

자바 시작하기



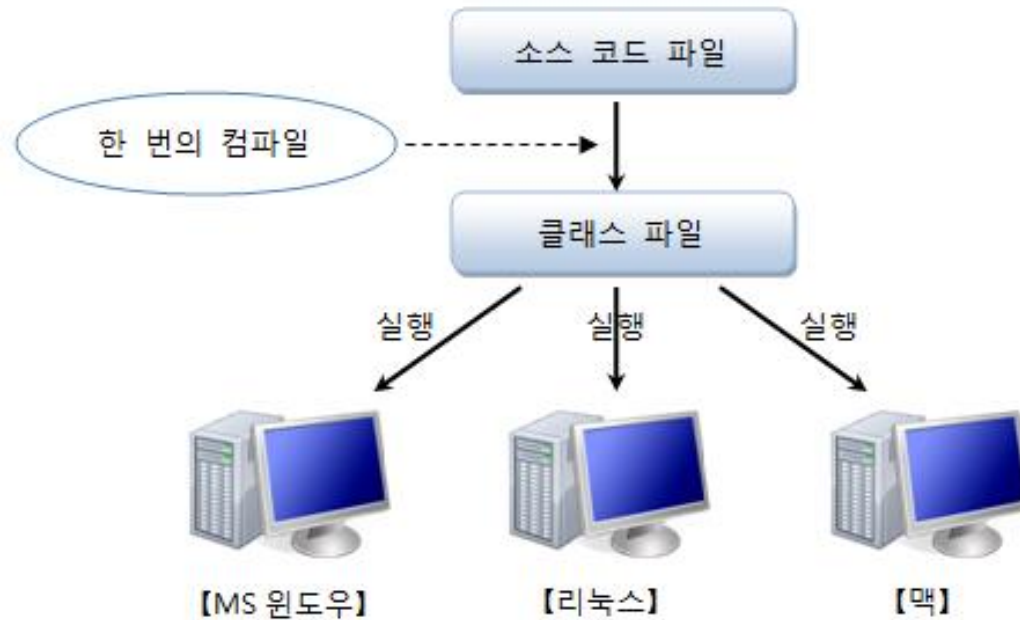
❖ 목차

- 자바의 특징
- 자바 프로그램 개발 순서
- 주석과 실행문
- 프린트 포맷
- 변수와 타입



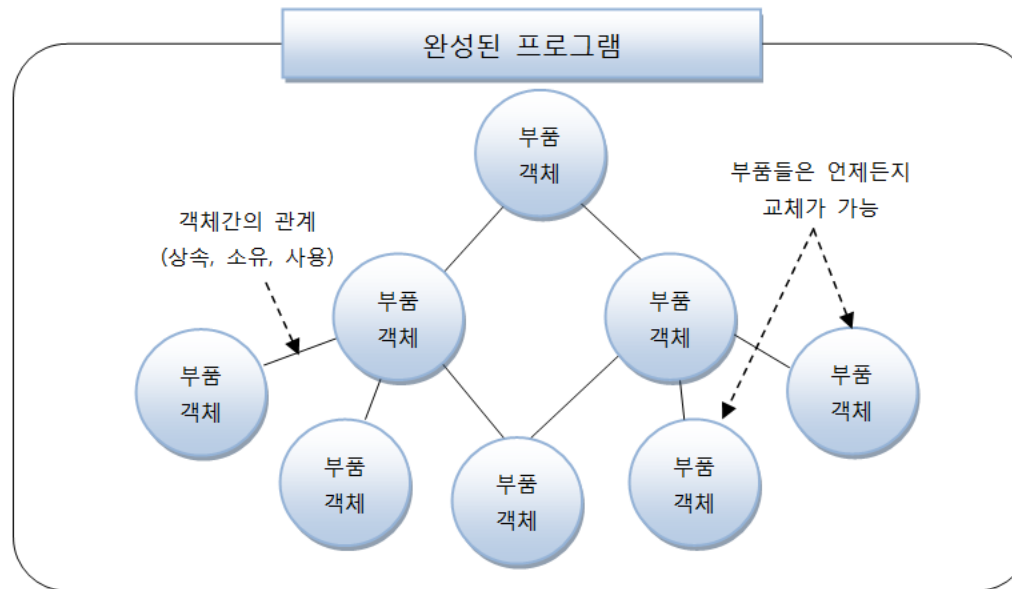
❖ 자바의 특징

- 이식성이 높은 언어



❖ 자바의 특징

- 객체 지향 언어 -> OOP(Object Oriented Programming)란?
 - 부품 객체를 먼저 만들고, 이것들을 조합해 전체 프로그램을 완성하는 기법
- 자바는 처음부터 OOP 개발용 언어로 설계
 - 캡슐화, 상속, 다형성 기능 완벽하게 지원



❖ 자바의 특징

■ 함수적 스타일 코딩 지원

- 함수적 스타일 코딩 방식인 람다식(Lambda Expressions) 지원
 - 코드 간결하게 작성 가능
 - 컬렉션 요소를 필터링, 매핑, 그룹핑, 집계 처리시 주로 사용

■ 메모리를 자동으로 관리

- 사용하지 않는 객체를 자동으로 메모리에서 제거
- 핵심 기능 코드에 집중할 수 있도록 하는 기능

■ 다양한 애플리케이션 개발 가능



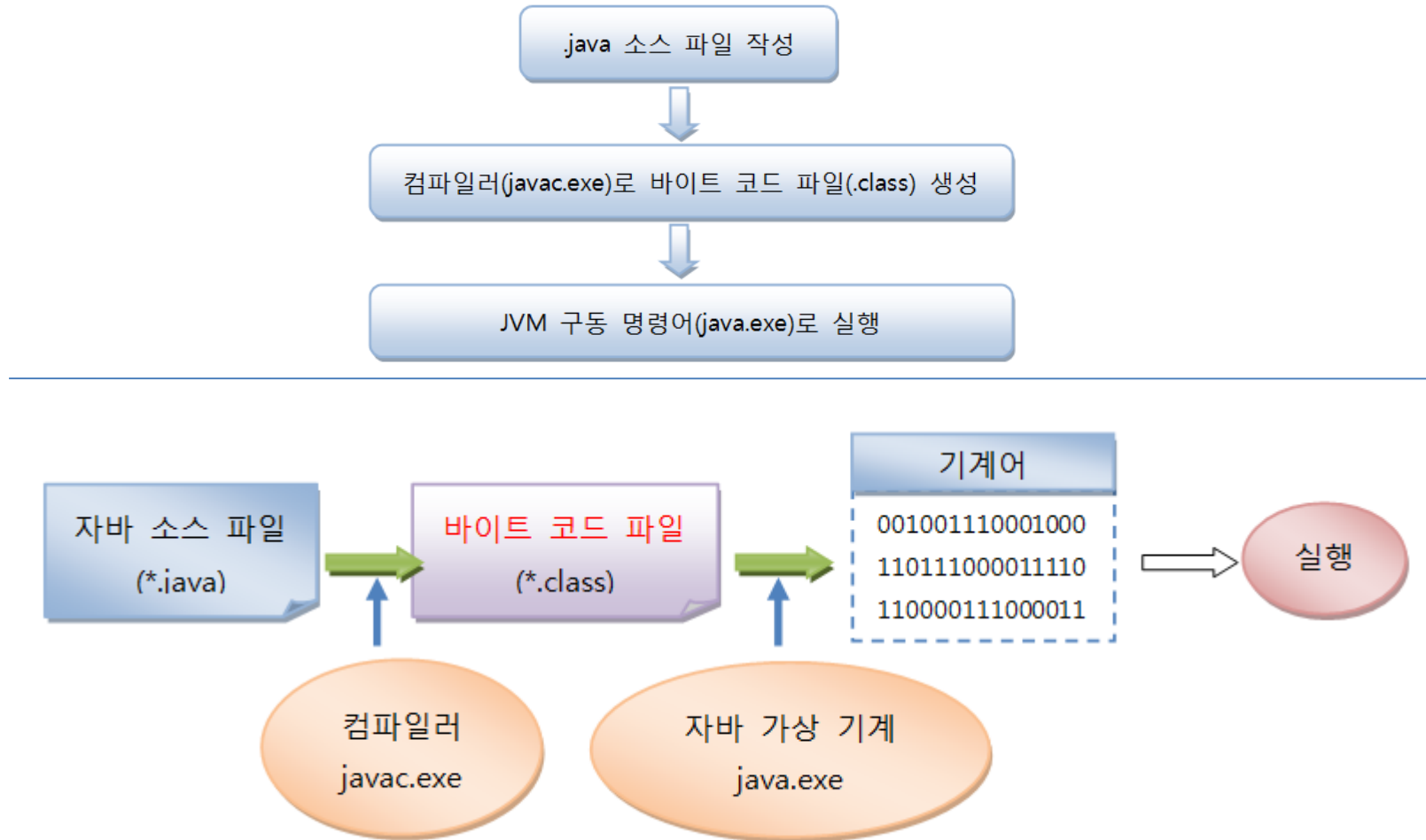
❖ 자바의 특징

- 멀티 스레드(Multi-Thread) 쉽게 구현 가능
 - 동시에 여러 가지 작업을 할 경우
 - 대용량 작업을 빨리 처리할 경우
- 동적 로딩(Dynamic Loading) 지원
 - 미리 객체를 만들어 놓지 않고 필요한 시점에 동적 로딩해 객체 생성
 - 유지 보수 시 특정 객체만 쉽게 수정 및 교체해 사용
- 막강한 오픈 소스 라이브러리 풍부



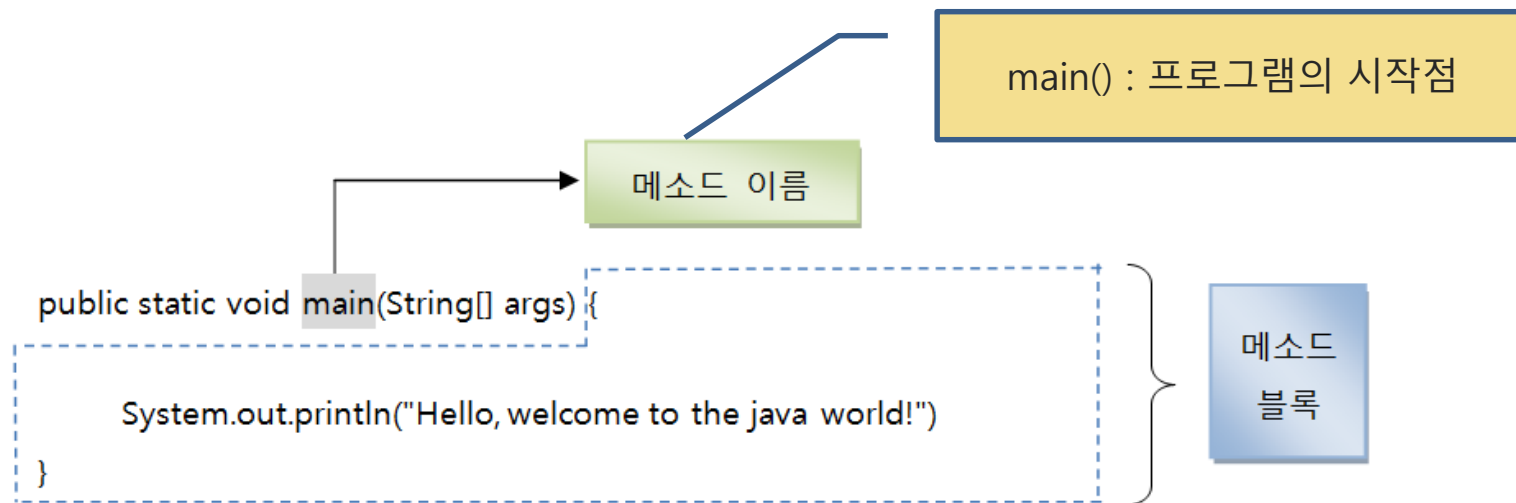
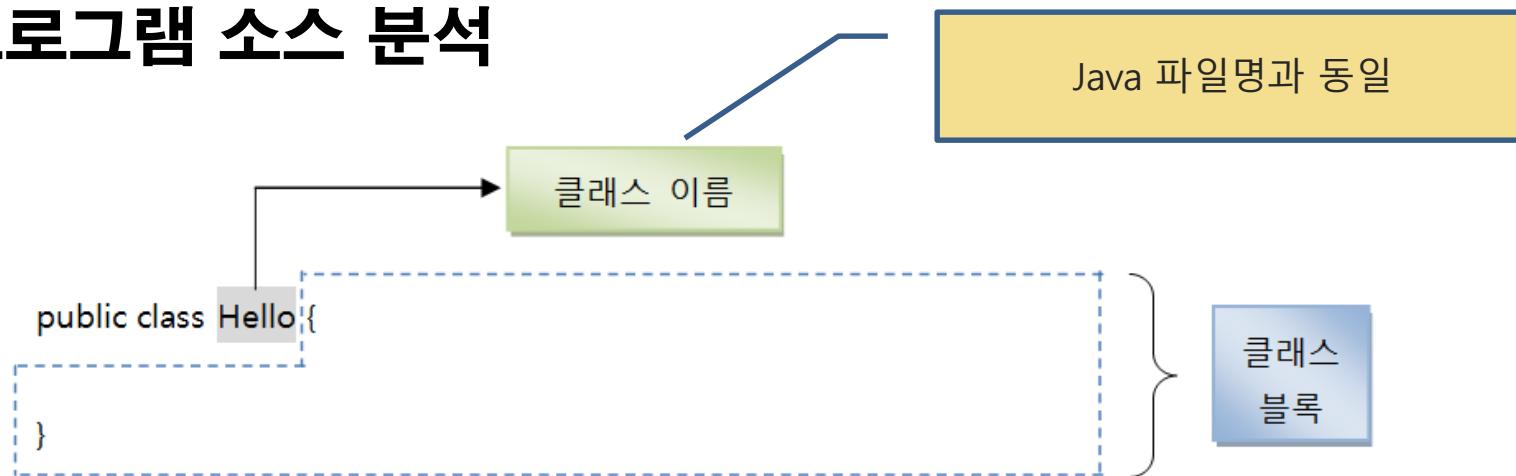
자바 프로그램 개발 순서

❖ 소스 작성에서부터 실행까지



자바 프로그램 개발 순서

❖ 프로그램 소스 분석



실습 예제

❖ 이클립스를 이용하여 다음 문장을 출력하는 자바 프로그램을 작성하시오

Java Project : week2 (File > New > Java Project)

클래스명 : JavaWorld

출력결과 :

Java는 객체 지향 언어이다.

Java는 이식성이 높은 언어이다.

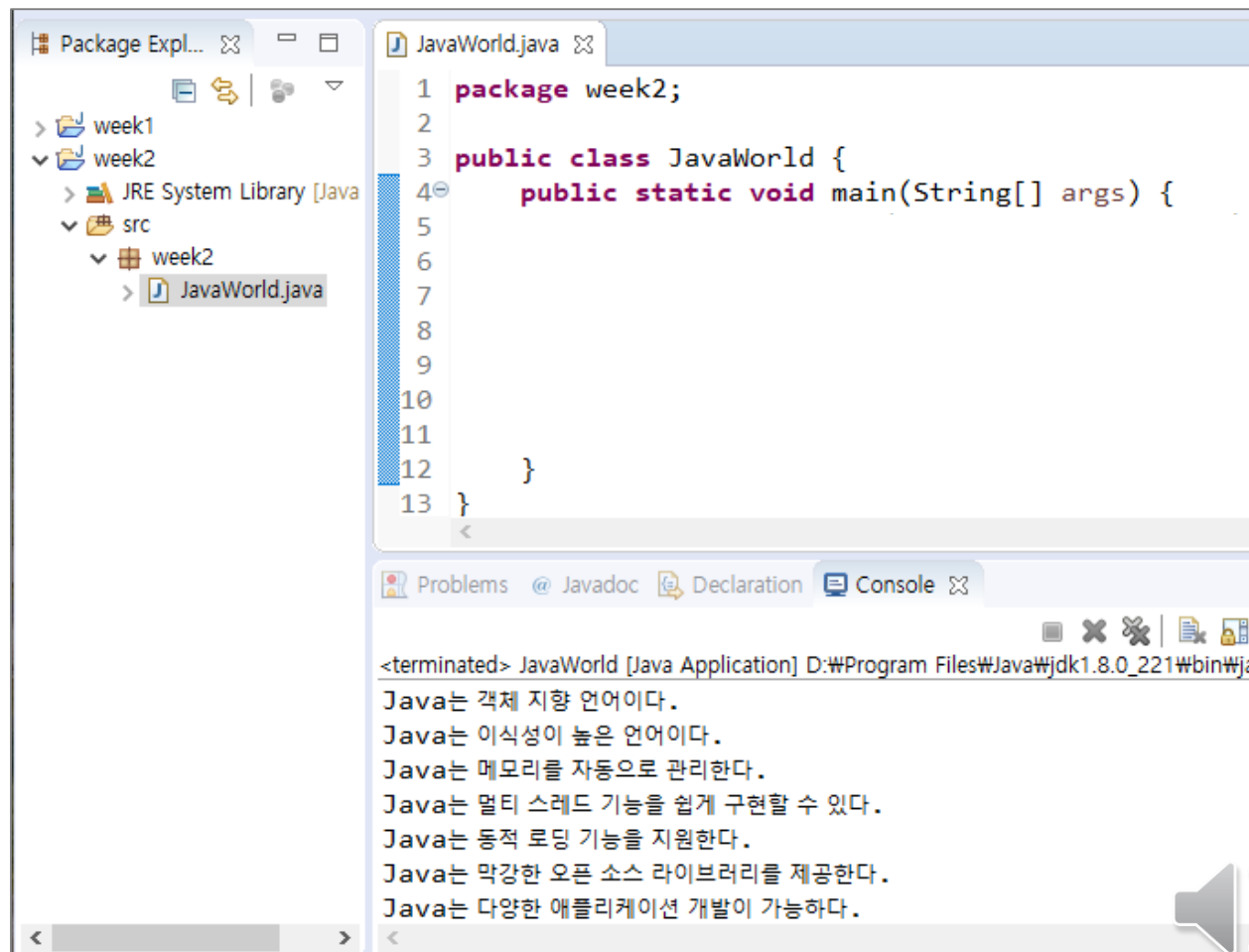
Java는 메모리를 자동으로 관리한다.

Java는 멀티 스레드 기능을 쉽게 구현할 수 있다.

Java는 동적 로딩 기능을 지원한다.

Java는 막강한 오픈 소스 라이브러리를 제공한다.

Java는 다양한 애플리케이션 개발이 가능하다.



The screenshot shows the Eclipse IDE interface. On the left, the Package Explorer displays a project structure with folders 'week1' and 'week2'. Inside 'week2', there is a 'src' folder containing 'JavaWorld.java'. The main editor window shows the code for 'JavaWorld.java':

```
1 package week2;
2
3 public class JavaWorld {
4     public static void main(String[] args) {
5
6
7
8
9
10
11
12     }
13 }
```

At the bottom, the Console window shows the output of the program:

```
<terminated> JavaWorld [Java Application] D:\Program Files\Java\jdk1.8.0_221\bin\java.exe
Java는 객체 지향 언어이다.
Java는 이식성이 높은 언어이다.
Java는 메모리를 자동으로 관리한다.
Java는 멀티 스레드 기능을 쉽게 구현할 수 있다.
Java는 동적 로딩 기능을 지원한다.
Java는 막강한 오픈 소스 라이브러리를 제공한다.
Java는 다양한 애플리케이션 개발이 가능하다.
```

실습 예제

❖ 이클립스를 이용하여 다음 문장을 출력하는 자바 프로그램을 작성하시오

Java Project : week2

클래스명 : JavaWorld

출력결과 :

Java는 객체 지향 언어이다

Java는 이식성이 높은 언어이다

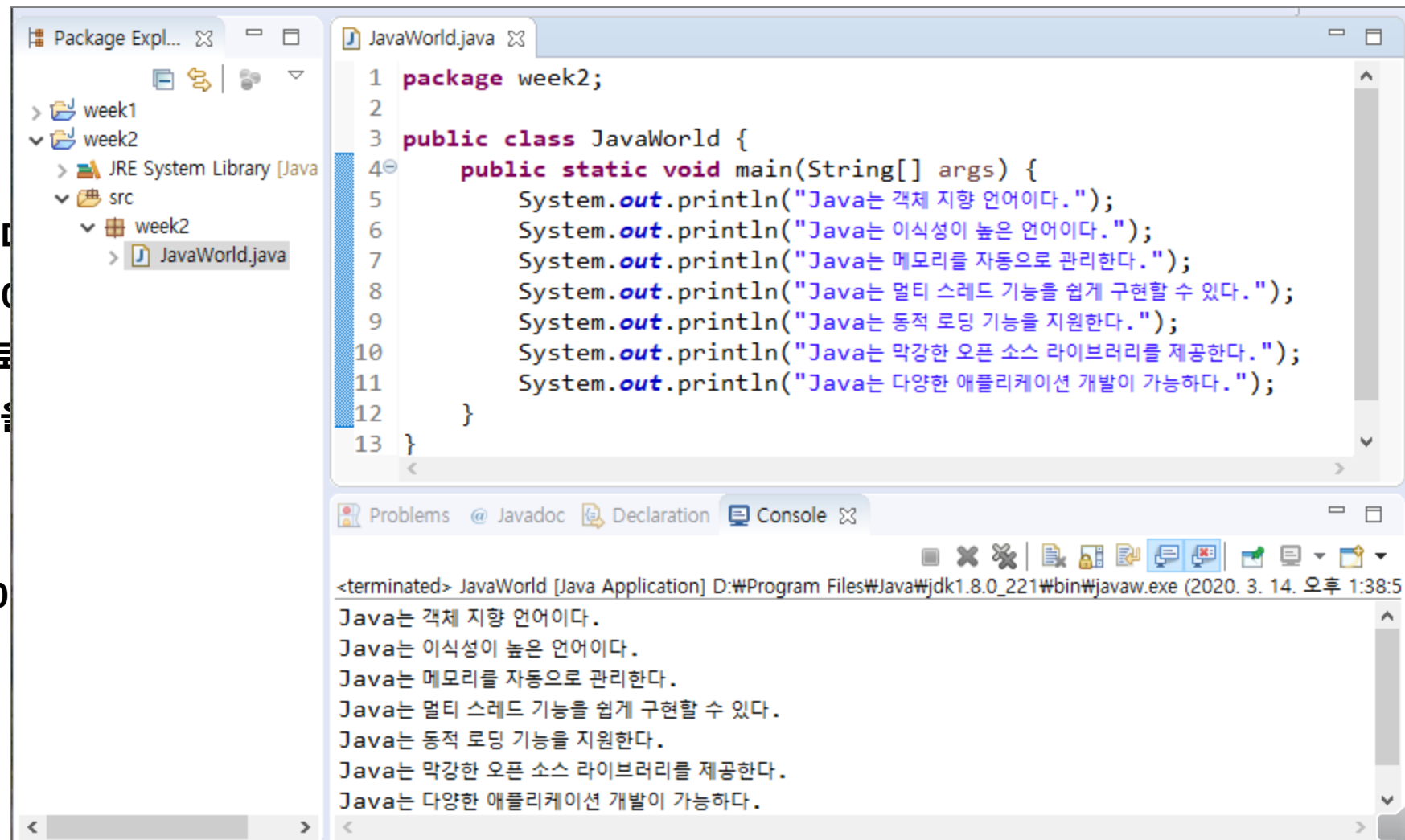
Java는 메모리를 자동으로 관리한다

Java는 멀티 스레드 기능을 쉽게 구현할 수 있다

Java는 동적 로딩 기능을 지원한다

Java는 막강한 오픈 소스 라이브러리를 제공한다

Java는 다양한 애플리케이션 개발이 가능하다



```
1 package week2;
2
3 public class JavaWorld {
4     public static void main(String[] args) {
5         System.out.println("Java는 객체 지향 언어이다.");
6         System.out.println("Java는 이식성이 높은 언어이다.");
7         System.out.println("Java는 메모리를 자동으로 관리한다.");
8         System.out.println("Java는 멀티 스레드 기능을 쉽게 구현할 수 있다.");
9         System.out.println("Java는 동적 로딩 기능을 지원한다.");
10        System.out.println("Java는 막강한 오픈 소스 라이브러리를 제공한다.");
11        System.out.println("Java는 다양한 애플리케이션 개발이 가능하다.");
12    }
13 }
```

<terminated> JavaWorld [Java Application] D:\Program Files\Java\jdk1.8.0_221\bin\javaw.exe (2020. 3. 14. 오후 1:38:5)

Java는 객체 지향 언어이다.
Java는 이식성이 높은 언어이다.
Java는 메모리를 자동으로 관리한다.
Java는 멀티 스레드 기능을 쉽게 구현할 수 있다.
Java는 동적 로딩 기능을 지원한다.
Java는 막강한 오픈 소스 라이브러리를 제공한다.
Java는 다양한 애플리케이션 개발이 가능하다.

❖ 주석 사용하기

- 프로그램 실행과는 상관없이 코드에 설명 붙인 것
- 컴파일 과정에서 주석은 무시되고 실행문만 바이트 코드로 번역
- 코드에서 사용하는 주석문의 종류

주석 기호	설명
//	//부터 라인 끝까지 주석으로 처리한다. (행 주석)
/* ~ */	/*와 */ 사이에 있는 모든 범위를 주석으로 처리한다. (범위 주석)



❖ 실행문과 세미콜론(;)

■ 실행문

- 변수 선언, 값 저장, 메소드 호출에 해당하는 코드
- 실행문 끝에는 반드시 세미콜론(;)을 붙여 실행문의 끝 표시

```
int x = 1;           //변수 x를 선언하고 1을 저장
int y = 2;           //변수 y를 선언하고 2를 저장
int result = x + y;   //변수 result를 선언하고 변수 x와 y를 더한 값을 저장
System.out.println(result); //콘솔에 출력하는 메소드 호출
```

```
int x = 1; int y = 2;
int result =
    x + y;
```





```
package week2;
```

```
//Print1의 P가 대문자로 작성되어야 한다
```

```
/* 파일명과 프로그램 내 클래스명은 반드시 동일해야 한다*/
```

```
/* 이클립스를 사용하면 클래스 파일 생성시 입력한 파일명이 자동으로 클래스명에 적용된다*/
```

```
public class Print1{
```

```
    // main 메소드는 반드시 있어야 함
```

```
    // 자바 프로그램을 시작하는 진입점
```

```
    // main 메소드는 문자열 매개변수를 가짐
```

```
    /* String의 S는 대문자로 작성되어야 함 */
```

```
    public static void main(String[] args) {
```

```
        /* 아래 실행문은 'Hello Java!'를 출력하는 문장임 */
```

```
        System.out.print("Hello Java!");
```

```
        //System의 S는 대문자로 작성되어야 함
```

```
        System.out.print("Java ");
```

```
        //각 실행문의 마지막에는 반드시 세미콜론(;)을 붙여주어야 함
```

```
        System.out.print("World!");
```

```
    }
```

```
}
```





```
package week2;
```

```
public class Print2{
```

```
    // main() 함수 자동 생성 단축키
```

```
    // "main" + ctrl+spacebar => main method 선택
```

```
    // "sysout" + ctrl+spacebar => System.out.println() 자동생성 단축키
```

```
    public static void main(String[] args) {
```

```
        public static void main(String[] args) {
```

```
            /*System.out.println( " 객체");
```

```
            System.out.println( " 지향");*/
```

```
            System.out.*자바*/println( " 프로그래밍");
```

```
            System.out.println( " 프로/*자바*/그래밍");
```

```
        }
```

```
    }
```



```
package week2;
```

```
public class Print2{
```

```
    // main() 함수 자동 생성 단축키
```

```
    // "main" + ctrl+spacebar => main method 선택
```

```
    // "sysout" + ctrl+spacebar => System.out.println() 자동생성 단축키
```

```
    public static void main(String[] args) {
```

```
        public static void main(String[] args) {
```

```
            /*System.out.println( " 객체");
```

```
            System.out.println( " 지향");*/
```

```
            System.out.*자바*/println( " 프로그래밍");
```

```
            System.out.println( " 프로/*자바*/그래밍");
```

```
        }
```

```
    }
```



```
public class Print3{  
    public static void main(String[] args) {  
        System.out.println(17 + 35);  
        System.out.println("17 + 35");  
  
        //자바에서는 문자열을 표시할 때는 반드시 ""를 사용하여야 한다  
        System.out.println("17 + 35 = " + 17+35);  
        System.out.println("17 + 35 = " + (17+35));  
    }  
}
```



프린트 포맷

//Print4.java

```
public class Print4 {  
    public static void main(String[] args) {
```

```
        /* 문자열 출력 포맷(printf) */
```

```
        /*
```

```
        * % : 프린트 포맷 기호
```

```
        * s : 문자열(String)
```

```
        * Wn : 줄바꿈
```

```
        */
```

```
        //프린트 포맷에 정의된 변수의 갯수보다 많은 값을 나열하는 것은 가능
```

```
        System.out.printf("이름 : %s, %sWn", "자바", "이클립스", "hello");
```

```
        //프린트 포맷에 정의된 변수의 갯수보다 적은 값을 나열하는 것은 에러발생
```

```
        //System.out.printf("이름 : %s, %s, %s", "자바", "이클립스");
```

```
        System.out.printf("이름 : %s, %s, %s", "자바", "이클립스", "hello");
```

```
    }
```

```
}
```



프린트 포맷

//Print5.java

```
public class Print5 {  
    public static void main(String[] args) {  
        System.out.println(23 + 18);  
        System.out.println("23 + 18 = " + (23+18));  
  
        /* 숫자 출력 포맷 */  
        //% : 프린트 포맷 기호  
        //d : decimal(숫자)  
        System.out.printf("%d\n", (23+18));  
        System.out.printf("숫자 연산 : 23 + 18 = %d", (23+18));  
  
        System.out.println("\n사칙연산(53, 17)");  
        System.out.printf("더하기 : 53 + 17 = %d\n", (53+17));  
        System.out.printf("빼기 : 53 - 17 = %d\n", (53-17));  
        System.out.printf("곱하기 : 53 * 17 = %d\n", (53*17));  
        System.out.printf("나누기 : 53 / 17 = %d\n", (53/17));  
    }  
}
```



2장. 변수와 타입



1절. 변수

- 변수란?
- 변수의 선언
- 변수의 사용
- 변수의 사용 범위

2절. 데이터 타입

- 기본 타입
- 정수 타입
- 실수 타입
- 논리 타입



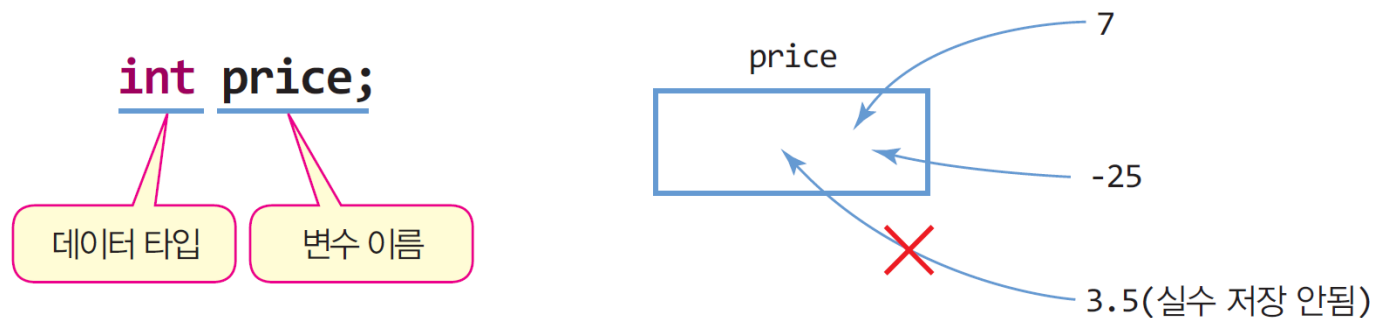
1절. 변수

❖ 변수

- 프로그램 실행 중에 값을 임시 저장하기 위한 메모리 공간
 - 변수 값은 프로그램 수행 중 변경될 수 있음
- 데이터 타입에서 정한 크기의 메모리 할당

❖ 변수 선언

- 변수의 타입 다음에 변수 이름을 적어 변수를 선언



```
int age ;
```

```
double value ;
```



❖ 변수 이름을 위한 명명 규칙(naming convention)

작성 규칙	예
첫번째 글자는 문자이거나 '\$', '_' 여야 하고 숫자로 시작할 수 없다. (필수)	가능: price, \$price, _companyName 안됨: 1v, @speed, \$#value
영어 대소문자가 구분된다. (필수)	firstname 과 firstName 은 다른 변수
첫문자는 영어 소문자로 시작하되, 다른 단어가 붙을 경우 첫자를 대문자로 한다. (관례)	maxSpeed, firstName, carBodyColor
문자 수(길이)의 제한은 없다.	
자바 예약어는 사용할 수 없다. (필수)	책 참조



1절. 변수

❖ 변수의 사용

■ 변수값 저장

```
int score;    //변수 선언  
score = 90;   //값저장
```

초기값은 변수를 선언함과 동시에 줄 수도 있다.

```
int score = 90;
```



❖ 변수의 사용

■ 변수값 읽기

- 변수는 초기화가 되어야 읽기 가능
- 잘못된 코딩의 예

```
int value;           //변수 value 선언 (초기화 안됨)  
int result = value + 10; //변수 value 값을 읽고 10 을 더한 결과값을 변수 result 에 저장
```

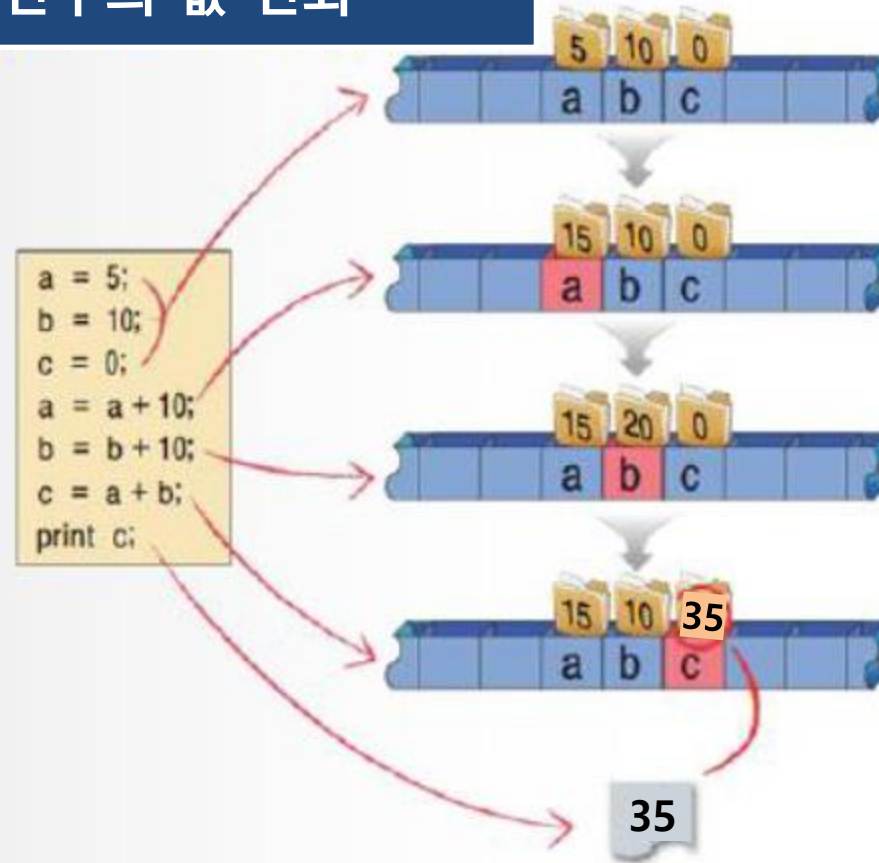
- 맞게 고친 후의 코드

```
int value = 30;       //변수 value 가 30 으로 초기화 됨  
int result = value + 10; //변수 value 값을 읽고 10 을 더한 결과값(40)을 변수 result 에 저장
```



1절. 변수

메모리에 저장된 변수의 값 변화



메모리에 저장된 변수값은 프로그램의 진행에 따라 변화된다



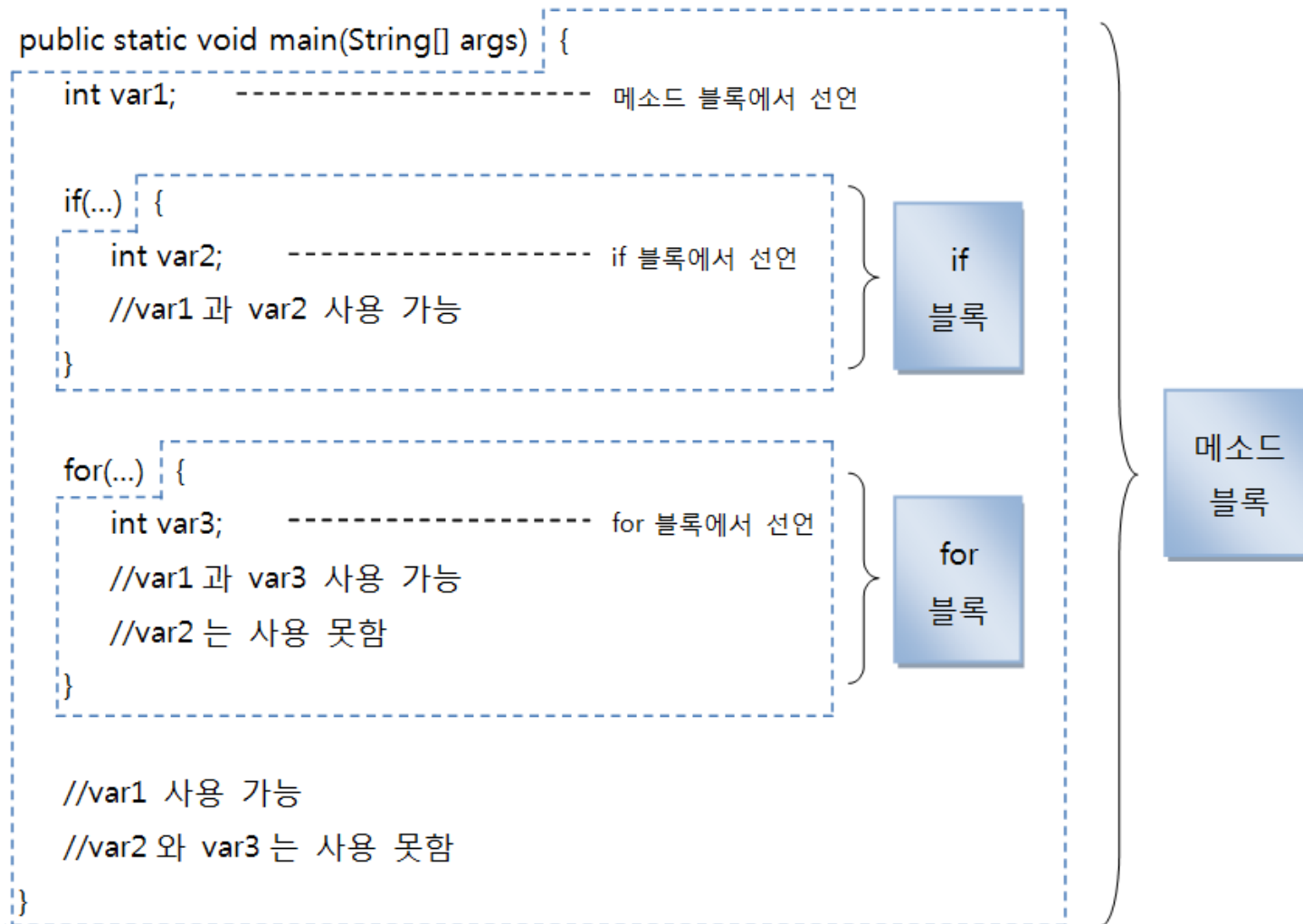
❖ 리터럴(literal)

- 소스 코드 내에서 직접 입력된 변수의 초기값
- 소스 코드 내에서 익숙해지는 것이 point
- 종류
 - 정수 리터럴 : 0, 75, -100
 - 실수 리터럴 : 0.25, -3.14
 - 문자 리터럴 : 'a' , '한'
 - 문자열 리터럴 : "Java" , "컴퓨터정보과"
 - 논리 리터럴 : true, false



❖ 변수의 사용 범위

- 변수는 중괄호 블록 {} 내에서 선언되고 사용



2절. 데이터 타입

❖ 기본(primitive) 타입

- 정수, 실수, 문자, 논리 리터럴을 직접 저장하는 타입
- 메모리의 최소 기억단위인 bit가 모여 byte 형성

1 byte = 8 bit

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

값의 종류	기본 타입	메모리 사용 크기		저장되는 값의 범위
정수	byte	1 byte	8 bit	$-2^7 \sim 2^7 - 1$ (-128 ~ 127)
	char	2 byte	16 bit	$0 \sim 2^{16} - 1$ (유니코드: \u0000 ~ \uFFFF, 0 ~ 65535)
	short	2 byte	16 bit	$-2^{15} \sim 2^{15} - 1$ (-32,768 ~ 32,767)
	int	4 byte	32 bit	$-2^{31} \sim 2^{31} - 1$ (-2,147,483,648 ~ 2,147,483,647)
	long	8 byte	64 bit	$-2^{63} \sim 2^{63} - 1$
실수	float	4 byte	32 bit	$(+/-)1.4E-45 \sim (+/-)3.4E38$
	double	8 byte	64 bit	$(+/-)4.9E-324 \sim (+/-)1.7E308$
논리	boolean	1 byte	8 bit	true, false



2절. 데이터 타입

정수 값을 저장하는데 사용된다. 주로 사용하는 것은 int와 long이며, byte는 이진데이터를 다루는데 사용된다.

❖ 기본(primitive) 타입

값의 종류	기본 타입	메모리 사용 크기		저장되는 값의 범위
정수	byte	1 byte	8 bit	$2^7 \sim 2^7 - 1$ (-128 ~ 127)
	char	2 byte	16 bit	$0 \sim 2^{16} - 1$ (유니코드: \u0000 ~ \uFFFF, 0 ~ 65535)
	short	2 byte	16 bit	$-2^{15} \sim 2^{15} - 1$ (-32,768 ~ 32,767)
	int	4 byte	32 bit	$-2^{31} \sim 2^{31} - 1$ (-2,147,483,648 ~ 2,147,483,647)
	long	8 byte	64 bit	$-2^{63} \sim 2^{63} - 1$
실수	float	4 byte	32 bit	$(+/-)1.4E-45 \sim (+/-)3.4E38$
	double	8 byte	64 bit	$(+/-)4.9E-324 \sim (+/-)1.7E308$
논리	boolean	1 byte	8 bit	true, false



2절. 데이터 타입

❖ 기본(primitive) 타입

값의 종류	기본 타입	메모리 사용 크기		저장되는 값의 범위
정수	byte	1 byte	8 bit	$2^7 \sim 2^7 - 1$ (-128 ~ 127)
	char	2 byte	16 bit	$0 \sim 2^{16} - 1$ (유니코드: \u0000 ~ \uFFFF, 0 ~ 65535)
	short	2 byte	16 bit	$-2^{15} \sim 2^{15} - 1$ (-32,768 ~ 32,767)
	int	4 byte	32 bit	$-2^{31} \sim 2^{31} - 1$ (-2,147,483,648 ~ 2,147,483,647)
	long	8 byte	64 bit	$-2^{63} \sim 2^{63} - 1$
실수	float	4 byte	32 bit	$(+/-)1.4E-45 \sim (+/-)3.4E38$
	double	8 byte	64 bit	$(+/-)4.9E-324 \sim (+/-)1.7E308$
논리	boolean	1 byte	8 bit	true, false

실수 값을 저장하는데 사용된다.
float과 **double**이 있다.



2절. 데이터 타입

❖ 기본(primitive) 타입

값의 종류	기본 타입	메모리 사용 크기		저장되는 값의 범위
정수	byte	1 byte	8 bit	$2^7 \sim 2^7 - 1$ (-128 ~ 127)
	char	2 byte	16 bit	$0 \sim 2^{16} - 1$ (유니코드: \u0000 ~ \uFFFF, 0 ~ 65535)
	short	2 byte	16 bit	$-2^{15} \sim 2^{15} - 1$ (-32,768 ~ 32,767)
	int	4 byte	32 bit	$-2^{31} \sim 2^{31} - 1$ (-2,147,483,648 ~ 2,147,483,647)
	long	8 byte	64 bit	$-2^{63} \sim 2^{63} - 1$
실수	float	4 byte	32 bit	(+/-)1.4E-45 ~ (+/-)3.4E38
	double	8 byte	64 bit	(+/-)4.9E-324 ~ (+/-)1.7E308
논리	boolean	1 byte	8 bit	true, false

true와 false 중 하나를 값으로 가지며
조건식과 논리적 계산에 사용된다



2절. 데이터 타입

❖ 기본(primitive) 타입

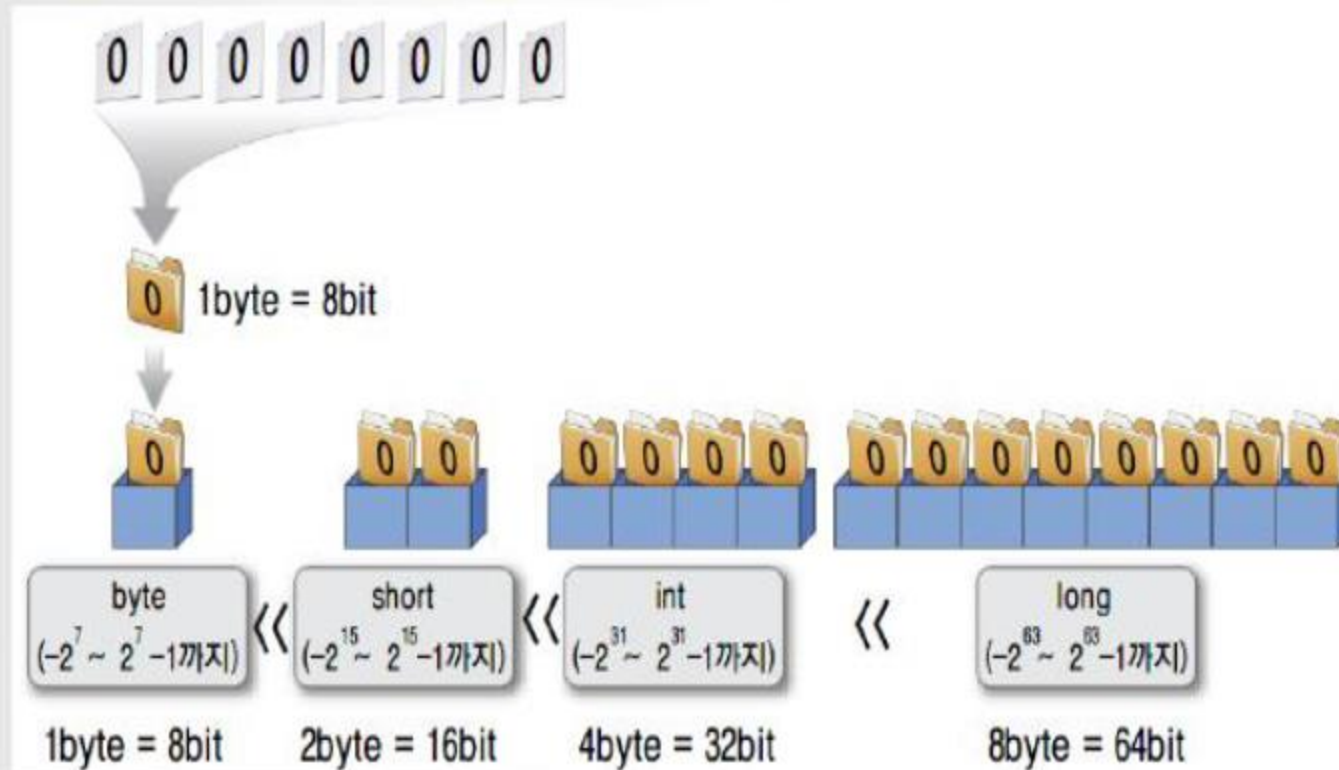
값의 종류	기본 타입	메모리 사용 크기		저장되는 값의 범위
정수	byte	1 byte	8 bit	$2^7 \sim 2^7 - 1$ (-128 ~ 127)
	char	2 byte	16 bit	$0 \sim 2^{16} - 1$ (유니코드: \u0000 ~ \uFFFF, 0 ~ 65535)
	short	2 byte	16 bit	$-2^{15} \sim 2^{15} - 1$ (-32,768 ~ 32,767)
	int	4 byte	32 bit	$-2^{31} \sim 2^{31} - 1$ (-2,147,483,648 ~ 2,147,483,647)
	long	8 byte	64 bit	$-2^{63} \sim 2^{63} - 1$
실수	float	4 byte	32 bit	(+/-)1.4E-45 ~ (+/-)3.4E38
	double	8 byte	64 bit	(+/-)4.9E-324 ~ (+/-)1.7E308
논리	boolean	1 byte	8 bit	true, false

문자를 저장하는데 사용되며
변수당 하나의 문자만을 저장할 수 있다



2절. 데이터 타입

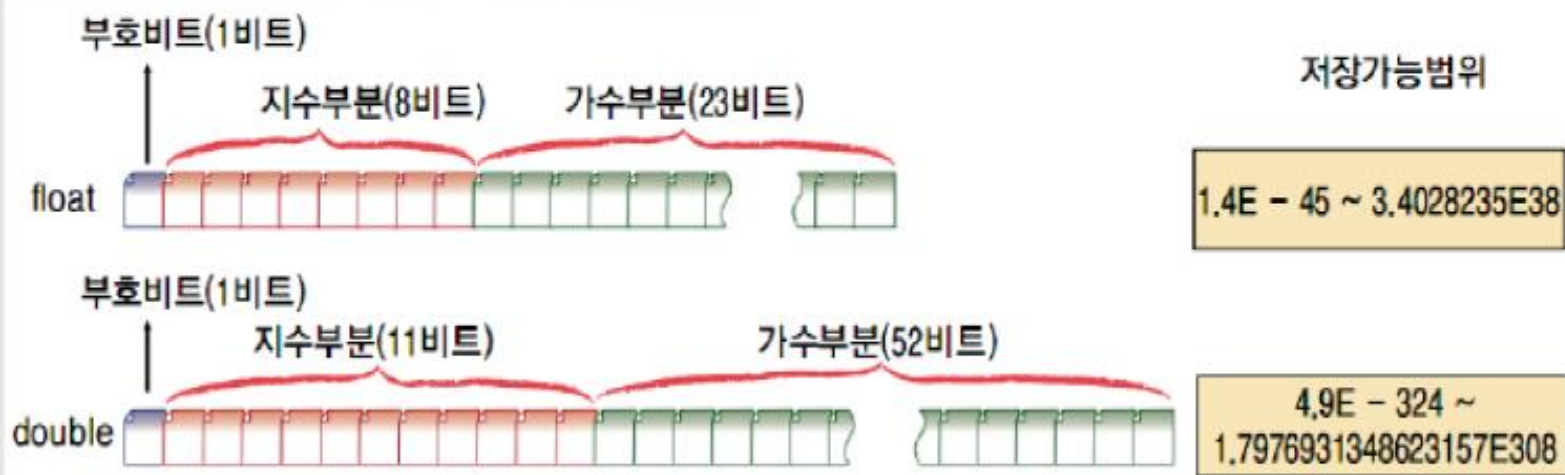
- 정수형 : 4가지의 자료형이 제공



2절. 데이터 타입

● 실수형

- 부호와 지수(exponential)부분, 가수(mantissa)부분으로 구성
- 저장할 수 있는 크기에 따라 float형과 double형으로 구분
- 묵시적(default) 데이터형은 double형



2절. 데이터 타입

❖ 문자형 – char

- 자바는 모든 문자를 유니코드로 처리
- 유니코드
 - 세계 각국의 문자들을 코드값으로 매핑
 - 국제 표준 규약
 - 하나의 문자에 대해 하나의 코드값
 - 2byte 크기의 char 타입 사용
 - 유니코드에는 음수가 없다

10진수	ASCII	10진수	ASCII	10진수	ASCII	10진수	ASCII
0	NULL	32	SP	64	@	96	.
1	SOH	33	!	65	A	97	a
2	STX	34	"	66	B	98	b
3	ETX	35	#	67	C	99	c
4	EOT	36	\$	68	D	100	D
5	ENQ	37	%	69	E	101	e
6	ACK	38	&	70	F	102	f
7	BEL	39	'	71	G	103	g
8	BS	40	(72	H	104	h
9	HT	41)	73	I	105	i
10	LF	42	*	74	J	106	j
11	VT	43	+	75	K	107	k
12	FF	44	,	76	L	108	l
13	CR	45	-	77	M	109	m
14	SO	46	,	78	N	110	n
15	SI	47	/	79	O	111	o
16	DLE	48	0	80	P	112	p
17	DC1	49	1	81	Q	113	q
18	SC2	50	2	82	R	114	r
19	SC3	51	3	83	S	115	s
20	SC4	52	4	84	T	116	t



2절. 데이터 타입

❖ 문자형 – char

- 자바는 모든 문자를 유니코드로 처리
- 유니코드
 - 세계 각국의 문자들을 코드값으로 매핑
 - 국제 표준 규약
 - 하나의 문자에 대해 하나의 코드값
 - 2byte 크기의 char 타입 사용
 - 유니코드에는 음수가 없다

10진수	ASCII	10진수	ASCII	10진수	ASCII	10진수	ASCII
20	SC4	52	4	84	T	116	t
21	NAK	53	5	85	U	117	u
22	SYN	54	6	86	V	118	v
23	ETB	55	7	87	W	119	w
24	CAN	56	8	88	X	120	x
25	EM	57	9	89	Y	121	y
26	SUB	58	:	90	Z	122	z
27	ESC	59	;	91	[123	{
28	FS	60	<	92	\	124	
29	GS	61	=	93]	125	}
30	RS	62	>	94	^	126	~
31	US	63	?	95	_	127	DEL



```
DataType1.java
1 package week3;
2
3 public class DataType1 {
4     public static void main(String[] args) {
5         int    iData    = 65;
6         byte   bData    = 65;
7         short  sData    = 65;
8         long   lData    = 65;
9         float  fData    = 65.12f;
10        double dData    = 65.12d;
11        char   cData    = 65;
12        boolean boolData = true;
13
14        System.out.println("iData = " + iData);
15        System.out.println("bData = " + bData);
16        System.out.println("sData = " + sData);
17        System.out.println("lData = " + lData);
18        System.out.println("fData = " + fData);
19        System.out.println("dData = " + dData);
20        System.out.println("cData = " + cData);
21        System.out.println("boolData = " + boolData);
22    }
23 }
```



```
DataType1.java
1 package week3;
2
3 public class DataType1 {
4     public static void main(String[] args) {
5         int iData = 65;
6         byte bData = 65;
7         short sData = 65;
8         long lData = 65;
9         float fData = 65.12f;
10        double dData = 65.12d;
11        char cData = 65;
12        boolean boolData = true;
13
14        System.out.println("iData = " + iData);
15        System.out.println("bData = " + bData);
16        System.out.println("sData = " + sData);
17        System.out.println("lData = " + lData);
18        System.out.println("fData = " + fData);
19        System.out.println("dData = " + dData);
20        System.out.println("cData = " + cData);
21        System.out.println("boolData = " + boolData);
22    }
23 }
```

```
Console
<terminated> DataType1 [Java Application] C:\Program Files\Java
iData = 65
bData = 65
sData = 65
lData = 65
fData = 65.12
dData = 65.12
cData = A
boolData = true
```



```
DataType2.java
1 package week3;
2
3 public class DataType2 {
4     public static void main(String[] args) {
5         //실수값 저장
6         float fValue = 3.14f;
7         double dValue = 3.14;
8         System.out.println("fValue = " + fValue);
9         System.out.println("dValue = " + dValue);
10
11         //정밀도 테스트
12         fValue = 0.1234567890123456789f;
13         dValue = 0.1234567890123456789;
14         System.out.println("fValue = " + fValue);
15         System.out.println("dValue = " + dValue);
16
17         //e 사용하기
18         int iData = 4500000;
19         float fData = 3e7f;
20         double dData1 = 3e7;
21         double dData2 = 3e-7;
22         System.out.println("iData = " + iData);
23         System.out.println("fData = " + fData);
24         System.out.println("dData1 = " + dData1);
25         System.out.println("dData2 = " + dData2);
26     }
27 }
```



```
DataType2.java
1 package week3;
2
3 public class DataType2 {
4     public static void main(String[] args) {
5         //실수값 저장
6         float fValue = 3.14f;
7         double dValue = 3.14;
8         System.out.println("fValue = " + fValue);
9         System.out.println("dValue = " + dValue);
10
11         //정밀도 테스트
12         fValue = 0.1234567890123456789f;
13         dValue = 0.1234567890123456789;
14         System.out.println("fValue = " + fValue);
15         System.out.println("dValue = " + dValue);
16
17         //e 사용하기
18         int iData = 4500000;
19         float fData = 3e7f;
20         double dData1 = 3e7;
21         double dData2 = 3e-7;
22         System.out.println("iData = " + iData);
23         System.out.println("fData = " + fData);
24         System.out.println("dData1 = " + dData1);
25         System.out.println("dData2 = " + dData2);
26     }
27 }
```

```
Console
<terminated> DataType2 [Java Application] C:\Program Files\Java
fValue = 3.14
dValue = 3.14
fValue = 0.12345679
dValue = 0.12345678901234568
iData = 4500000
fData = 3.0E7
dData1 = 3.0E7
dData2 = 3.0E-7
```




```
DataType2.java  x  DataType4.java
11      //정밀도 테스트
12      fValue = 0.1234567890123456789f;
13      dValue = 0.1234567890123456789;
14      System.out.println("fValue = " + fValue);
15      System.out.println("dValue = " + dValue);
16
17      //e 사용하기
18      int    iData  = 4500000;
19      float  fData  = 3e7f;
20      double dData1 = 3e7;
21      double dData2 = 3e-7;
22      System.out.println("iData  = " + iData);
23      System.out.println("fData  = " + fData);
24      System.out.println("dData1 = " + dData1);
25      System.out.println("dData2 = " + dData2);
26
27      System.out.println();
28
29      System.out.printf("iData  = %d\n", iData);
30      System.out.printf("fData  = %f\n", fData);
31      System.out.printf("dData1 = %f\n", dData1);
32      System.out.printf("dData2 = %f\n", dData2);
33
34      System.out.println();
35      System.out.printf("fData  = %.4f\n", fData);
36      System.out.printf("dData1 = %.2f\n", dData1);
37      System.out.printf("dData2 = %10.8f\n", dData2);
38  }
39 }
```



```

11 //정밀도 테스트
12 fValue = 0.1234567890123456789f;
13 dValue = 0.1234567890123456789;
14 System.out.println("fValue = " + fValue);
15 System.out.println("dValue = " + dValue);
16
17 //e 사용하기
18 int iData = 4500000;
19 float fData = 3e7f;
20 double dData1 = 3e7;
21 double dData2 = 3e-7;
22 System.out.println("iData = " + iData);
23 System.out.println("fData = " + fData);
24 System.out.println("dData1 = " + dData1);
25 System.out.println("dData2 = " + dData2);
26
27 System.out.println();
28
29 System.out.printf("iData = %d\n", iData);
30 System.out.printf("fData = %f\n", fData);
31 System.out.printf("dData1 = %f\n", dData1);
32 System.out.printf("dData2 = %f\n", dData2);
33
34 System.out.println();
35 System.out.printf("fData = %.4f\n", fData);
36 System.out.printf("dData1 = %.2f\n", dData1);
37 System.out.printf("dData2 = %10.8f\n", dData2);
38 }
39 }
    
```

Console

<terminated> DataType2 [Java Application] C:\Program Files\Java

```

fValue = 3.14
dValue = 3.14
fValue = 0.12345679
dValue = 0.12345678901234568
iData = 4500000
fData = 3.0E7
dData1 = 3.0E7
dData2 = 3.0E-7

iData = 4500000
fData = 30000000.000000
dData1 = 30000000.000000
dData2 = 0.000000

fData = 30000000.0000
dData1 = 30000000.00
dData2 = 0.00000030
    
```



DataType3.java

```
1 package week3;
2
3 public class DataType3 {
4     public static void main(String[] args) {
5         char cData1 = 'A';           //문자를 직접 저장
6         char cData2 = 65;             //10진수로 저장
7         char cData3 = '\u0041';       //16진수로 저장
8         System.out.println("cData1 = " + cData1);
9         System.out.println("cData2 = " + cData2);
10        System.out.println("cData3 = " + cData3);
11
12        char cData4 = '가';           //문자를 직접 저장
13        char cData5 = 44032;          //10진수로 저장
14        char cData6 = '\uac00';       //16진수로 저장
15        System.out.println("cData4 = " + cData4);
16        System.out.println("cData5 = " + cData5);
17        System.out.println("cData6 = " + cData6);
18
19        int cData7 = 'B';              //char 타입 변수의
20        int cData8 = '하';             //유니코드를 알고 싶으면
21        int cData9 = '&';              //int 타입 변수에 저장
22        System.out.println("cData7 = " + cData7);
23        System.out.println("cData8 = " + cData8);
24        System.out.println("cData9 = " + cData9);
25
26        int cData10 = cData7+cData9;
27        char cData11 = (char) (cData7+cData9);
28
29        System.out.println("cData7 + cData9 = " + cData10);
30        System.out.println("(cData7+cData9)의 char = " + cData11);
31    }
32 }
```



DataType3.java

```

1 package week3;
2
3 public class DataType3 {
4     public static void main(String[] args) {
5         char cData1 = 'A';           //문자를 직접 저장
6         char cData2 = 65;             //10진수로 저장
7         char cData3 = '\u0041';       //16진수로 저장
8         System.out.println("cData1 = " + cData1);
9         System.out.println("cData2 = " + cData2);
10        System.out.println("cData3 = " + cData3);
11
12        char cData4 = '가';           //문자를 직접 저장
13        char cData5 = 44032;          //10진수로 저장
14        char cData6 = '\uac00';       //16진수로 저장
15        System.out.println("cData4 = " + cData4);
16        System.out.println("cData5 = " + cData5);
17        System.out.println("cData6 = " + cData6);
18
19        int cData7 = 'B';              //char 타입 변수의
20        int cData8 = '하';             //유니코드를 알고 싶으면
21        int cData9 = '&';              //int 타입 변수에 저장
22        System.out.println("cData7 = " + cData7);
23        System.out.println("cData8 = " + cData8);
24        System.out.println("cData9 = " + cData9);
25
26        int cData10 = cData7+cData9;
27        char cData11 = (char) (cData7+cData9);
28
29        System.out.println("cData7 + cData9 = " + cData10);
30        System.out.println("(cData7+cData9)의 char = " + cData11);
31    }
32 }
    
```

Console

<terminated> DataType3 [Java Application] C:\Program Files\Java

```

cData1 = A
cData2 = A
cData3 = A
cData4 = 가
cData5 = 가
cData6 = 가
cData7 = 66
cData8 = 54616
cData9 = 38
cData7 + cData9 = 104
(cData7+cData9)의 char = h
    
```



❖ 다음 프로그램 소스에서 잘못된 부분을 고쳐서 출력하시오

```
DataType4.java
1 package week3;
2
3 public class DataType4 {
4     public static void main(String[] args) {
5         byte    bData = 128;
6         char     cData = -90;
7         float    fData = 3.14;
8         long     lData = 100000000000;
9         boolean  boolD = 10;
10
11         System.out.println("bData = " + bData);
12         System.out.println("cData = " + cData);
13         System.out.println("fData = " + fData);
14         System.out.println("lData = " + lData);
15         System.out.println("boolD = " + boolD);
16     }
17 }
```



```
DataType4.java
1 package week3;
2
3 public class DataType4 {
4     public static void main(String[] args) {
5         byte    bData = 127;
6         char     cData = 90;
7         float    fData = 3.14f;
8         long     lData = 10000000000L;
9         boolean  boolD = true;
10
11         System.out.println("bData = " + bData);
12         System.out.println("cData = " + cData);
13         System.out.println("fData = " + fData);
14         System.out.println("lData = " + lData);
15         System.out.println("boolD = " + boolD);
16     }
17 }
```

Console

<terminated> DataType4 [Java Application] C:\Program Files\Java\jre6\bin\java.exe

bData = 127
cData = Z
fData = 3.14
lData = 10000000000
boolD = true

