

02 자바 프로그래밍 기초

Sangwook Lee

02 자바 프로그래밍 기초

1. 자바 구조와 문장

2. 자료형과 변수

3. 자료의 입력과 출력

1. 자바 구조와 문장

- 클래스와 패키지

- 클래스(class)

- 자바의 프로그램 단위
 - public 클래스 이름 = 파일 이름

- 패키지(package)

- 자바 클래스가 모여 있는 폴더
 - 생성 방법 2가지
 - ❖ 클래스 생성 시 패키지 이름 기술
 - ❖ 메뉴에서 **File > New > Package** 선택
 - package 이름 지정 문장은 자바 소스 맨 앞에 위치

1. 자바 구조와 문장

- 클래스 구조

- 클래스 = 필드 + 메소드

- 필드(field)

- 변수 ➡ 속성 구현

- 메소드(method)

- 함수 ➡ 기능 구현

- 형태

- ❖ 헤드 = 접근 수정자 + 반환형 + 이름 + (+ 인자(들) +)

- ❖ 블록 = { + 문장(들) + }

- public static void main(String[] args)

- ❖ 프로그램 시작 메소드

1. 자바 구조와 문장

• 클래스 구조

[예제 2-1] 메소드 구현과 호출하기

HelloProgramming.java

```
01 package programming.basic;
02
03 public class HelloProgramming {
04
05     //필드 선언
06     static String fd = " -- 필드";
07     static String md = " -- 메소드";
08
09     //자바 프로그램을 실행하는 main() 메소드 구현
10     public static void main(String[] args) {
11         System.out.println("자바 클래스의 구조");
12         write(fd);
13         write(md);
14     }
15
16     //프로그래머가 정의하는 메소드 write()의 구현
17     public static void write(String word) {
18         System.out.println(word);
19     }
20 }
```

메소드 헤드(머리)

메소드 구현(정의)

write() 메소드 호출

Note

메소드 main()의 구현 11, 12, 13줄을 실행하고 종료한다. 각각 12, 13줄에서 17줄로 이동하여 18줄의 출력을 실행한 후 다시 main()으로 돌아온다.

결과

자바 클래스의 구조
-- 필드
-- 메소드

1. 자바 구조와 문장

- 키워드와 식별자

- 키워드(keyword)

- (미래 정의해 놓은) 문법적으로 의미 있는 단어
 - 키워드 = 예약어(reserved word)

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

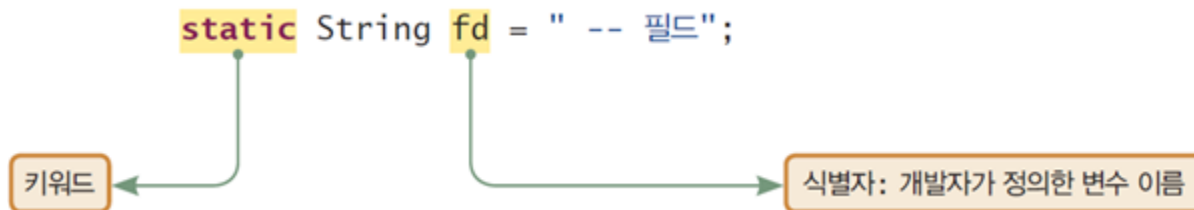
<자바의 키워드>

1. 자바 구조와 문장

- 키워드와 식별자

- 식별자(identifiers)

- 프로그래머가 정의하여 사용하는 단어
 - 클래스 이름, 변수 이름, 함수 이름 등
 - 식별자 규칙
 - ❖ 영문자, 숫자, 밑줄(_), 달러 기호(\$)
 - ❖ 키워드는 식별자가 될 수 없음
 - ❖ 첫 문자로 숫자가 나올 수 없음
 - ❖ 중간에 공백 문자가 들어갈 수 없음



<키워드와 식별자>

1. 자바 구조와 문장

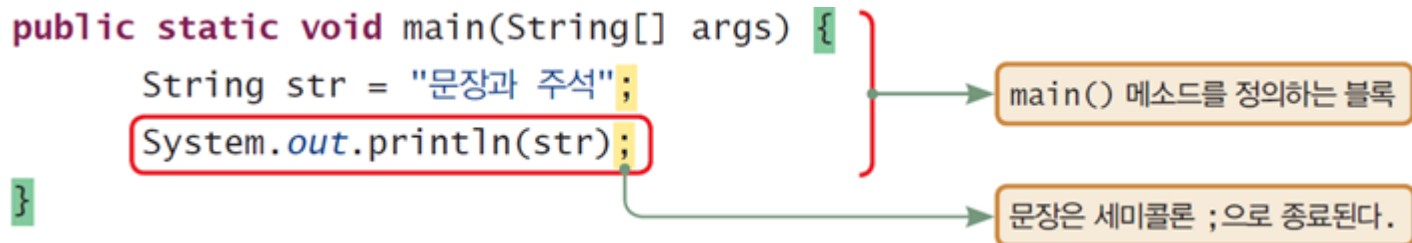
- 문장과 주석

- 문장(statement)

- 컴퓨터에게 명령을 내리는 최소 단위
 - 세미콜론(;)으로 종료

- 블록(block)

- 여러 문장의 묶음
 - 클래스나 메소드 정의 시 사용



<문장과 블록>

1. 자바 구조와 문장

- 문장과 주석

- 주석(comments)

- 프로그램 실행에 영향을 미치지 않는 설명문
 - //
 - ❖ 한 줄 주석
 - ❖ // 이후부터 그 줄의 마지막까지 주석 처리
 - /* ~ */
 - ❖ 블록 주석
 - ❖ 여러 줄에 걸쳐 주석 처리

02 자바 프로그래밍 기초

1. 자바 구조와 문장

2. 자료형과 변수

3. 자료의 입력과 출력

2. 자료형과 변수

- 상수, 변수, 자료형
 - 상수(literal)
 - 자료 자체
 - 변수(variable)
 - 자료 저장 공간
 - 자료형(data type)
 - 자료 형태

2. 자료형과 변수

- 자바의 자료형

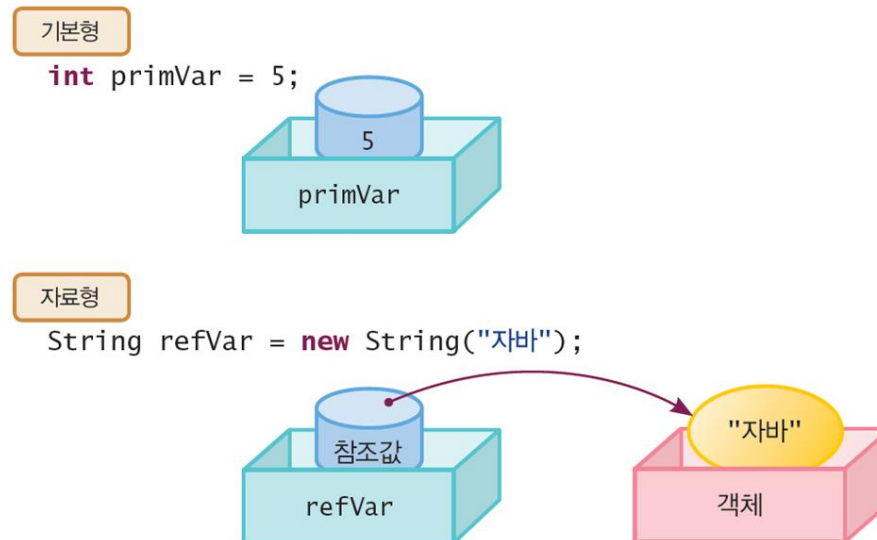
구분	분류	키워드
기본형	논리형	boolean
	문자형	char
	정수형	byte, short, int, long
	실수형	float, double
참조형	배열	int [], float [] 등 다양
	클래스	String, Date 등 다양
	인터페이스	Runnable, Enumeration 등 다양

<변수 선언 시 지정하는 자료형>

2. 자료형과 변수

- 자바의 자료형

- 기본형(primitive data type)
 - 저장 공간에 자료 자체 저장
- 참조형(reference data type)
 - 저장 공간에 객체 주소 저장



<기본형과 참조형의 차이>

2. 자료형과 변수

- 자바의 자료형
 - 자료형 크기

분류	키워드	크기	상대적 크기 비교	최소 ~ 최대	지수형태 범위
논리형	boolean	1바이트		false, true	
문자형	char	2바이트		\u0000 ~ \uffff, [0 ~ 65,535]	$0 \sim 2^{16}-1$
정수형	byte	1바이트		-128 ~ 127	$-2^7 \sim 2^7-1$
	short	2바이트		-32,768 ~ 32,767	$-2^{15} \sim 2^{15}-1$
	int	4바이트		-2,147,483,648 ~ 2,147,483,647	$-2^{31} \sim 2^{31}-1$
	long	8바이트		$-2^{63} \sim 2^{63}-1$	$-2^{63} \sim 2^{63}-1$
실수형	float	4바이트		(+, -)1.4E-45 ~ 3.4028235E38	
	double	8바이트		(+, -)4.9E-324 ~ 1.7976931348623157E308	

2. 자료형과 변수

- 상수와 표현
 - 상수

분류	자료형	상수	비고
논리형	boolean	true, false	2개 이외에는 없음
문자형	char	'A', '자', '%', '\', '\n', '\23'(8진수 코드 값) '\u2344' (16진수 코드 값)	작은 따옴표로 표기
정수형	int	10, 45(십진수), 0b1010(이진수 1010), 020(8진수 20), 0x1d(16진수 1d),	0X1D와 같이 대문자도 가능
	long	32l, 2200000000L	정수 뒤에 l이나 L 표기
실수형	float	3.14f, 5.25F	반드시 실수 뒤에 f나 F 표기
	double	2.98e2(2.98*10 ²) 5.8E-2(5.8*10 ⁻²) 3.75, 3.2d, 5.78D	e와 E 모두 가능 실수 뒤에 d나 D도 사용 가능
문자열	String	"문자의 모임이 문자열이래!"	큰 따옴표로 표기

2. 자료형과 변수

[예제 2-3] 상수 표현하기 (p.47)

HelloLiterals.java

```
01  /*
02   * 내용: 상수와 그 표현을 살펴보는 프로그램
03   */
04  package programming.basic;
05
06  public class HelloLiterals {
07      static long creditCardNumber = 1234_5678_9012_3456L;
08      static long socialSecurityNumber = 999_99_9999L;
09      static float pi = 3.14_15F;
10
11      public static void main(String[] args) {
12          System.out.println(0b1010);    //이진수
13          System.out.println(016);       //8진수
14          System.out.println(0x1F);     //16진수
15          System.out.println(3.14f);     //float 형
16          System.out.println(3.1415D);   //double 형
17          System.out.println(creditCardNumber);
18          System.out.println(socialSecurityNumber);
19          System.out.println(pi);
20      }
21  }
```

결과

```
10
14
31
3.14
3.1415
1234567890123456
999999999
3.1415
```

2. 자료형과 변수

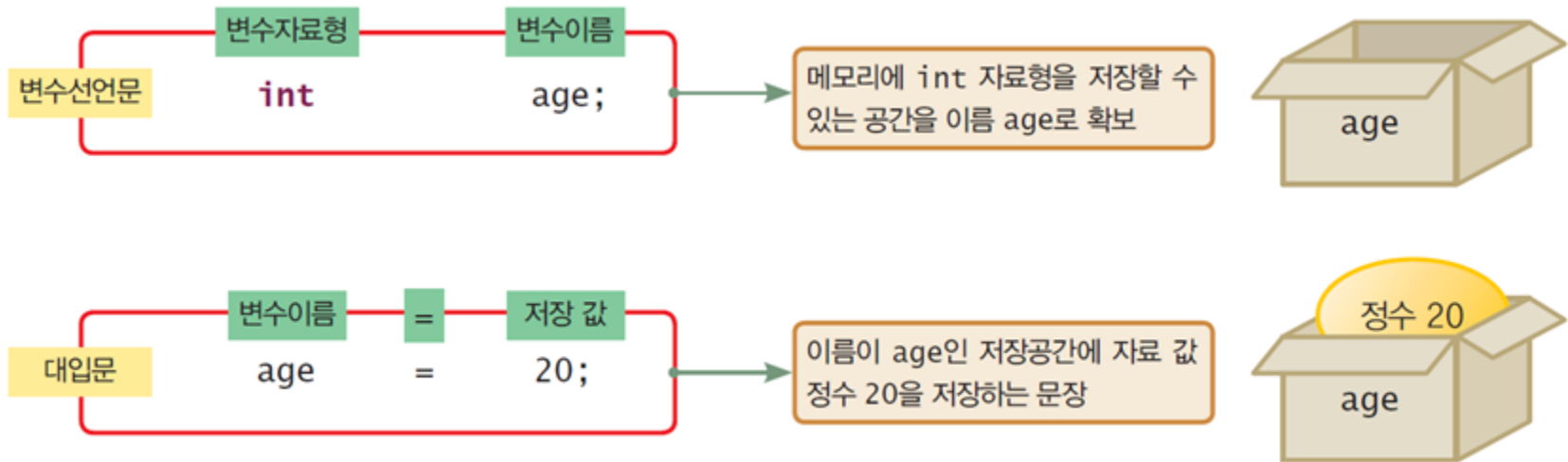
- 상수와 표현
 - 특수 문자

문자표현	이름	의미
\n	New line	새로운 행으로 이동
\t	Tab	탭만큼 이동
\b	Back space	하나 뒤로 이동
\r	Carriage return	현재 행의 처음으로 이동
\f	Form feed	새 페이지의 처음으로 이동
\\	Back slash	\\(역슬래쉬) 문자
\'	Single quote	'(작은 인용부호) 문자
\"	Double quote	”(큰 인용부호) 문자
\ddd	d는 8진수의 한 자릿수	\0에서 \377(\u00ff)까지, 8비트의 정보만 입력 가능
\udddd	d는 16진수의 한 자릿수	반드시 4자리를 기술

2. 자료형과 변수

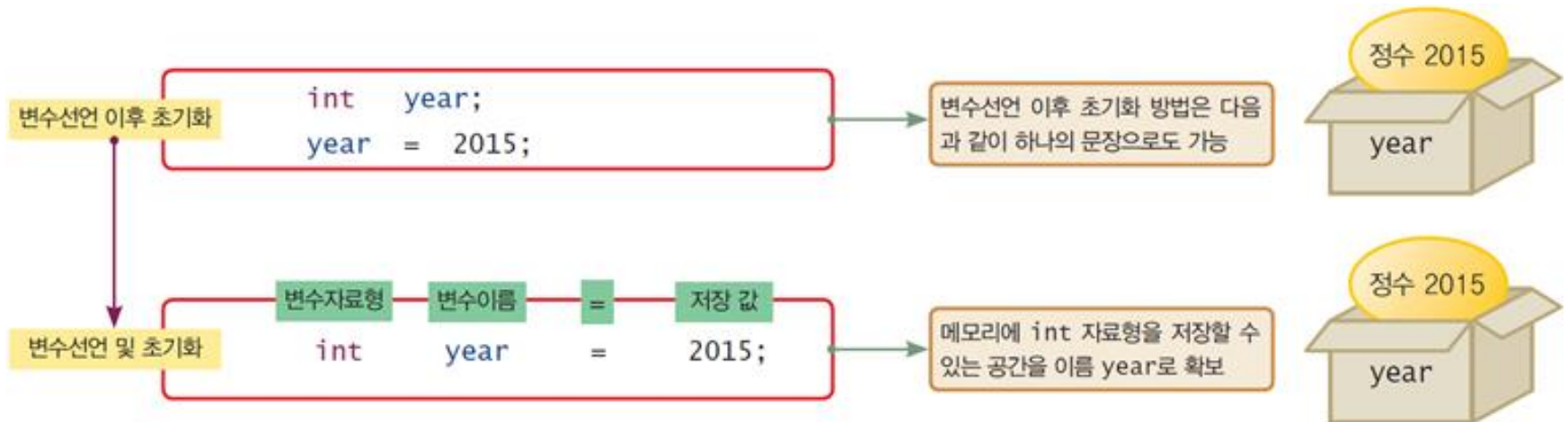
• 변수 선언과 초기화

- 변수를 사용하기 위해 변수 선언 필요
- 변수 선언은 자료형과 변수 이름 나열
- 선언된 변수에 자료를 저장하려면 대입연산자(=) 사용



2. 자료형과 변수

- 변수 선언과 초기화
 - 변수 초기화



2. 자료형과 변수

- 소속변수와 지역변수

- 소속변수(membered variables) = 필드(field)

- 클래스 내부에서 선언
 - 여러 메소드에서 사용
 - 초기 값 미지정 시 기본 값(0 / false / null) 저장

- 지역변수(local variables)

- 메소드 내부에서 선언
 - 선언된 메소드에서만 사용
 - 초기 값 미지정 시 컴파일 오류

2. 자료형과 변수

- 소속변수와 지역변수
 - 필드와 지역변수의 선언과 사용

```
public class HelloDefaultVaule {  
    //필드 선언  
    static int def; 필드 def  
  
    public static void main(String[] args) {  
        //지역변수 선언  
        int n = 100; 지역변수 n  
        System.out.println(n);  
  
        //필드 사용 가능  
        System.out.println(def);  
    }  
  
    public static void test() {  
        ...  
        //필드 사용 가능  
        System.out.println(def);  
    }  
}
```

필드 사용 가능 범위

지역변수 사용 가능 범위

2. 자료형과 변수

[예제 2-6] 필드 초기 값 출력하기 (p.53)

HelloDefaultValue.java

```
01  /*
02   * 내용: 필드와 지역변수의 초기 값 체크 프로그램
03   */
04  package programming.basic;
05
06  public class HelloDefaultVaule {
07      //필드 선언
08      static int def;      //현재 static을 빼면 문법오류 발생
09      static boolean bool; //현재 static을 빼면 문법오류 발생
10      static String name;  //현재 static을 빼면 문법오류 발생
11
12      public static void main(String[] args) {
13          //지역변수 선언
14          int n = 100;
15          System.out.println(n);
16
17          //필드 사용 가능
18          System.out.println(def);
19          System.out.println(bool);
20          System.out.println(name);
21      }
22  }
```

결과
100
0
false
null

3. 자료의 입력과 출력

- 자료 입력

- Scanner 클래스 사용

- 메소드

- next(): 입력 문자를 String 반환

- nextInt(): 입력 정수를 int 반환

- nextFloat(): 입력 실수 float로 반환

- 문자열 입력 예

① `java.util.Scanner input = new java.util.Scanner(System.in);`

자료형

변수이름 = new

클래스이름(인자);

객체를 만들려면 키워드 new
다음에 생성자를 호출

② `String name = input.next();`

3. 자료의 입력과 출력

[예제 2-7] Scanner 클래스로 자료 입력하기 (p.55)

HelloInput.java

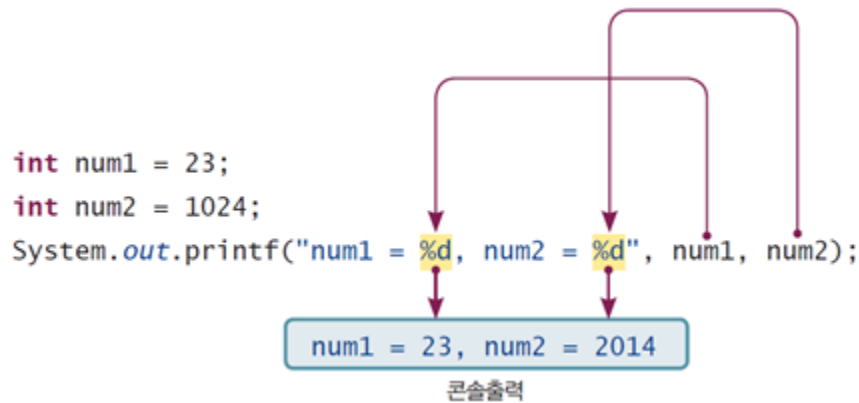
```
01 package programming.io;
02
03 public class HelloInput {
04     public static void main(String[] args) {
05         java.util.Scanner input = new java.util.Scanner(System.in);
06         System.out.println("이름과 학번을 입력하세요.");
07         String name = input.next();
08         int num = input.nextInt();
09         System.out.print("이름: " + name + ", 학번: " + num);
10     }
11 }
```

결과 이름과 학번을 입력하세요.
홍길동 2013
이름: 홍길동, 학번: 2013

3. 자료의 입력과 출력

- 자료 출력

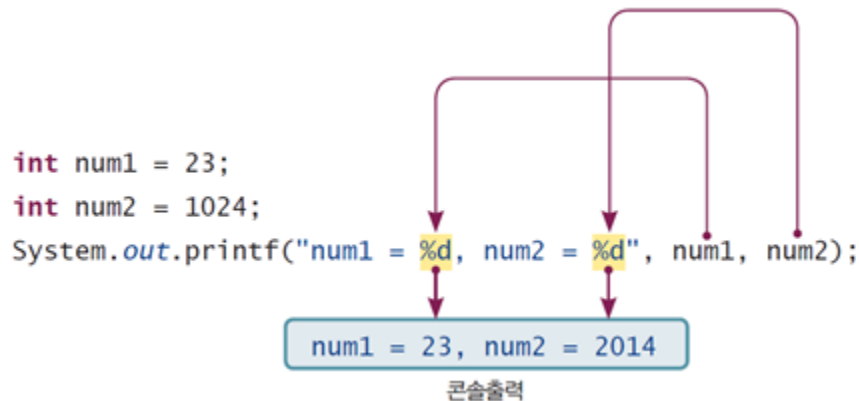
- System 클래스 사용
- System의 out 필드 메소드
 - print(): 자료 출력
 - println(): 자료 출력 후 다음 줄로 이동
 - printf(): 인자를 지정한 형식대로 출력
- printf() 출력



3. 자료의 입력과 출력

- 자료 출력

- System 클래스 사용
- System의 out 필드 메소드
 - print(): 자료 출력
 - println(): 자료 출력 후 다음 줄로 이동
 - printf(): 인자를 지정한 형식대로 출력
- printf() 출력



3. 자료의 입력과 출력

- 자료 출력

- printf() 메소드의 형식 지정자(format specifiers)

출력 지정자	설명
%b %B	논리 값 출력
%h %H	16진수로 출력
%s %S	문자열 출력
%c %C	문자 출력
%d	십진수로 출력
%o	8진수로 출력
%x %X	16진수로 출력
%f	실수로 출력
%%	% 출력
%n	새로운 줄(new line)로 이동

3. 자료의 입력과 출력

[예제 2-9] printf() 메소드로 자료 출력하기 (p.58)

HelloSimplePrintf.java

```
01 package programming.io;
02
03 public class HelloSimplePrintf {
04     public static void main(String[] args) {
05         byte age = 27;
06         int weight = 70;
07         double height = 175.8;
08
09         System.out.printf("나이: %d, 몸무게: %d, 키: %f, ", age, weight, height);
10         System.out.printf("%s: %c 형", "혈액형", 'O');
11     }
12 }
```

결과 나이: 27, 몸무게: 70, 키: 175.800000, 혈액형: O 형

프로그래밍 연습

- 클래스 `PersonInfo`에서 다음 조건을 만족하는 프로그램을 작성하시오.
 - 몸무게(정수)와 키(실수)를 입력 받아서 변수에 저장
 - 입력 받은 몸무게와 키를 출력