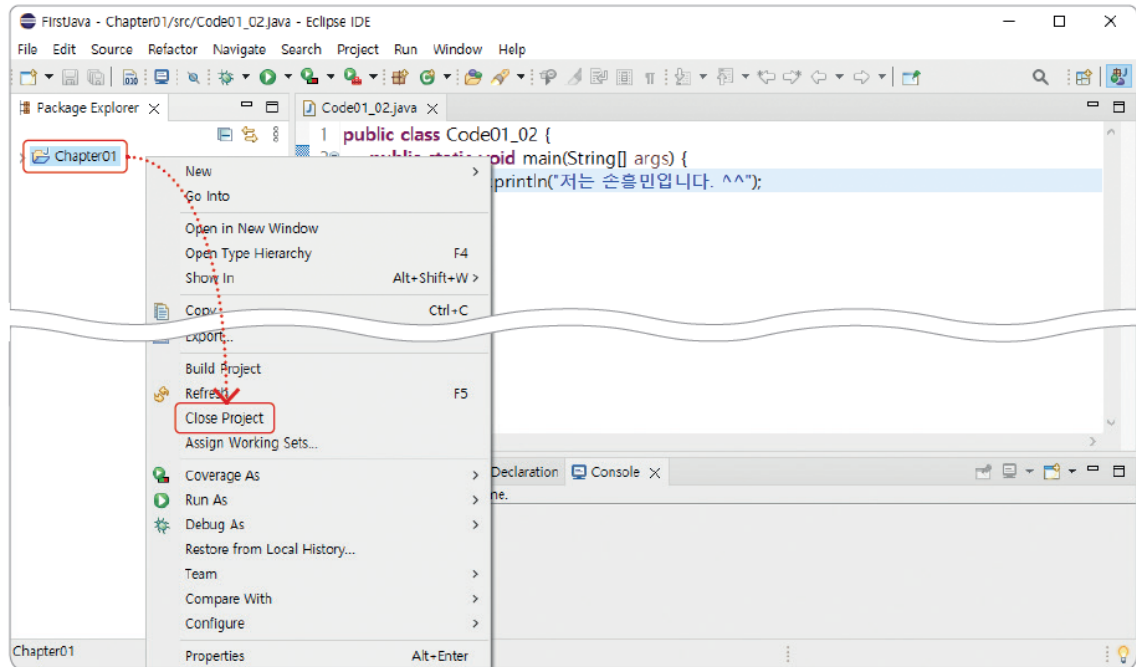


그림 2-3 Chapter01 프로젝트 닫기

1. 프로젝트 생성

1. [Eclipse IDE]가 실행되면 메뉴에서 [File]-[New]-[Java Project] 선택
2. [Create a Java Project] 창에서 'Project name'에 'Chapter02'를 입력하고 아래쪽의 'Module'에서 'Create module-info.java file'의 체크가 해제되었는지 확인하고 이제 <Finish>를 클릭하 프로젝트 생성

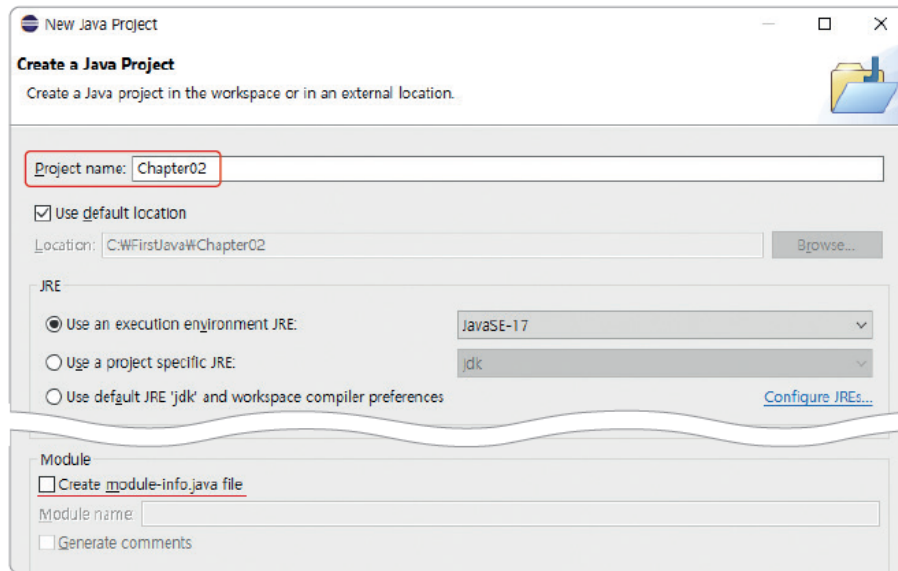
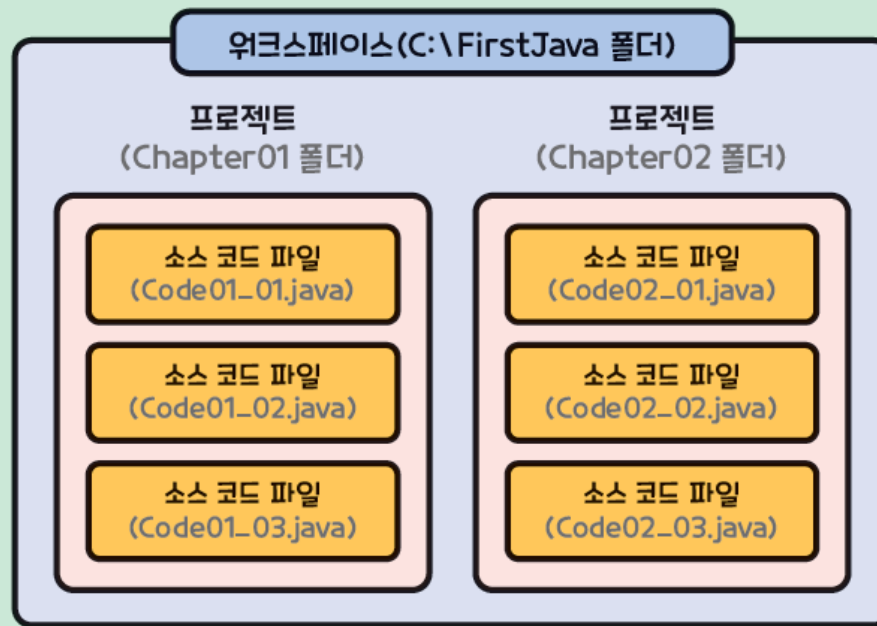


그림 2-4 프로젝트 생성

1. 프로젝트 생성

[하나 더 알기] 워크스페이스, 프로젝트, 클래스의 관계

워크스페이스는 프로젝트를 모아놓은 작업 공간으로, 폴더와 매한가지입니다. 이 책의 모든 프로젝트는 C:\FirstJava 폴더에 생성되며, 이 폴더가 바로 워크스페이스입니다. 다음 그림은 워크스페이스, 프로젝트, 소스 코드 파일(클래스)의 관계를 보여줍니다.



2. 클래스 추가

1. [Package Explorer]의 프로젝트명(Chapter02)을 확장한 뒤 src 폴더에서 마우스 오른쪽 버튼을 클릭하고 [New]-[Class]를 선택
2. [Java Class] 창에서 'Name'에 'Code02_01'을 입력한 뒤 아래쪽의 'public static void main(String[] args)'에 체크하고 <Finish> 클릭

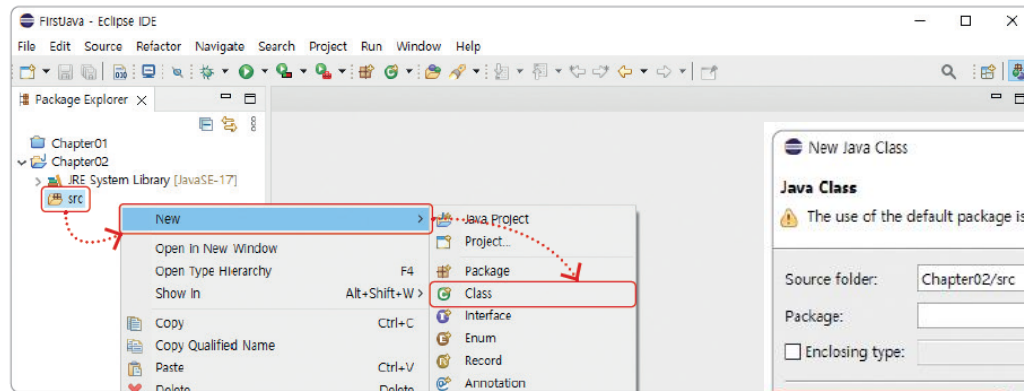


그림 2-5 클래스(소스 파일) 생성 1

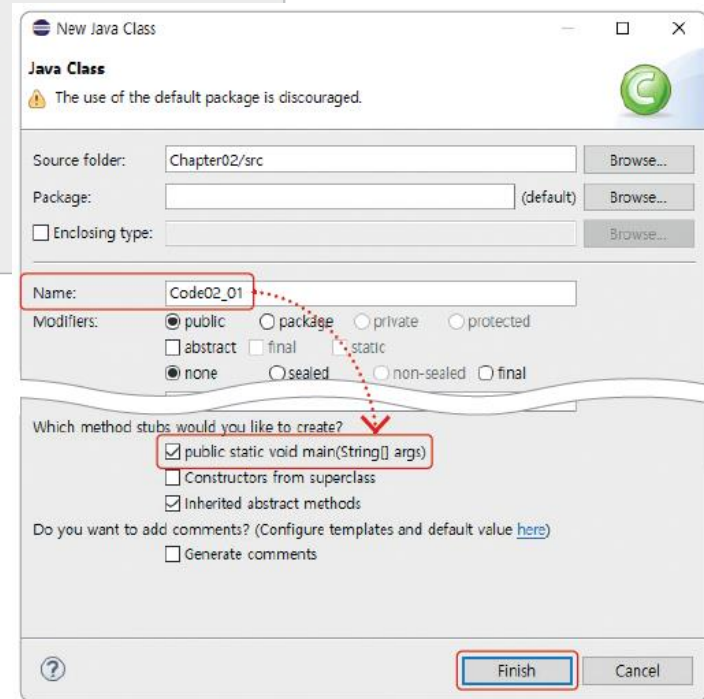


그림 2-6 클래스(소스 파일) 생성 2

3. 코드 작성 및 수정

1. 자동 생성된 코드의 주석(//)을 지우고 코드를 입력
 - 이때 빈 행을 삭제해도 됨
2. 메뉴에서 [File]-[Save]를 선택하거나 <Ctrl> + <S>를 눌러 저장

코드 2-1

Code02_01.java

```
01 public class Code02_01 {  
02     public static void main(String[] args) {  
03         System.out.println(100+200);  
04     }  
05 }
```

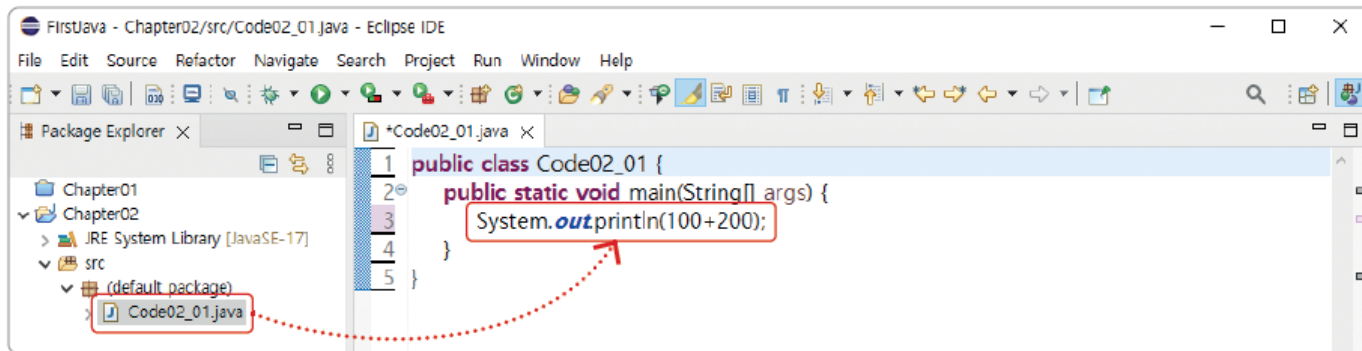


그림 2-7 코드 입력

3. 코드 작성 및 수정

[하나 더 알기] 자바의 주석

- `//`
`//` 이후의 내용은 한 행만 주석으로 처리됩니다

```
// 이 행은 주석입니다.  
System.out.println(100); // 이 부분은 주석입니다.
```

- `/*~*/`
여러 행을 묶어서 주석으로 처리할 때 사용합니다

```
/*  
이 행은 주석입니다.  
이 행도 주석입니다.  
*/
```

4. 실행 및 결과 확인

1. 메뉴에서 [Run]-[Run]을 선택하거나 <Ctrl> + <F11>을 눌러 코드 실행

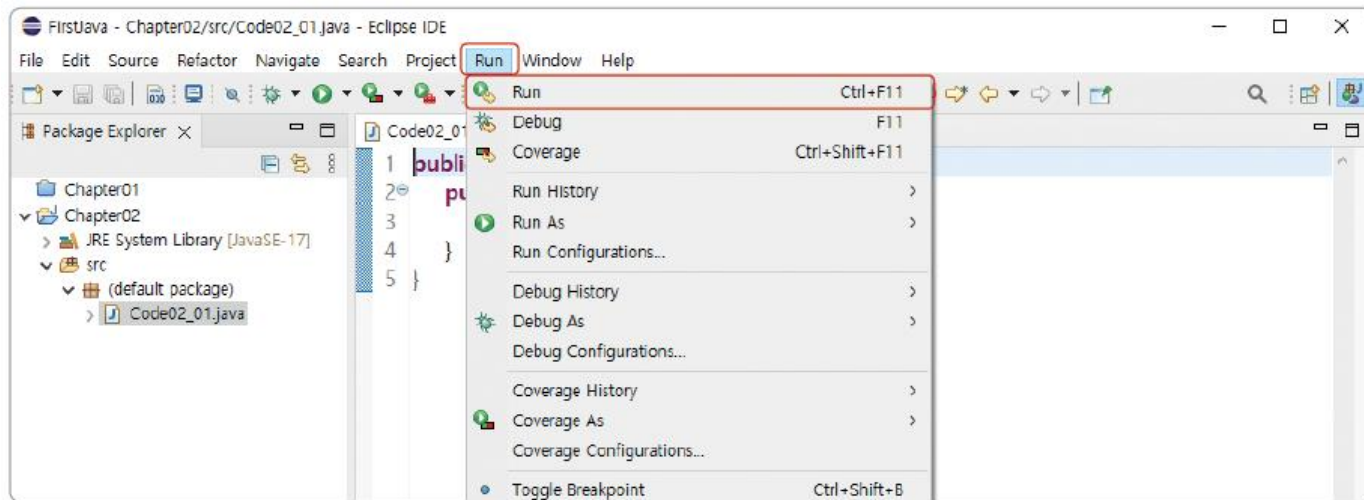


그림 2-8 코드 실행

4. 실행 및 결과 확인

2. 아래쪽 [Console]에 출력된 내용이 정상적인 결과인지 확인

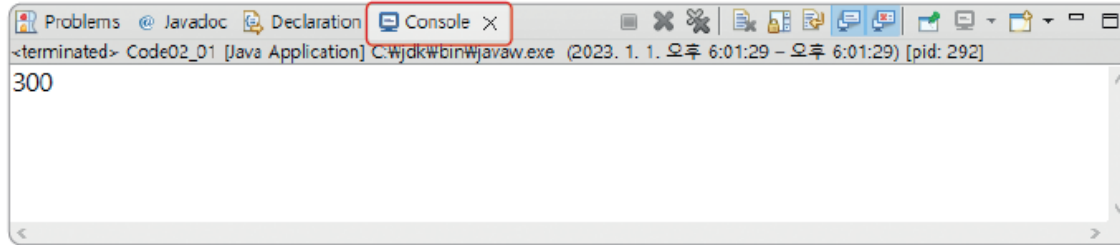


그림 2-9 정상적인 실행 결과

3. 만약 오류가 발생하거나 원하는 결과가 아니라면 [코드 2-1]과 다른 부분이 있는지 확인하고 코드를 수정

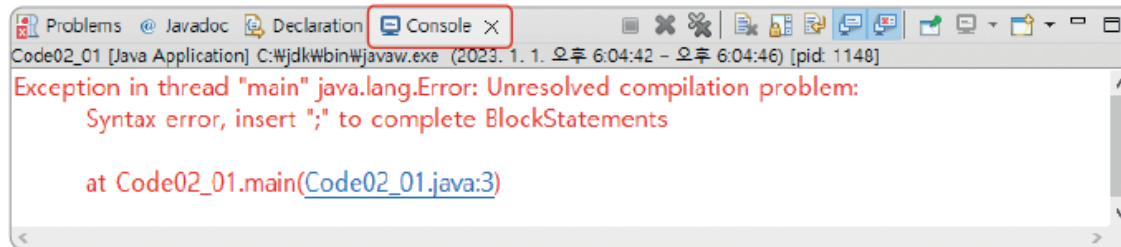


그림 2-10 비정상적인 실행 결과(오류 발생)

4. 실행 및 결과 확인

4. 코드가 정상적으로 실행된 것을 확인했다면 메뉴에서 [File]-[Close Editor]를 선택하거나 [Code02_01.java] 창의 [x]를 클릭하여 코드 창을 닫기

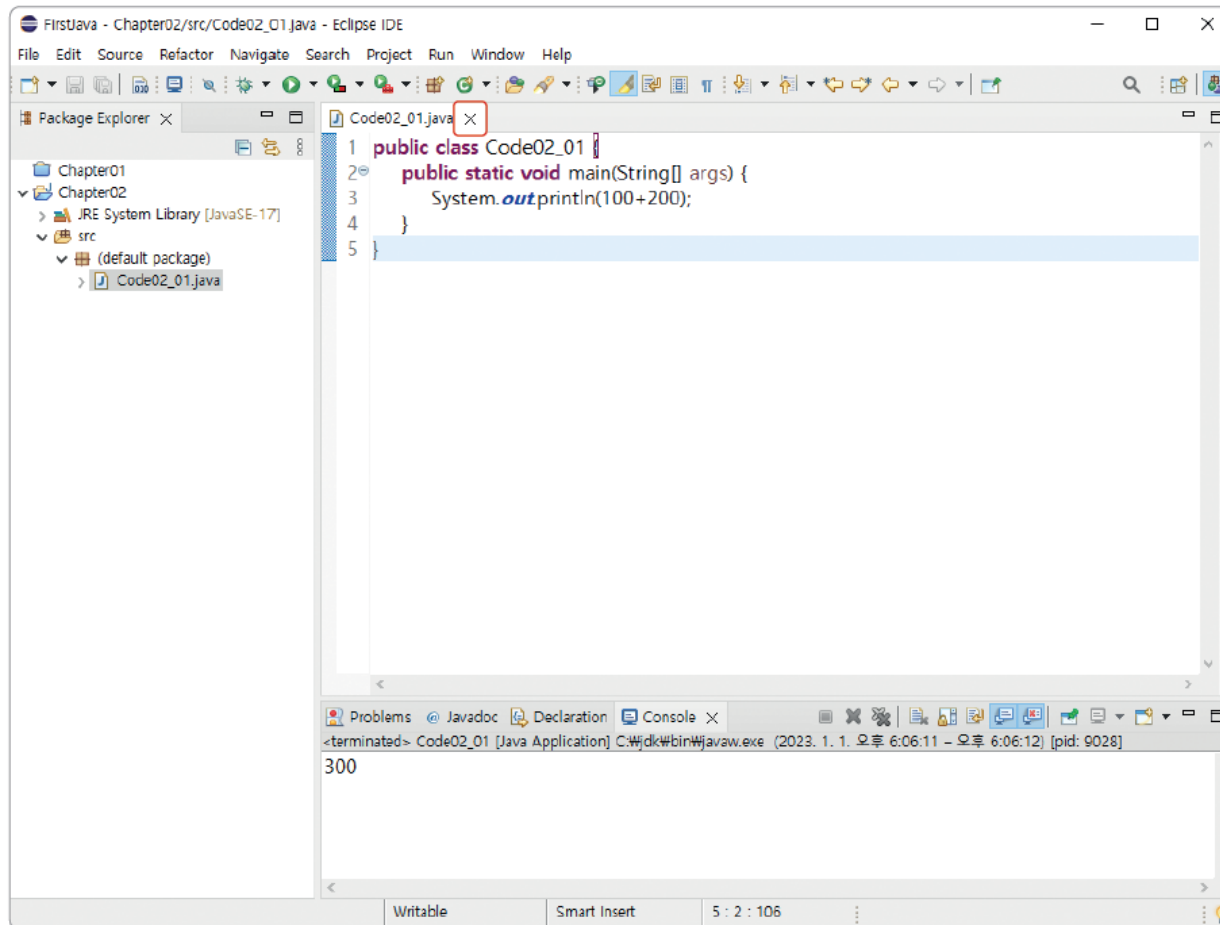


그림 2-11 코드 창 닫기

Section 02

변수

1. 변수의 개념

■ 그릇에 비유되는 변수

- 변수: 값을 저장하는 메모리 공간
 - 변수는 무엇을 담는 그릇이라고 할 수 있음
- 100과 200을 저장하려면 이 두 값을 담을 그릇(변수) 두 개가 필요함

코드 2-2

Code02_02.java

```
01  public class Code02_02 {  
02      public static void main(String[] args) {  
03          int a = 100;  
04          int b = 200;  
05      }  
06  }
```

아무것도 출력되지 않음

- 화면에 값을 출력하는 메서드인 `System.out.println()`을 사용하지 않으면 화면에 아무것도 나오지 않음
- 또한 자바 코드는 문장의 끝에 반드시 세미콜론(`;`)을 넣어야 함

1. 변수의 개념

■ 그릇에 비유되는 변수

- int: 정수형을 뜻하는 integer의 약자
- 대입 연산자(=): 오른쪽의 값을 왼쪽에 넣으라는 의미

a ← 100
b ← 200

그림 2-12 대입 연산자 =의 의미

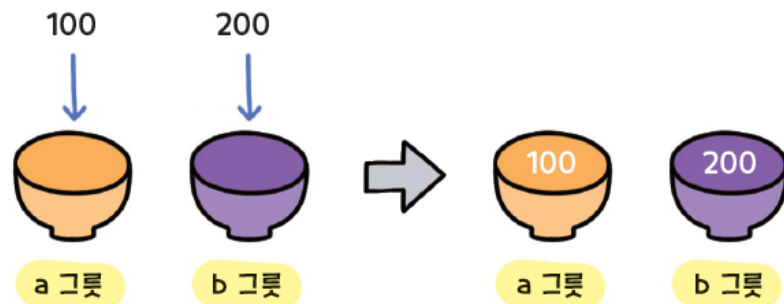


그림 2-13 변수에 값 대입(그릇에 값 넣기)

[하나 더 알기] 그릇의 종류

그릇에는 밥그릇, 국그릇, 반찬 그릇 등이 있고 밥그릇에는 밥을, 국그릇에는 국을, 반찬 그릇에는 반찬을 담습니다. 이와 마찬가지로 변수도 정수(int)를 담는 정수형 변수, 실수(double)를 담는 실수형 변수, 문자(char)를 담는 문자형 변수, 문자열(String)을 담는 문자열형 변수 등 여러 가지 종류가 있습니다. 이를 데이터형이라고 부르는데, 이에 대해서는 4장에서 자세히 살펴보겠습니다.

1. 변수의 개념

■ 그릇에 비유되는 변수

- 두 변수에 들어 있는 값을 더하면 어딘가에 담아야 하므로 새로운 변수가 필요함
- a의 100과 b의 200을 합한 값인 300을 c 그릇에 넣는 코드

코드 2-3

Code02_03.java

```
01 public class Code02_03 {  
02     public static void main(String[] args) {  
03         int a = 100;  
04         int b = 200;  
05  
06         int c = a + b;  
07     }  
08 }
```

아무것도 출력되지 않음

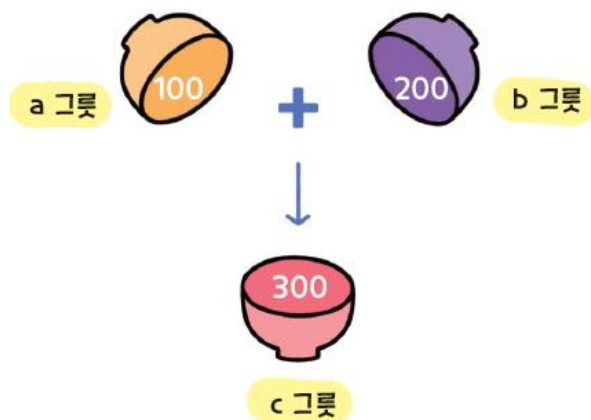


그림 2-14 변수의 값 더하기(더한 값을 그릇에 담기)

1. 변수의 개념

■ 값 출력

- 결과를 저장한 c 그릇의 내용을 System.out.println()을 사용하여 출력하는 코드

코드 2-4

Code02_04.java

```
01 public class Code02_04 {  
02     public static void main(String[] args) {  
03         int a = 100;  
04         int b = 200;  
05  
06         int c = a + b;  
07         System.out.println(c);  
08     }  
09 }
```

300

1. 변수의 개념

■ 값 출력

- 전체 계산식($100 + 200 = 300$)이 출력되도록 코드를 수정

코드 2-5

Code02_05.java

```
01 public class Code02_05 {  
02     public static void main(String[] args) {  
03         int a = 100;  
04         int b = 200;  
05  
06         int c = a + b;  
07         System.out.println(a + "+" + b + "=" + c);  
08     }  
09 }
```

$100 + 200 = 300$

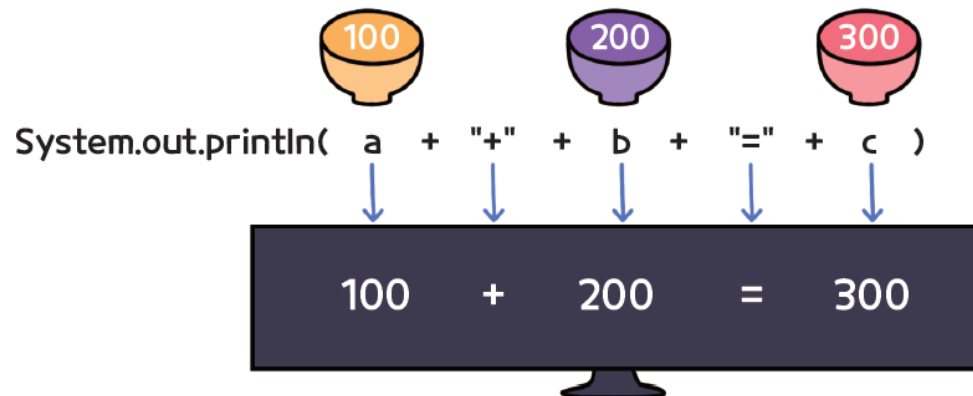


그림 2-15 System.out.println()의 기능

1. 변수의 개념

[하나 더 알기] 메서드

메서드(method)는 특정 기능을 하도록 미리 만들어놓은 것으로, 무조건 뒤에 괄호가 붙습니다. [코드 2-5]에서 사용한 `System.out.println()` 메서드는 괄호 안의 내용을 화면에 출력하는 기능을 합니다. `System.out.println()`은 자바가 미리 만들어서 제공하는 메서드로, 이 밖에도 다양한 메서드가 있습니다. 앞으로 프로젝트에 새로운 메서드가 등장할 때마다 소개할 것이며, 메서드에 대한 자세한 내용은 10장에서 다루겠습니다.

1. 변수의 개념

확인문제

다음 빈칸에 알맞은 말을 넣으시오.

- 변수와 그릇은 비슷한 개념이다. `int a=100;`은 정수형 변수 에 값 을 대입하라는 의미이다.
- 화면에 변수 또는 값을 출력하는 메서드는 이다.

정답

Click!

2. 변수 선언과 값 대입

■ 변수를 선언하고 값을 대입할 때 주의해야 할 점

■ 올바른 값 대입

→ 대입 연산자인 =의 오른쪽 부분이 모두 계산된 후 왼쪽에 대입된다는 것을 주의해야 함

코드 2-6

Code02_06.java

```
01 public class Code02_06 {  
02     public static void main(String[] args) {  
03         int num1, num2, result;  
04  
05         num1 = 100;  
06         num2 = 50;  
07  
08         result = num1 + num2;  
09         System.out.println(num1 + "+" + num2 + "=" + result);  
10     }  
11 }
```

100 + 50 = 150

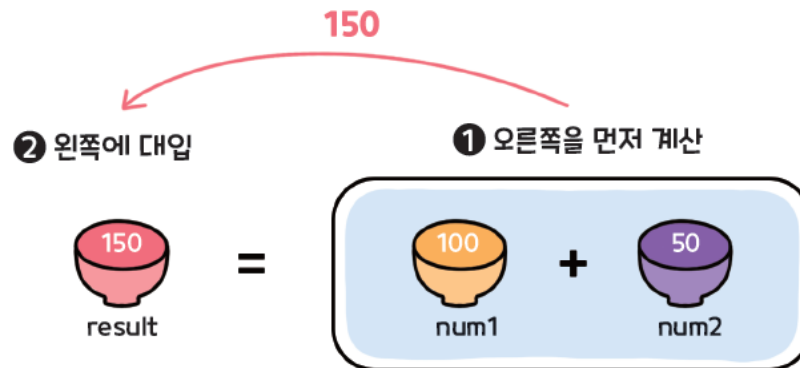


그림 2-16 대입 연산자의 정상적인 구조 1

2. 변수 선언과 값 대입

■ 변수를 선언하고 값을 대입할 때 주의해야 할 점

■ 올바른 값 대입

- 모든 코드에서 =의 왼쪽에는 변수가 있어야 함
- 하지만 =의 오른쪽이 모두 변수일 필요는 없음
- = 오른쪽에서는 값과 값의 연산이든 변수와 값의 연산이든 문제없이 처리됨

```
result = num1 + 200;
```

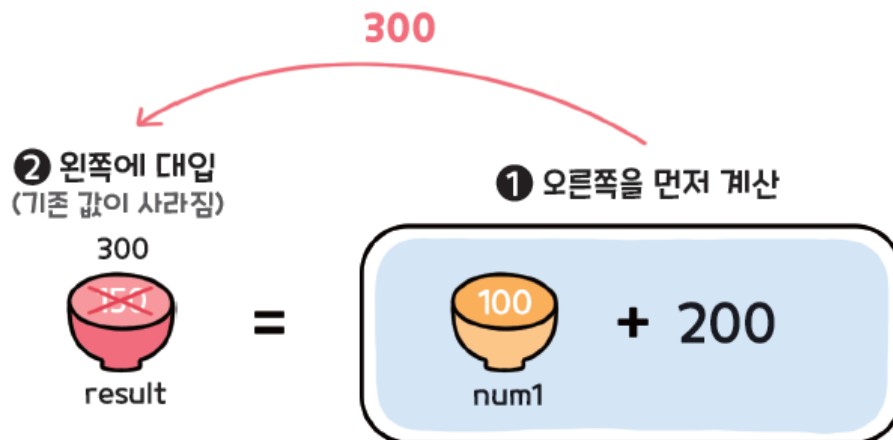


그림 2-17 대입 연산자의 정상적인 구조 2

2. 변수 선언과 값 대입

■ 변수를 선언하고 값을 대입할 때 주의해야 할 점

■ 잘못된 값 대입

→ 대입 연산자의 왼쪽에 무조건 변수가 있어야 함

```
100 = num1 + num2;
```

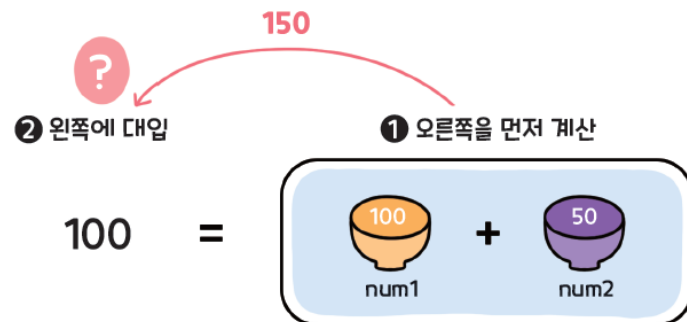


그림 2-18 대입 연산자의 비정상적인 구조 1

→ 대입 연산자의 왼쪽에 변수가 있지만 잘못된 코드

```
num1 + num2 = result;
```

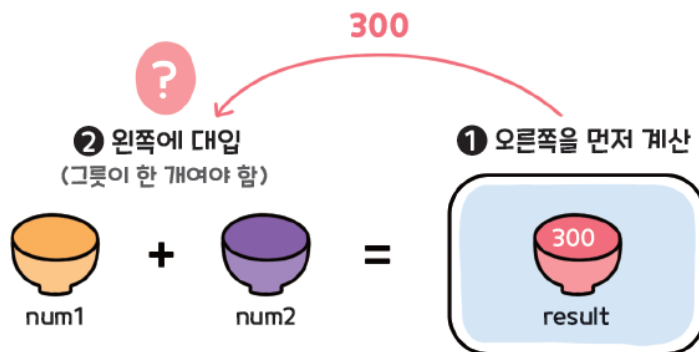


그림 2-19 대입 연산자의 비정상적인 구조 2

2. 변수 선언과 값 대입

확인문제

다음 중 오류가 발생하는 것을 모두 고르시오.

- ① `100 = 150 + 200;`
- ② `result = 100 + 50;`
- ③ `result + 100 = 50;`
- ④ `result = num1 + num2;`

정답

Click!

3. 변수를 사용한 코드 출력1: 숫자

■ 빼기 연산

```
result = num1 - num2 ;  
System.out.println(num1 + "-" + num2 + "=" + result);
```

100-50=50

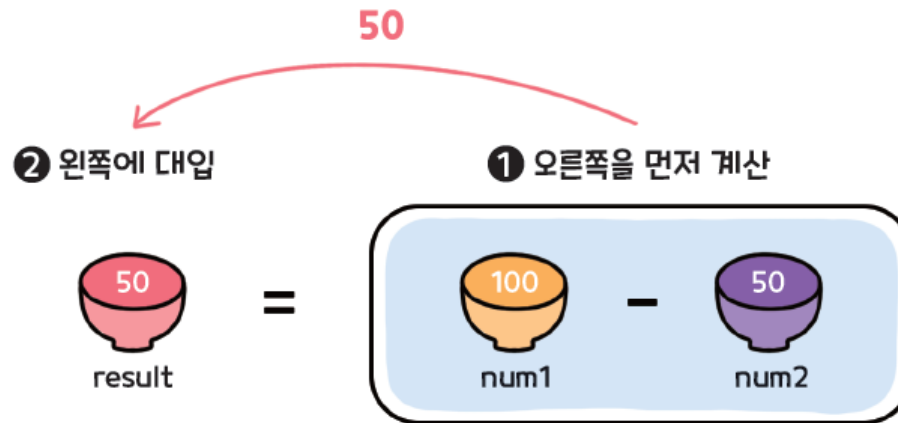


그림 2-20 빼기 연산

3. 변수를 사용한 코드 출력1: 숫자

■ 곱하기 연산

```
result = num1 * num2;  
System.out.println(num1 + "x" + num2 + "=" + result);
```

100x50=5000

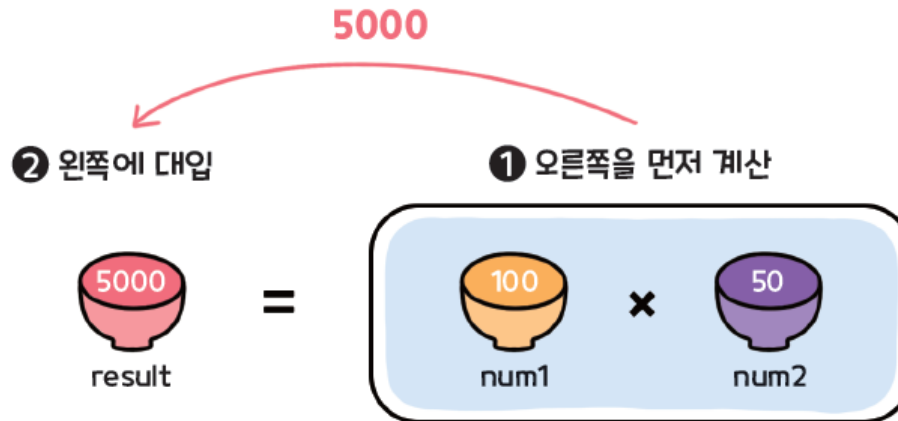


그림 2-21 곱하기 연산

3. 변수를 사용한 코드 출력1: 숫자

■ 나누기 연산

```
result = num1 / num2;  
System.out.println(num1 + "/" + num2 + "=" + result);
```

100/50=2

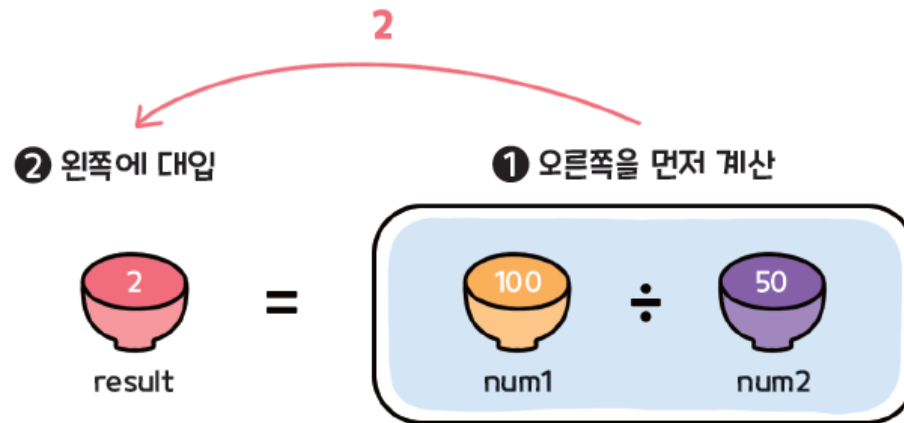


그림 2-22 나누기 연산

3. 변수를 사용한 코드 출력1: 숫자

■ 사칙연산 계산기 코드

코드 2-7

Code02_07.java

```
01 public class Code02_07 {  
02     public static void main(String[] args) {  
03         int num1, num2, result;  
04         num1 = 100;  
05         num2 = 50;  
06  
07         result = num1 + num2;  
08         System.out.println(num1 + "+" + num2 + "=" + result);  
09  
10         result = num1 - num2;  
11         System.out.println(num1 + "-" + num2 + "=" + result);  
12  
13         result = num1 * num2;  
14         System.out.println(num1 + "x" + num2 + "=" + result);  
15  
16         result = num1 / num2;  
17         System.out.println(num1 + "/" + num2 + "=" + result);  
18     }  
19 }
```

100+50=150
100-50=50
100x50=5000
100/50=2

4. 변수를 사용한 코드 출력1: 숫자

■ 변수에 문자열 대입

- 문자열(String)에는 큰따옴표(" ")를 사용
- 한 글자로 이루어진 문자(char)에는 작은따옴표(' ')를 사용

```
String str1 = "난생처음";  
String str2 = "자바";  
System.out.print(str1);  
System.out.println(str2);
```

난생처음자바



그림 2-23 변수에 문자열 대입

4. 변수를 사용한 코드 출력1: 숫자

■ 문자열의 더하기·빼기 연산

- 더하기 연산: 문자열 연결
- 빼기 연산: 오류 발생

```
String result = str1 + str2;  
System.out.println(result);
```

난생처음자바



그림 2-24 문자열의 더하기 연산

4. 변수를 사용한 코드 출력1: 숫자

■ 문자열의 더하기.빼기 연산

코드 2-8

Code02_08.java

```
01  public class Code02_08 {  
02      public static void main(String[] args) {  
03          String str1 = "난생처음";  
04          String str2 = "자바";  
05  
06          System.out.print(str1);  
07          System.out.println(str2);  
08  
09          String result = str1 + str2;  
10  
11          System.out.println(result);  
12      }  
13  }
```

난생처음자바
난생처음자바

5. 간단한 사칙연산 계산기

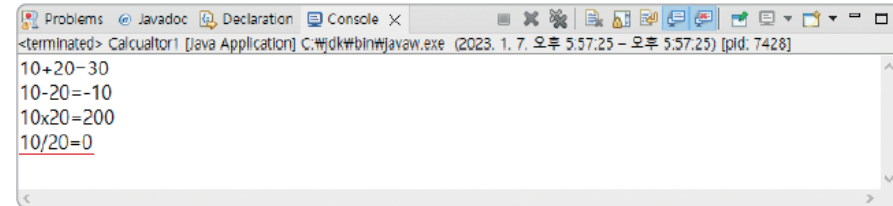
■ 10과 20으로 +, -, *, / 연산을 출력하는 계산기

- Q) 다음 코드에서 이상한 점 찾아보기

코드 2-9

Code02_09.java

```
01 public class Code02_09 {  
02     public static void main(String[] args) {  
03         int number1 = 10, number2 = 20;  
04         int resAdd;  
05         int resSub;  
06         int resMul;  
07         int resDiv;  
08  
09         resAdd = number1 + number2;  
10         resSub = number1 - number2;  
11         resMul = number1 * number2;  
12         resDiv = number1 / number2;  
13  
14         System.out.println(number1 + "+" + number2 + "=" + resAdd);  
15         System.out.println(number1 + "-" + number2 + "=" + resSub);  
16         System.out.println(number1 + "x" + number2 + "=" + resMul);  
17         System.out.println(number1 + "/" + number2 + "=" + resDiv);  
18     }  
19 }
```



```
<terminated> Calculator1 [Java Application] C:\jdk\bin\javaw.exe (2023. 1. 7. 오후 5:57:25 - 오후 5:57:25) [pid: 7428]  
10+20=30  
10-20=-10  
10x20=200  
10/20=0
```

5. 간단한 사칙연산 계산기

■ 10과 20으로 +, -, *, / 연산을 출력하는 계산기

- A) 실행 결과에서 더하기, 빼기, 곱하기 연산은 문제가 없으나 나누기 연산의 결과는 0.5가 아니라 0으로 나옴

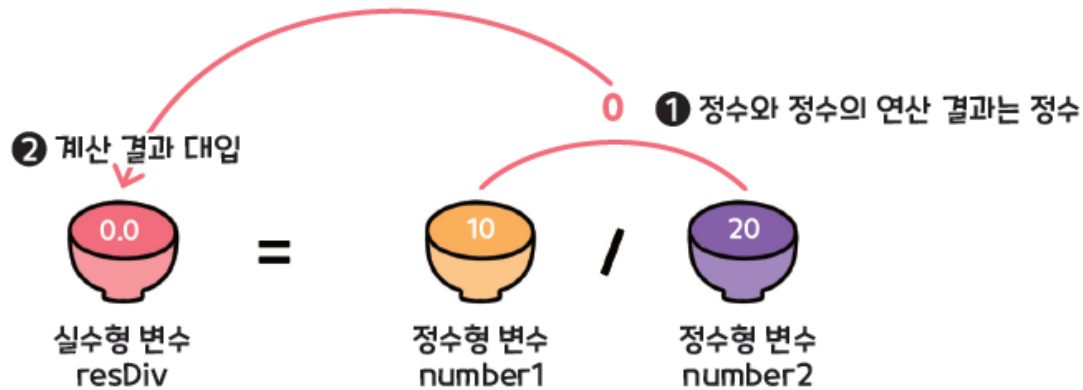


그림 2-25 정수와 정수의 연산

→ 정수와 실수 연산 사이의 규칙

정수	(+, -, *, /)	정수	=	정수
정수	(+, -, *, /)	실수	=	실수
실수	(+, -, *, /)	실수	=	실수

5. 간단한 사칙연산 계산기

■ 10과 20으로 +, -, *, / 연산을 출력하는 계산기

■ 10/20=0.5가 출력되게 하는 방법

→ 변수 앞에 (데이터형) 형식을 붙이면 그 변수의 데이터형이 잠깐 바뀜

→ [코드 2-9]의 7행과 12행을 다음과 같이 수정하고 다시 실행해보기

```
double resDiv;  
resDiv = (double) number1 / number2;
```

10 + 20 = 30

10 - 20 = -10

10 x 20 = 200

10 / 20 = 0.5

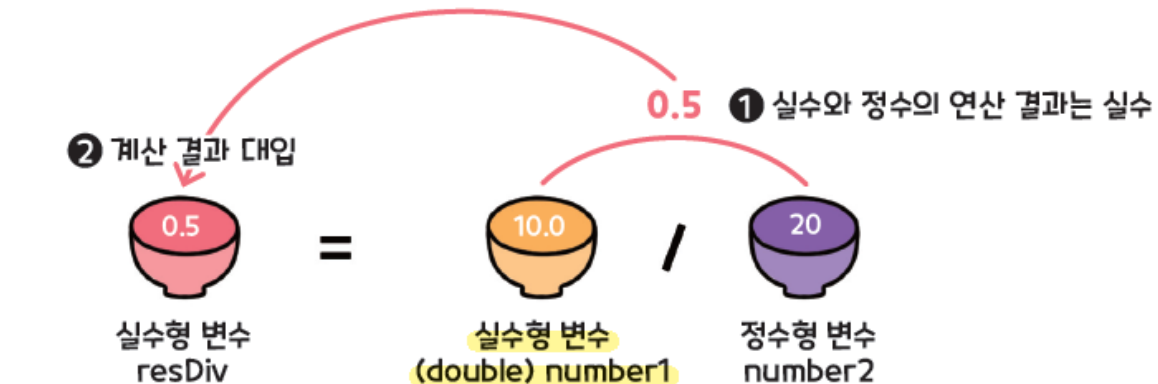


그림 2-26 실수와 정수의 연산

5. 간단한 사칙연산 계산기

확인문제

다음 코드에서 문법상 잘못된 부분을 찾아 바르게 고치시오.

- ① num1=10;
- ② num2=20;
- ③ num1 + num2 = res;
- ④ System.out.println(res);

정답

Click!

Section 03

변수명 작성 방법

1. 변수명 규칙

■ 영문과 숫자 사용 가능

- 변수명은 영문이 맨 앞에 와야 함

→ 변수명을 숫자로만 구성해도 안 되고, 영문과 숫자가 섞였더라도 숫자로 시작하면 안 됨

```
int a = 100;  
int abcd = 200;  
int zzzz = 300;  
int p1234 = 400;  
int love = 500;  
int its4you = 600;
```

- 오류가 발생하는 경우

```
int 333 = 100;  
int 3abcd = 200;
```



한글판 윈도우 환경에서는 변수명에 한글을 지정할 수 있으나 문제가 많이 발생하므로 변수명에 한글을 사용하는 것은 권장하지 않습니다.

1. 변수명 규칙

■ 언더바 사용 가능

- 변수명에는 띄어쓰기를 허용하지 않음
- 띄어쓰기 자리에 언더바(_)를 사용하면 의미를 쉽게 알아볼 수 있음
 - 언더바는 변수명의 맨 앞, 중간, 맨 뒤 어디에든 넣을 수 있음
 - _1과 같은 변수명도 허용

```
int input_number = 100;  
int _number = 200;  
int _data_ = 300;  
int __my__ = 400;  
int _1 = 500;  
int _1234 = 500;
```

1. 변수명 규칙

■ 대·소문자 구분

- 변수명의 대문자와 소문자는 같지 않음
 - 변수명에 대문자와 소문자를 섞으면 언더바 없이도 의미를 부여할 수 있음
 - 변수명은 소문자로 시작하고 클래스명은 대문자로 시작하는 것을 권장함

■ 예약어 사용 불가

- 예약어: 자바에서 이미 문법상 사용하고 있는 단어

표 2-1 자바의 예약어

boolean	byte	char	short	int	long	float	double
private	protected	public	class	abstract	interface	extends	implements
enum	void	return	if	else	switch	case	default
for	do	while	break	continue	true	false	try
catch	finally	throw	throws	transient	volatile	package	import
synchronized	native	final	static	strictfp	assert		

1. 변수명 규칙

■ 클래스명 사용 지양

- 클래스명을 변수로 사용하면 문법상 오류가 발생하지는 않음
 - 하지만 클래스 본래의 기능을 잃어버리기 때문에 프로그램이 엉망이 될 수도 있음
- 클래스명을 변수로 사용해 오류가 발생하는 경우

```
int System = 100;  
int String = 200;  
  
int result = System + String;  
System.out.println("난생처음");
```

Exception in thread "main" java.lang.Error: Unresolved compilation problem:
The primitive type int of System does not have a field out

2. 바람직한 변수명

■ 바람직하지 않은 변수명

- 변수의 용도를 알 수 없는 경우

```
int a = 100;  
int bbb = 200;  
int asdf = 300;  
int AA = 400;
```

- 너무 긴 경우

```
int first_number_of_input_values = 100;  
int second_number_of_input_values = 200;
```

■ 바람직한 변수명

- 짧으면서도 의미를 한눈에 파악할 수 있는 경우

```
int first_num = 100;  
int num_input = 200;  
int inputData = 300;
```

2. 바람직한 변수명

확인문제

다음 중 변수명에 대한 설명으로 옳은 것을 고르시오.

- ① 변수명은 영문과 숫자로 시작해야 한다.
- ② 변수명은 언더바로 시작할 수 있다.
- ③ 변수명에 공백을 넣을 수 있다.
- ④ 예약어를 변수명으로 사용할 수 있다.

정답

Click!

Section 04

값을 입력받는 Scanner 클래스

1. Scanner 클래스의 개요

■ Scanner 클래스

- 변수의 값을 키보드로 입력받도록 도와주는 역할을 함

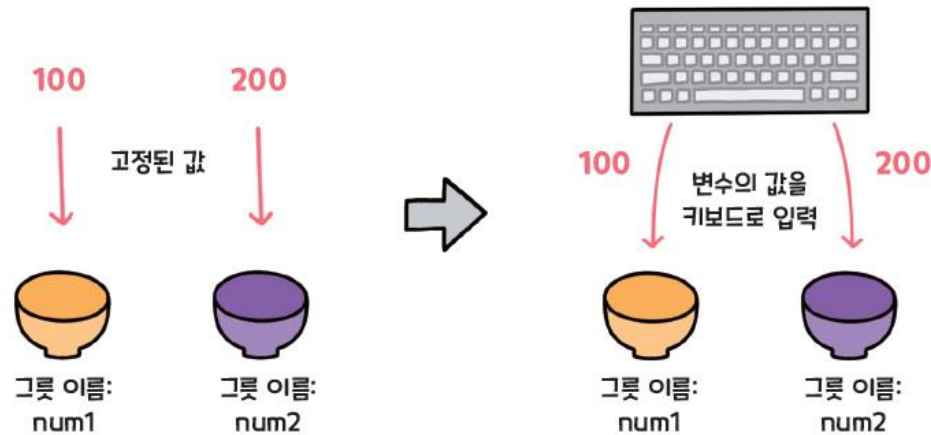


그림 2-27 변수의 값을 키보드로 입력하는 경우

■ Scanner 클래스 사용 형식

- 사용자로부터 값을 입력받고자 할 때는 다음 네 행을 추가해야 함

```
import java.util.Scanner;  
Scanner s = new Scanner(System.in);  
변수명 = s.nextInt();  
s.close();
```

2. 사용자가 입력한 숫자 출력

■ nextInt()

- 사용자로부터 정수를 입력받을 때 사용하는 메서드

코드 2-10

Code02_10.java

```
01 import java.util.Scanner;
02 public class Code02_10 {
03     public static void main(String[] args) {
04         Scanner s = new Scanner(System.in);
05
06         int num;
07
08         num = s.nextInt();
09         System.out.println("사용자가 입력한 값 ==> " + num);
10
11         s.close();
12     }
13 }
```

100 ● ————— 사용자가 입력

사용자가 입력한 값 ==> 100

2. 사용자가 입력한 숫자 출력

■ nextDouble(), next()

- [코드 2-10]을 실행했을 때 '3.14' 또는 '난생처음'을 입력하면 오류가 발생함
→ nextInt()는 정수만 입력받을 수 있기 때문임

```
정수를 입력하세요 : 3.14 ●———— 사용자 입력
Exception in thread "main" java.util.InputMismatchException
at java.base/java.util.Scanner.throwFor(Scanner.java:939)
~~~생략~~~
```

- 실수를 입력받으려면 nextDouble()을, 문자열을 입력받으려면 next()를 사용해야 함

2. 사용자가 입력한 숫자 출력

■ nextDouble(), next()

코드 2-11

Code02_11.java

```
01  import java.util.Scanner;
02  public class Code02_11 {
03      public static void main(String[] args) {
04          Scanner s = new Scanner(System.in);
05
06          double num;
07          System.out.print("실수를 입력하세요 :");
08          num = s.nextDouble();
09          System.out.println("사용자가 입력한 값 ==> " + num);
10
11          String str;
12          System.out.print("문자열을 입력하세요 :");
13          str = s.next();
14          System.out.println("사용자가 입력한 값 ==> " + str);
15
16          s.close();
17      }
18  }
```

실수를 입력하세요 : 3.14 ● ————— 사용자가 입력

사용자가 입력한 값 ==> 3.14

문자열을 입력하세요 : 난생처음 ● ————— 사용자가 입력

사용자가 입력한 값 ==> 난생처음

2. 사용자가 입력한 숫자 출력

확인문제

다음 빈칸에 알맞은 말을 넣으시오.

- 키보드로 값을 입력받는 기능을 제공하는 클래스는 , 정수를 입력받는 메서드는 이다.
- 문자열을 입력받는 메서드는 s. 이고, 실수를 입력받는 메서드는 s. 이다.

정답

Click!

3. 문자열을 숫자로 변환하는 메서드

- Integer.parseInt()
 - 입력받은 문자열을 정수로 변환하는 메서드
- Double.parseDouble()
 - 입력받은 문자열을 실수로 변환하는 메서드

코드 2-12

Code02_12.java

```
01 public class Code02_12 {  
02     public static void main(String[] args) {  
03         int num1;  
04         double num2;  
05         String str;  
06  
07         str = "1234";  
08         num1 = Integer.parseInt(str);  
09         System.out.println("문자열 ==> 정수 : " + num1);  
10  
11         str = "3.14";  
12         num2 = Double.parseDouble(str);  
13         System.out.println("문자열 ==> 실수 : " + num2);  
14     }  
15 }
```

문자열 ==> 정수 : 1234
문자열 ==> 실수 : 3.14

3. 문자열을 숫자로 변환하는 메서드

확인문제

다음 빈칸에 알맞은 말을 넣으시오.

문자열을 정수로 변환하는 메서드는 Integer. 이고, 문자열을 실수로 변환하는 메서드는 Double. 이다.

정답

Click!

[LAB] 택배 배송 정보 입력하기



난생이가 택배를 보내려고 편의점에 갔습니다.

택배를 보내려면 받는 사람의 이름과 주소, 택배의 무게를 입력해야 합니다.

택배비는 그램(g)당 5원이며 자동으로 계산됩니다.

이러한 정보를 입력받아 배송비와 함께 출력하는 프로그램을 만들어봅시다.

```
Console x
<terminated> Lab02_01 [Java Application] C:\Program Files\Java\jdk-11\bin\javaw.exe (2023. 6. 2. 오후 2:15:
## 택배 보내기입니다. 다음을 각각 입력하세요 ##
받는 사람 : 김난생
주소 : 서울 영등포구 여의도동 } — 사용자 입력
무게 (g) : 721
** 받는 사람 ==>김난생
** 주소 ==>서울 영등포구 여의도동
** 배송비 ==>3605원
```



[LAB] 택배 배송 정보 입력하기



1. Chapter02 프로젝트에 Lab02_01.java 파일을 만들고, 키보드로 값을 입력받기 위해 Scanner 클래스를 사용할 수 있도록 준비하기

```
import java.util.Scanner;  
  
Scanner s = new Scanner(System.in);
```

2. 택배를 받는 사람의 이름과 주소, 택배의 무게를 저장할 변수를 선언하기

```
String personName, ppersonAddr;  
int weight;
```

3. 택배를 받는 사람의 이름과 주소, 택배의 무게를 키보드로 입력받는 코드를 추가하기
 - `nextLine()`: 띄어쓰기나 Enter를 입력받는 메서드
 - 받는 사람과 주소는 문자열을 입력받고, 택배의 무게는 숫자를 입력받음

```
System.out.println("## 택배 보내기입니다. 다음을 각각 입력하세요 ##");  
System.out.print("받는 사람 : ");  
personName = s.nextLine();  
System.out.print("주소 : ");  
ppersonAddr = s.nextLine();  
System.out.print("무게(g) : ");  
weight = s.nextInt();
```



4. 택배를 받는 사람의 이름과 주소, 배송료를 출력하는데, 배송료는 그램당 5원을 곱한 결과를 출력하기

```
System.out.println("** 받는 사람 ==>" + personName);  
System.out.println("** 주소 ==>" + psonAddr);  
System.out.println("** 배송비 ==>" + weight*5 + "원");
```

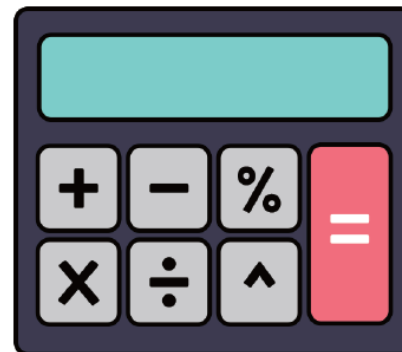

[LAB] 숫자를 입력하는 계산기 만들기



사용자가 키보드로 두 정수를 입력하면 더하기, 빼기, 곱하기, 나누기, 나머지 연산을 하는 계산기를 만들어봅시다

```
Console x
<terminated> Lab02_02 [Java Application] C:\Program Files\Java\jdk-11\bin\javaw.exe (2023. 6. 2. 오후 2:20:1)
정수1 ==>333
정수2 ==>100
333+100=433
333-100=233
333x100=33300
333/100=3.33
333%100=33
```

사용자가 입력



[LAB] 숫자를 입력하는 계산기 만들기



1. Chapter02 프로젝트에 Lab02_02.java 파일을 만들고, 키보드로 값을 입력받기 위해 Scanner 클래스를 사용할 수 있도록 준비하기

```
import java.util.Scanner;  
  
Scanner s = new Scanner(System.in);
```

2. 계산기 기능을 코딩하기
 - 정수형 변수 두 개와 계산 결과를 저장할 변수 다섯 개를 준비하는데, 이때 나누기 결과는 실수형으로 해야 함

```
int number1, number2;  
int resAdd;  
int resSub;  
int resMul;  
double resDiv;  
int resRem;
```

[LAB] 숫자를 입력하는 계산기 만들기



3. 두 변수에 키보드로 정수를 입력받는 코드를 추가

```
System.out.print("정수1 ==>");  
number1 = s.nextInt();  
System.out.print("정수2 ==>");  
number2 = s.nextInt();
```

4. 다섯 가지 연산을 코딩하고 각각의 결과를 변수에 저장하기

```
resAdd = number1 + number2;  
resSub = number1 - number2;  
resMul = number1 * number2;  
resDiv = (double) number1 / number2;  
resRem = number1 % number2;
```

5. 다섯 가지 연산을 출력하기

```
System.out.println(number1 + "+" + number2 + "=" + resAdd);  
System.out.println(number1 + "-" + number2 + "=" + resSub);  
System.out.println(number1 + "x" + number2 + "=" + resMul);  
System.out.println(number1 + "/" + number2 + "=" + resDiv);  
System.out.println(number1 + "%" + number2 + "=" + resRem);
```

[실전 예제]
거북이가 나오는 프로그램

[실전 예제] 거북이가 나오는 프로그램

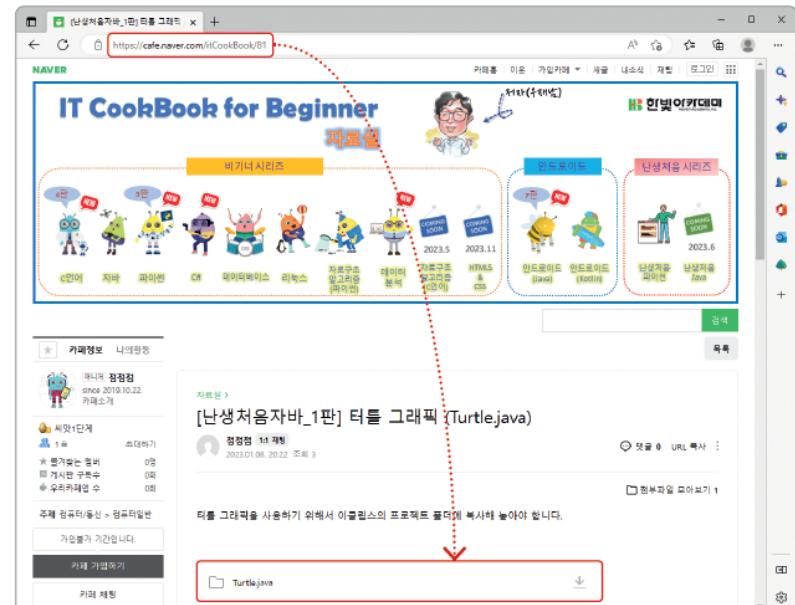
[문제]

터틀 그래픽이라는 그래픽 환경에서 자바를 재미있게 프로그래밍을 해봅시다.

[해결]

• 터틀 그래픽(Turtle Graphics) 준비하기

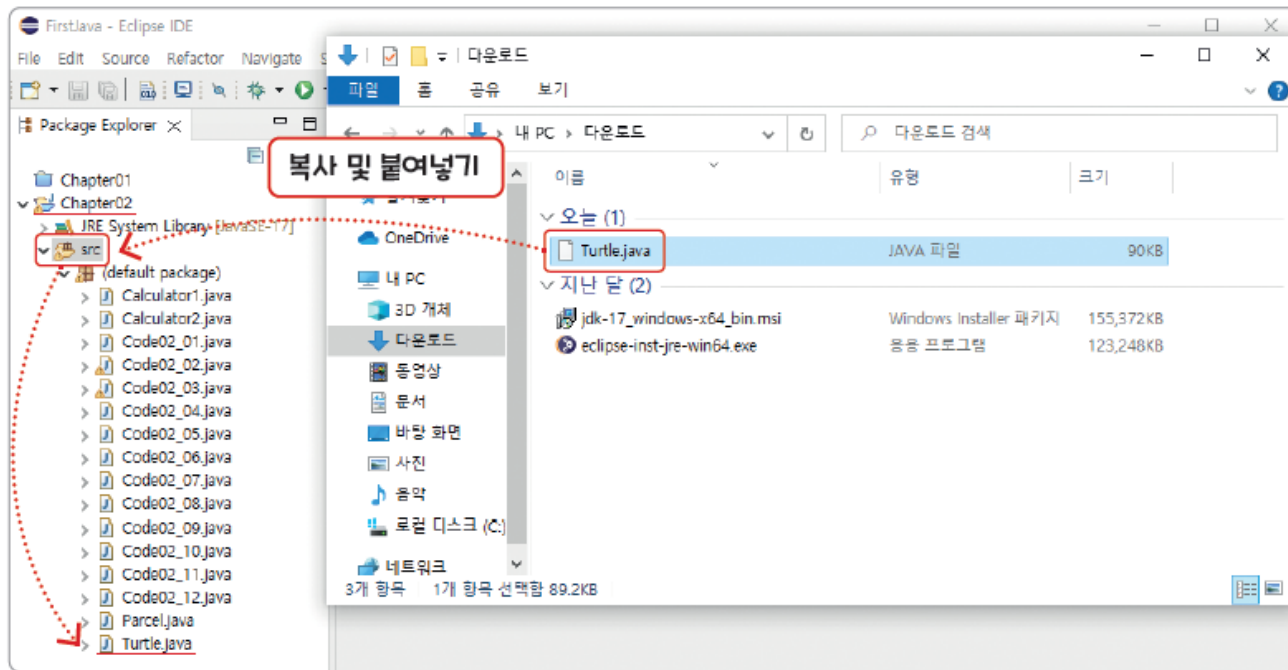
1. 자바에서 터틀 그래픽을 사용하려면 먼저 프로젝트에 Turtle.java 파일을 복사해야 함
 - IT CookBook for Beginner 카페(<https://cafe.naver.com/itCookBook>)에서 Turtle.java 파일을 다운로드하기
 - 또는 책의 자료실(<http://www.hanbit.co.kr/src/4662>)에서 예제소스를 다운받아 Turtle.java 파일 확인하기



[실전 예제] 거북이가 나오는 프로그램

[해결]

2. 다운로드한 Turtle.java를 복사하여 이클립스에서 프로젝트의 소스 폴더인 src 폴더에 붙여넣기

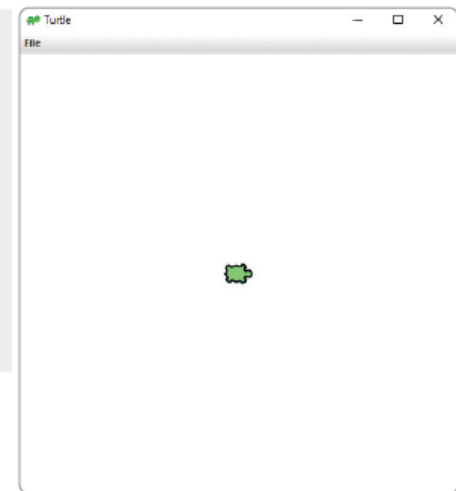


[실전 예제] 거북이가 나오는 프로그램

- 터틀 그래픽 시작하기

1. Chapter02 프로젝트에 Ex02_01.java 파일을 만들고, 터틀 그래픽이 나오도록 main() 메서드 안에 다음과 같이 한 행을 입력하고 [Run]-[Run]을 선택하여 실행해봅니다.
 - 가운데에 거북이가 있는 윈도우 창이 나타남

```
01  public class Ex02_01 {  
02      public static void main(String[] args) {  
03          Turtle turtle = new Turtle();  
04      }  
05  }  
06  }
```



[실전 예제] 거북이가 나오는 프로그램

- 거북이 이동 및 회전시키기

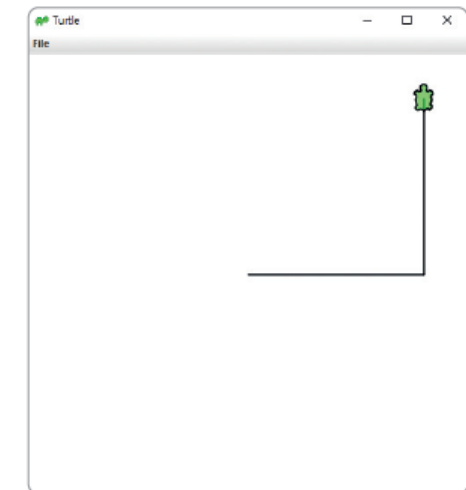
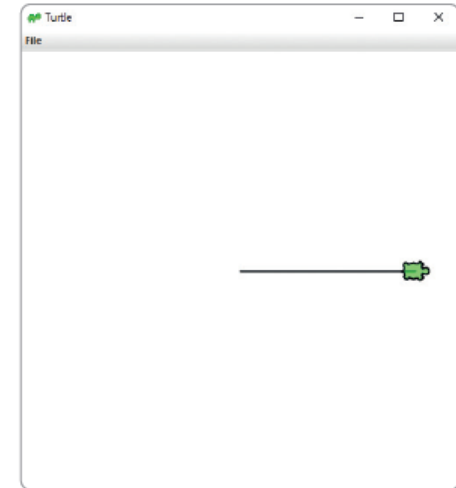
- 거북이 움직이기

```
turtle.forward(200);
```

앞으로 200만큼 이동

- 거북이를 위쪽으로 200만큼 움직이기

- 거북이를 왼쪽으로 90도 회전시킨 다음
앞으로 200만큼 이동



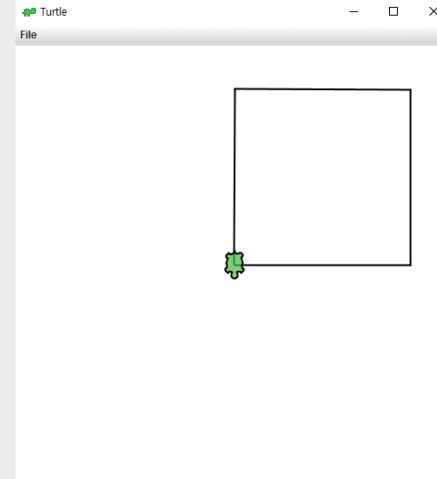
[실전 예제] 거북이가 나오는 프로그램

- 거북이 이동 및 회전시키기

3. 앞의 코드를 두 번 더 반복하여 거북이가 정사각형을 그리도록 만들기

Ex02_01.java

```
01 public class Ex02_01 {  
02     public static void main(String[] args) {  
03         Turtle turtle = new Turtle();  
04  
05         turtle.forward(200);  
06  
07         turtle.left(90);  
08         turtle.forward(200);  
09  
10         turtle.left(90);  
11         turtle.forward(200);  
12  
13         turtle.left(90);  
14         turtle.forward(200);  
15     }  
16 }
```



Thank you!