

JAVA 프로그램 실습

Data Type

경북대학교
소프트웨어융합과
교수 배희호

학습 목표

- JAVA 언어는 Memory에 직접 접근하거나 Bit 단위의 연산을 수행하여 Data를 효율적으로 처리할 수 있다. 이런 장점을 잘 활용하려면 Data가 Memory에 저장되는 방식을 알아야 한다. 이번에는 정수, 실수, 문자가 Compile된 후에 어떤 형태로 바뀌는지 살펴보자
 - 모든 Data는 Computer 안에서 0과 1의 Bit열로 바뀐다
 - 상수는 그 값을 변경할 수 없다
 - 10은 정수 리터럴, 10.0은 실수 리터럴, 'a'는 문자 리터럴, "a"는 문자열 리터럴
 - 양수는 4 Byte의 2진수로, 음수는 2의 보수로 번역
 - 실수는 IEEE 754 표준의 double 형식에 따라 번역

학습 목표

- Program이 처리하는 Data의 형태가 다양하고 연산 방법도 다양
 - 이런 특징을 잘 이해하면 효율적이고 신뢰성 있는 Program을 만들 수 있음
- 이번에는 Data Type에 따른 변수 선언과 사용 방법에 대해 알아보자
 - 변수 선언으로 Memory에 Data 저장 공간을 확보한다
 - 대입 연산자는 변수에 값을 저장한다
 - Data Type에 따라 적합한 변수를 선언하여야 함
 - 초기화하지 않은 변수는 Garbage 값을 가짐

Identifier와 Keyword

- Keyword는 Compiler와 약속된 단어이고, Identifier는 사용자가 새롭게 만들어낸 단어 임
- JAVA Identifier
 - 클래스 이름, 메소드 이름, 변수 이름 등
 - UniCode 사용
- 작성 규칙
 - 문자와 숫자의 연속
 - 문자로 시작
 - Keyword, true, false, null과 같은 철자를 사용할 수 없음
 - ‘_’, ‘\$’ 등으로 시작할 수 있음
 - 대·소문자를 구별
 - 길이에 제한이 없음

Identifier와 Keyword

■ 올바른 Identifier 예

- id
- userName
- user_name
- _userName
- \$userName

■ 틀린 Identifier 예

- user name // 공백
- 3d_studio // 숫자로 시작할 수 없음
- this // keyword임
- #arg // 특수문자 #는 사용할 수 없음

Identifier와 Keyword

■ JAVA Keyword (49개)

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

Identifier와 Keyword

■ JAVA Keyword

주의

- 1) const와 goto는 예약어로 지정됐지만 현재 JAVA에서는 사용되지 않음
- 2) "true", "false", "null"은 boolean/null 상수로서 예약어처럼 사용
- 3) assert는 J2SE 1.4에서 추가됨
- 4) enum은 J2SE 5.0에서 추가됨

Identifier와 Keyword 예제

■ 다음 중 Keyword가 아닌 것은? (모두 고르시오)

- ① if
- ② True
- ③ NULL
- ④ class
- ⑤ System

Java에서는 대소문자를 구별하기 때문에 true는 키워드이지만 True는 키워드가 아니다

JAVA Program 구조

```
public class Main {  
    String school = "경북대학교";
```

필드(Field) – 전역변수

```
    public Main() {  
        String school = "경북대학교";  
    }
```

생성자
Constructor

지역변수

클래스

```
    public static void main(String[] args) {  
        String school = "경북대학교";  
        System.out.printf(school);  
    }
```

메소드
Method

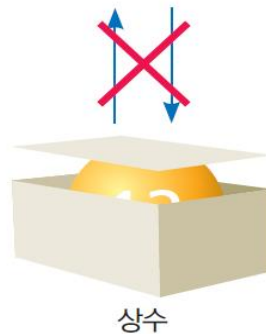
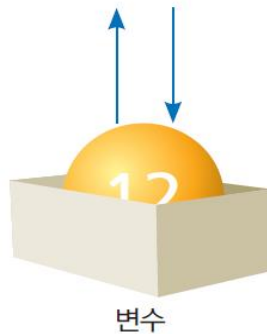
지역변수

```
}
```

지역변수가 전역변수보다 우선 사용

Data Types

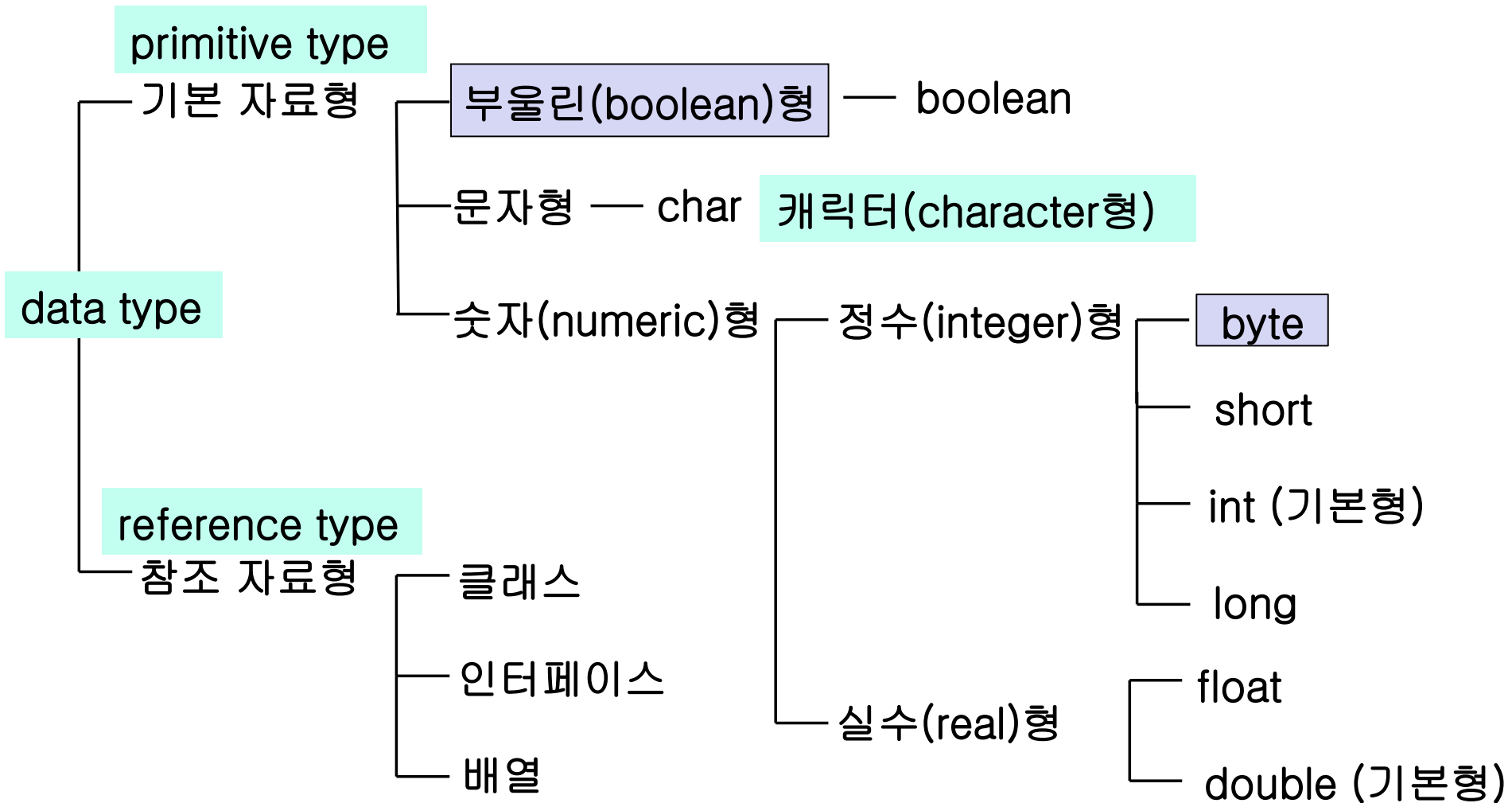
- Variable(변수)
 - Memory에 있는 저장 위치(RAM)
 - 저장된 값의 변경이 가능한 공간
- Constant(상수, 리터럴)
 - 값을 저장할 수 있는 Memory 공간
 - 저장된 값의 변경이 불가능한 공간
 - 예) 3.14, 100, 'A', "Hello World!"



변수는 실행도중에
값을 변경할수 있으나
상수는 한번 값이
정해지면 변경이
불가능 합니다.



Data Type



Data Type

- Primitive type

- 8개

- (boolean, char, byte, short, int, long, float, double)

- 실제 값을 저장

- Reference type

- 기본형을 제외한 나머지(String, System 등)

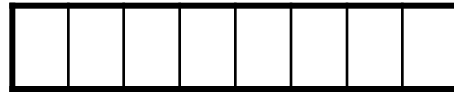
- 객체의 주소를 저장(4 byte, 0x00000000~0xffffffff)

Data Type

1 bit



8 bit = 1 byte



byte $-2^7 \sim 2^7-1$



short $-2^{15} \sim 2^{15}-1$



char $0 \sim 2^{16}-1$



int $-2^{31} \sim 2^{31}-1$



long $-2^{63} \sim 2^{63}-1$



float $1+8+23=32$ bit = 4 byte



double $1+11+52=64$ bit = 8 byte



Data Type

■ System에서 Data Type의 크기를 알아보자

```
public static void main(String[] args) {  
    System.out.println("byte : " + Byte.BYTES  
                        + "(" + Byte.SIZE + ")");  
    System.out.println("char : " + Character.BYTES  
                        + "(" + Character.SIZE + ")");  
    System.out.println("short : " + Short.BYTES  
                        + "(" + Short.SIZE + ")");  
    System.out.println("int : " + Integer.BYTES  
                        + "(" + Integer.SIZE + ")");  
    System.out.println("long : " + Long.BYTES  
                        + "(" + Long.SIZE + ")");  
    System.out.println("float : " + Float.BYTES  
                        + "(" + Float.SIZE + ")");  
    System.out.println("double : " + Double.BYTES  
                        + "(" + Double.SIZE + ")");  
}
```

Data Type

Data Type	Default Value
boolean	false
byte	0
short	0
int	0
long	0L
char	u0000
float	0.0f
double	0.0
object reference	null

Data Type

- 다음 보기에서 적절한 데이터 형을 모두 선택하시오.

보 기

char, byte, short, int, long , float, double

- ① 소숫점을 포함하는 실수형 값을 저장할 수 있는 데이터 형은? double, float
- ② 가장 큰 값을 저장할 수 있는 데이터 형은? double
- ③ 정수형 값을 저장할 수 있는 char, byte, short, int, long
- ④ 가장 큰 정수 값을 저장할 수 있는 데이터 형은? long

Data Type

- 다음과 같은 데이터를 처리하기 위한 적절한 데이터 형을 보기에서 모두 선택하시오.

보 기

char, byte, short, int, long, double, float

- ① 연령별 취업률 또는 실업률
- ② 국가별 인구수 비교
- ③ 월드컵 축구경기의 스코어
- ④ 세계 육상 대회에 출전한 선수들의 100m 달리기 기록
- ⑤ 태양계 행성간의 거리 비교
- ⑥ JAVA 언어 과목을 배운 학생들의 학점표시
- ⑦ 우리나라 각 지역의 온도변화

Data Type

- ① 실업률은 보통 소숫점을 포함한 실수로서 %로 표시됨
실수형 (float)
- ② 국가별 인구수의 최대값은 대략 15억(2003년 중국)
정수형(long)
- ③ 월드컵 축구경기의 최대 스코어가 100점 이상 되는 경우는 없음
정수형(short)
- ④ 세계 육상 대회에 출전한 선수들의 100m 달리기 기록은 대략 9초에서 12초 사이가 될 것이나 소숫점을 포함함
실수형 (float)
- ⑤ 태양과 수성사이의 거리는 57,910,000km이고, 태양과 해왕성 사이의 거리는 4,504,000,000km
정수형(long)
- ⑥ 만약 A~F로 학점을 표시한다면 char 형을,
4.5(또는 4.0)을 만점으로 표시한다면 float 형을 사용

Data Type

- ⑦ 우리나라는 섭씨온도를 이용하고, 제일 추운 곳은 북쪽의 중강진(영하 40도), 제일 더운 곳은 영상 40도 이상을 넘는 경우가 거의 없지만 소숫점을 포함할 수 있으므로 float 형을 사용

Variable, Constant, Literal

- Variable(변수)
 - 하나의 값을 저장하기 위한 공간
- Constant(상수)
 - 한 번만 값을 저장할 수 있는 공간
- Literal(리터럴)
 - 그 자체로 값을 의미하는 것

```
int score = 100;  
score = 200;  
char ch = 'A';  
String str = "abc";  
final int MAX = 100;  
MAX = 200; // 에러
```

Literal

- Literal 종류

- 정수 리터럴 (Integer literals)

- 실수 리터럴 (Floating-point literals)

- 문자 리터럴 (Character literals)

- 문자열 리터럴 (String literals)

- 논리형 리터럴 (Boolean literals)

Literal

■ 정수형 리터럴

리터럴의 종류	예	설명
10진수	4, 10, 80	
2진수	0b0101, 0b0111	맨 앞에 숫자 '0b'을 덧붙이면 2진 상수로 인식
8진수	04, 012, 0100	맨 앞에 숫자 '0'을 덧붙이면 8진 상수로 인식
16진수	0x4, 0xA, 0x6	맨 앞에 숫자 '0'과 영문자 'x', 즉 '0x'를 덧붙이면 16진 상수로 인식
long 형	10L, 034L, 0X2AL	10진, 8진, 16진 상수 뒤에 L을 덧붙임

Literal

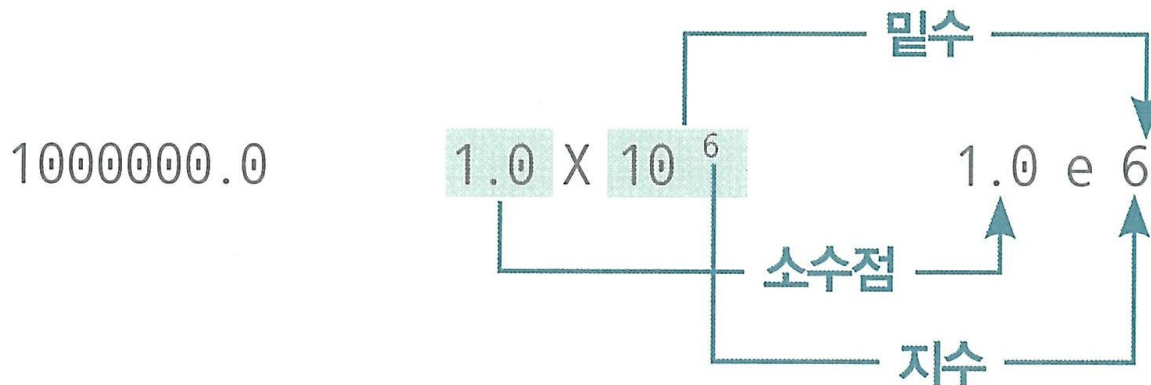
- 정수형 리터럴
 - 키워드 int를 기본
 - byte, short, int, long이 있음

byte		short		int		long
1 바이트	<	2 바이트	<=	4바이트 혹은 2바이트	<=	4바이트

Literal

■ 실수형 리터럴

리터럴 종류	예	설명
소수형	1234.5, 0.0000987	가장 일반적으로 사용하는 실수형 데이터
지수형	1.2345E3, 0.987E-5	영문자 E를 기준으로 앞에는 가수부 뒤에는 지수부를 기술함
float 형	1234.5f, 1.234E3f	실수형 상수 뒤에 'f'를 덧붙임
double 형	1234.5d, 1.234E3d	실수형 상수 뒤에 'd'를 덧붙임



Literal

■ 실수형 리터럴

- float(4 바이트) : 16.42f
- double(64 바이트) : 16.42
- float < double
- float는 소수점 이하 6자리까지 표현 가능
- double은 소수점 이하 15자리 가지 표현 가능
- 기본 Data Type은 double

소수점 형태

같은 값을 표현한 다양한 지수 형태

0.0000314

3.14e-5

0.0314e-3

0.00000314e1

314e-7

정규화

Literal

■ 논리형 리터럴

- 논리형 리터럴은 참(true)과 거짓(false)을 표현할 때 사용하는 Logical Data

- 다른 Data Type으로 변환하지 못함

- 예) `boolean flag = true; (0)`

`boolean b = 0; (X)`

// true와 false 의 값이 아닌 다른 형을 저장했으니
Error 발생

`boolean c = FALSE; (X)`

// false가 대문자로 표현되어 있으므로 Error 발생

Literal

■ 문자 리터럴

- 문자형은 'A' '밥' 과 같이 하나의 문자를 홑 따옴표(' ')를 사용해서 표현
- 영문자 이외의 Unicode 문자를 표현할 때는 '₩(역슬래시)u'를 이용해서 표현 (4자리 16진수)
- 예) '₩uC790' --> '자'
 '₩uBC14 --> '바'

Literal

■ null 리터럴

- 어떠한 Reference Type의 값으로도 사용 가능

- ~~int n = null;~~ // Primitive Data Type에는 사용 불가

- String str = null;

■ 문자열 리터럴

- 이중 인용부호로 묶어서 표현

- "Good", "Morning", "자바", "3.19", "26", "a"

- JAVA에서 문자열은 객체이므로 Primitive Type 아님

- 문자열 리터럴은 String 객체로 자동 처리

```
String str1 = "Welcome";  
String str2 = null;  
System.out.println(str1);
```

Literal

■ 리터럴과 접미사

```
boolean power = true;  
char ch = 'A';  
char ch = '₩u0041';  
char tab = '₩t';  
byte b = 127;  
short s = 32767;  
int i = 100;  
int oct = 0100;  
int hex = 0x100;  
long l = 1000000000000L;  
float f = 3.14f;  
double d = 3.14;  
float f = 100f;
```

```
10. -> 10.0  
.10 -> 0.10  
10f -> 10.0f  
3.14e3f -> 3140.0f  
1e1 -> 10.0
```

Literal 예제

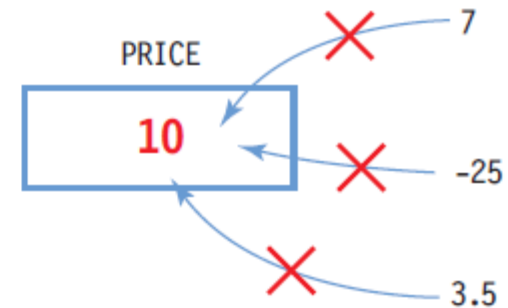
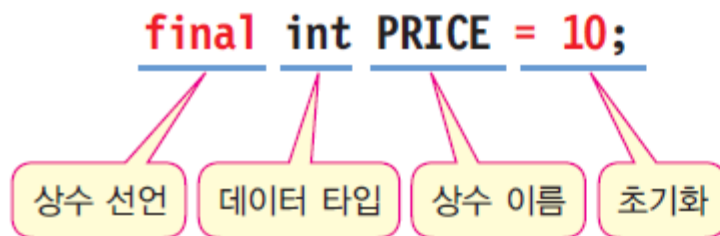
■ 다음 Program의 출력 결과는 ?

```
public static void main(String[] args) {  
    System.out.println(11);  
    System.out.println(011);  
    System.out.println(0x11);  
    System.out.println(11L);  
    System.out.println(1.5);  
    System.out.println(1.5f);  
    System.out.println('a');  
    System.out.println("abc");  
    System.out.println(true);  
}
```

11
9
17
11
1.5
1.5
a
abc
true

Constant

- 상수 선언
 - final 키워드 사용
 - 상수 이름은 관례상 대문자로 작성함
 - 선언 시 초기값 지정
 - 실행 중 값 변경 불가



- 상수 선언 사례

```
final double PI = 3.141592;  
final int LENGTH = 20;
```

Variables

■ 변수

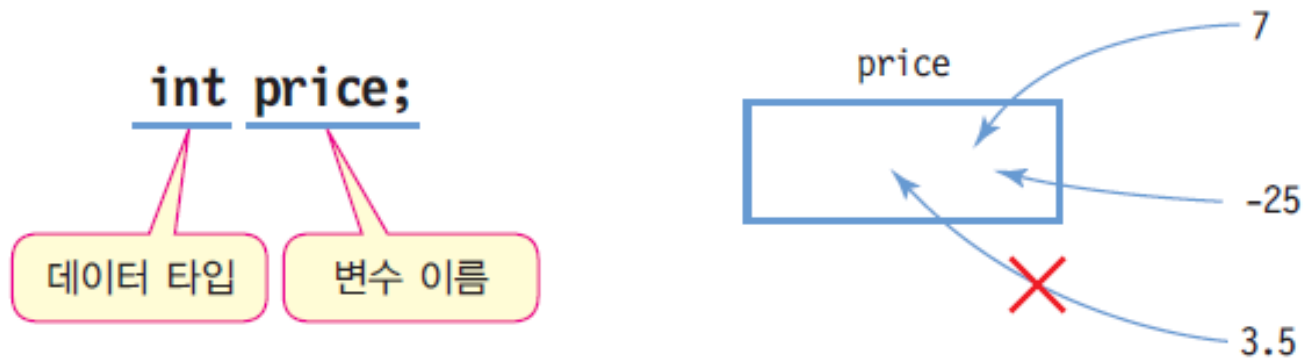
- Program 실행 중에 값을 임시 저장하기 위한 공간

- 변수 값은 Program 수행 중 변경될 수 있음

- Data Type에서 정한 크기의 Memory 할당

■ 변수 선언

- 변수의 Data Type 다음에 변수 이름을 적어 변수를 선언



Variables

- 변수의 종류

- 정수 변수 (Integer Variables)

- 실수 상수 (Floating-point Variables)

- 문자 상수 (Character constants)

- 문자열 상수 (String constants)

- 논리형 상수 (Boolean constants)

변수는 상수처럼 그냥 사용하지 못하고 변수 선언
과정을 거친 후 사용할 수 있음

Variables

■ 변수 선언 사례

```
int radius;  
char c1, c2, c3;    // 3 개의 변수를 한 번에 선언  
double weight;
```

■ 변수 선언과 초기화

■ 선언과 동시에 초기값 지정

```
int radius = 10;  
char c1 = 'a', c2 = 'b', c3 = 'c';  
double weight = 75.56;
```

■ 변수에 값 대입 (문장)

■ 대입 연산자인 = 다음에 식(expression)

```
radius = 10 * 5;  
c1 = 'r';  
weight = weight + 5.0;
```

Data Types 예제 0

- 정수 리터럴 5와 -5를 번역된 후 비트열을 표현하여라.

5 -> 00000000 00000000 00000000 00000101

5의 1의 보수-> 11111111 11111111 11111111 11111010

5의 2의 보수-> 11111111 11111111 11111111 11111011

Data Types 예제 1

■ 다음 중 기본형(primitive type)이 아닌 것은?

- ① int
- ② Byte
- ③ double
- ④ boolean

②

기본형은 boolean, byte, short, char, int, long, float, double 모두 8개이다.

그 외의 타입은 모두 참조형(reference type)이다.

Data Types 예제 2

■ JAVA에서 정수형과 실수형은 묵시적(default)으로 어떠한 형태(type)로 취급되는가?

① int, float

② int, double

③ byte, float

④ byte, double

②

JAVA에서 정수와 실수는 묵시적으로 int, double 형으로 나타낸다

Data Types 예제 3

- byte 형을 저장할 수 있는 데이터 값의 허용 범위는 얼마입니까?

-128 ~ 127

- char형의 저장할 수 있는 데이터 값의 허용 범위는 얼마입니까? (10진수로 적으시오)

0 ~ 65535

char는 2 byte($2 * 8 = 16$ bit)이므로 '2의 16제곱'개의 값을 표현할 수 있다. 2의 16제곱은 65536개이며, 0을 포함해야 하므로 0~65535(모두 65536개)가 char범위가 된다

Data Types 예제 4

- 주민등록번호를 숫자로 저장하고자 한다. 이 값을 저장하기 위해서는 어떤 자료형(data type)을 선택해야 할까?
- regNo라는 이름의 변수를 선언하고 자신의 주민등록번호로 초기화 하는 한 줄의 코드를 작성해보자

```
String regNo = "7701011111222";
```

주민등록 번호는 숫자처럼 보이지만, 문자이므로 문자열(String) 타입으로는 정의함

Data Types 예제 5

■ 참조형 변수(reference type)와 같은 크기의 기본형(primitive type)은? (모두 고르시오)

① int(4 byte)

② long(8 byte)

③ short(2 byte)

④ float(4 byte)

⑤ double(8 byte)

①, ④

모든 참조형 변수는 4 byte이므로, 크기가 4 byte인
기본형 타입을 고르면 됨

Variable

- 변수의 이름(identifier) 만드는 규칙
 - 대소문자가 구분되며 길이에 제한이 없음
 - True와 true는 서로 다른 것으로 간주
 - 예약어를 사용해서는 안 됨
 - true는 예약어라서 사용할 수 없지만, True는 가능
 - 숫자로 시작해서는 안 됨
 - top10은 허용하지만, 7up는 허용되지 않음
 - 특수문자는 '_'와 '\$'만을 허용
 - \$harp은 허용되지만, S#arp은 허용되지 않음

Variable

■ 다음 중 변수의 이름으로 사용할 수 있는 것은? (모두 고르시오)

① \$system

③ 7eleven

⑤ 자바

⑦ \$MAX_NUM

② channel#5

④ lf

⑥ new

⑧ hello@com

② : 허용하지 않는 특수문자 '#'를 사용할 수 없음

③ : 숫자로 시작하면 안 됨

⑥ : 예약어라서 사용할 수 없음

⑧ : 허용하지 않는 특수문자 '@'를 사용할 수 없음

Variable

■ 다음 중 변수를 잘못 초기화 한 것은? (모두 고르시오)

- ① byte b = 256;
- ② char c = '';
- ③ char answer = 'no';
- ④ float f = 3.14
- ⑤ double d = 1.4e3f;


- ① : byte의 범위(-128~127)를 넘는 값으로 초기화 할 수 없음
- ② : char는 반드시 한 개의 문자를 지정해야 함
- ③ : char에 두 개의 문자를 저장할 수 없음.
- ④ : 3.14는 3.14d의 생략된 형태. 접미사f를 붙이거나 형변환 필요
- ⑤ : double(8byte)에 float값(4byte)을 넣는 것이므로 OK

접미사가 있는 자료형은 세 개의 자료(long, float, double)이며, 접미사는 대소문자를 구별하지 않는다.

double은 접미사를 생략할 수 있으므로 float 리터럴에는 반드시 접미사를 붙여야 함

Variable

- 자동차 관리 프로그램을 개발하려고 한다. 주어진 변수의 자료형에 대한 설명으로 적절한 것은?

	항목	변수명	자료의 예
	차종	car	소형승용차
	연비	mileage	23.4
	배기량	displacement	1300

- ① 차종의 변수 car는 실수형으로 선언한다
- ② 연비의 변수 mileage는 실수형으로 선언한다.
- ③ 배기량의 변수 displacement는 정수형으로 선언한다

Variable


- 성적처리 프로그램을 개발하려고 한다. 주어진 데이터의 자료형으로 적절한 것은?

이름	정보기술기초	컴퓨터일반	프로그래밍	총점	평균
강정수	85	76	87	248	82.67
이지명	72	80	75	227	75.67
최희선	88	95	98	281	93.67

- ① 이름은 실수형으로 선언한다.
- ② 총점은 과목 점수의 합이므로 정수형으로 선언한다.
- ③ 평균은 소수점까지 나타내기 위하여 실수형으로 선언한다

Variable

- 망원경 관리 프로그램을 작성하고자 한다. 변수 선언과 사용 예로 가장 적절한 것은?

	항목	변수명	자료의 예
	품명	item	천체망원경
	수량	num	2
	렌즈	lens	1.5
	삼각대	tripod	알루미늄
	직경	diam	50

Variable

- 망원경 관리 프로그램을 작성하고자 한다. 변수 선언과 사용 예로 가장 적절한 것은?

① `int item;`
`item = "천체망원경";`

② `float num;`
`num = "2";`

③ `char lens;`
`lens = 1.5;`

④ `double tripod;`
`tripod = "알루미늄";`

⑤ `int diam;`
`diam = 50;`

Variable

- 동물의 정보를 관리하는 프로그램을 개발하려 한다. 변수 선언과 초기값을 대입하여라.



자료명	변수명	자료의 예
이름	Name	메리
나이	Age	3
몸무게	Weight	2.7
종	Species	치와와
출산횟수	Birth_R	2

Variable

- 그림과 같이 학생신상정보를 관리하는 프로그램을 개발하려고 한다. 이를 위한 변수 선언과 초기값을 대입하여라

학생신상정보

학생신상정보 입력 화면

이름(name)	김철수
번호(number)	10
자격증보유(license)	예
나이(age)	19
전화번호(phone)	123-4567

Variable

- 엘리베이터 제어 프로그램을 개발하고자 한다. 변수 선언 및 초기화하세요


입력 자료	변수명	자료 값
제작회사	company	우리회사
정원(명)	persons	20
승객용(Yes/No)	passenger	Yes
소비전력 (kW/h)	power	310.34
정격하중 (kg)	weight	1350

Expression과 Statement

- Expression
- Statement
- 의미 있는 문장


변수에 상수 대입

`a = 10;`




변수에 변수 대입

`b = a;`



변수에 수식 대입

`c = a + 20;`



문장 예제 1

■ 다음 중 main 메소드의 선언부로 알맞은 것은? (모두 고르시오)

- ① `public static void main(String[] args)`
- ② `public static void main(String args[])`
- ③ `public static void main(String[] arv)`
// 매개변수 args의 이름은 달라도 됨
- ④ `public void static main(String[] args)`
// void는 반드시 main앞에 와야 함
- ⑤ `static public void main(String[] args)`
// public과 static은 위치가 바뀌어도 됨

①, ②, ③, ⑤

배열을 의미하는 기호인 '['는 타입 뒤에 붙여도 되고 변수명 뒤에 붙여도 되기 때문에 'String[] args'와 'String args[]'는 같은 뜻

문장 예제 2

- 다음의 문장에서 리터럴, 변수, 상수, 키워드를 적으시오.

```
int i = 100;  
long l = 100L;  
final float PI = 3.14f;
```

- 리터럴 :
- 변수 :
- 키워드 :
- 상수 :

문장 예제 3

■ 다음 중 타입과 기본값이 잘못 연결된 것은? (모두 고르시오)

① boolean – false

② char – 'Wu0000'

③ float – 0.0

④ int – 0

⑤ long – 0

⑥ String – ""

③ : float는 0.0f가 기본값. 0.0은 0.0d에서 접미사 d가 생략된 것

⑤ : long은 0L이 기본값

⑥ : String은 참조형 타입. 모든 참조형 타입의 기본값은 null

리터럴의 접미사는 대소문자를 구분하지 않으므로, long의 경우 'L'또는 'l'(소문자)을 사용할 수 있음. 'l'은 숫자 '1'과 혼동하기 쉬우므로 대문자를 사용하는 것이 좋음

문장 예제 4

- 다음을 보고 선언문으로 작성하여라

자료형	기본값
boolean	false
char	'\u0000'
byte	0
short	0
int	0
long	0L
float	0.0f
double	0.0d 또는 0.0
참조형 변수	null

문장 예제 4

- 다음을 보고 선언문으로 작성하여라

```
boolean flag = false;  
char grade = ' '; // 공백  
byte b = 0;  
short s = 0;  
int i = 0;  
long l = 0;        // 0L로 자동변환  
float f = 0;        // 0.0f로 자동변환  
double d = 0;       // 0.0로 자동변환  
String s1 = null;  
String s2 = "";    // 빈 문자열
```


문장 예제 5

- 변수 a ~ e 중 대입한 자료가 손실 또는 오류 없이 정확하게 저장되는 것은 ?

```
char a, b;  
int c, d;  
float e;  
a = "A";  
b = 100;  
c = 6000;  
d = 3.14159;  
e = 3.5E-15;
```

문장 예제 6

■ 다음 중 올바른 대입문을 고른 것은?

① `total1 + total2 = 100;`

② `2004 = year + 1;`

③ `score = 1000;`

④ `total = total + sum;`

문장 예제 7

■ 다음 문장에서의 에러를 찾아보고 올바르게 수정하십시오.

- ① `char a = "a";`
- ② `int money = 2,300;`
- ③ `double float phi = 3.14;`
- ④ `printf("%d", 'A');` //문자 A 를 출력

문장 예제 8

- 잘못된 문장을 골라내고 그 이유를 설명하시오.

```
public static void main(String[] args) {  
    char ch0 = 'a';           ①  
    char ch1 = "a";           ②  
    String string0 = "a";     ③  
    String string1 = 'a';     ④  
    char char2 = 'ab';        ⑤  
    String string2 = "ab";    ⑥  
}
```

- ② : char 변수에는 문자열 상수를 저장할 수 없음
- ④ : String 변수에는 문자 상수를 저장할 수 없음
- ⑤ : 문자 상수는 문자 한 개만 단일 따옴표로 묶을 수 있음

문장 예제 9

- 잘못된 문장을 골라내고 그 이유를 설명하시오.

```
public static void main(String[] args) {  
    char ch1 = 'A';           ①  
    char ch2 = 'AB';          ②  
    char ch3 = "";            ③  
    String s1 = "AB";         ④  
    String s2 = "";           ⑤  
    String s3 = "A" + "B";    ⑥  
    String s4 = 'A';          ⑦  
}
```

- ② : char는 둘 이상의 문자를 저장할 수 없다
- ③ : char는 반드시 한 문자를 지정해야 함
- ⑦ : String은 “”를 사용해야 함

문장 예제 10

- 잘못된 문장을 골라내고 그 이유를 설명하시오.

```
public static void main(String[] args) {  
    byte var1 = 128;           ①  
    short var2 = 128;          ②  
    int var3 = 28L;             ③  
    long var4 = 128L;           ④  
    float var5 = 123456.789123; ⑤  
    double var6 = 123456.789123; ⑥  
}
```

- ① : byte 이기에 128은 저장할 수 없음
- ③ : int 형은 long 형 상수 값을 저장할 수 없음
- ⑤ : float 형은 유효숫자 부가 긴 값을 저장할 수 없음

문장 예제 11

- 잘못된 문장을 고쳐서 에러가 발생하지 않도록 수정하시오

```
public static void main(String[] args) {  
    byte var1 = 127;  
    short var2 = 128;  
    int var3 = 128;  
    long var4 = 128L;  
    var4 = var1;  
    System.out.println(var1 + ", " + var2);  
    var1 = (byte) var3;  
    System.out.println(var1 + ", " + var3);  
}
```

문장 예제 12

- 잘못된 문장을 고쳐서 에러가 발생하지 않도록 수정하시오

```
public static void main(String[] args) {  
    float var5 = 123456.789123f;  
    double var6 = 123456.789123;  
    var5 = (float) var6;  
    System.out.println(var5+ ", " + var6);  
    var6 = var5;  
    System.out.println(var5+ ", " + var6);  
}
```


int = integer
char = character

자료형 (Data Types)

데이터 타입	예	키워드	읽는법	변환 사양자
부동 소수점 형 (float)	3.14f	float	플로트	%f
배 정도 부동 소수점 형 (double float)	3.14	double	더블	%f
10진 정수(decimal)	3	short	쇼트	%d
	3	int	인트	%d
	3L	long	롱	%d
2진 정수(binary)	0b11	int	인트	
8진 정수(octal)	07			%o
16진 정수(hexadecimal)	0x07			%x
문자(character)	'A'	char	캐릭터, 차	%c
문자열(string)	"A"	String	스트림	%s



문장 예제 13

- 문자열과 덧셈 연산을 하면 그 결과는 항상 문자열

문자열 + any type → 문자열 + 문자열 → 문자열
any type + 문자열 → 문자열 + 문자열 → 문자열

- ① "" + 7 + 7 → "7" + 7 → "77"
- ② 7 + "" → "7" + "" → "7"
- ③ 7 + " " → "7" + " " → "7 "
- ④ " " + 7 → " " + "7" → " 7"
- ⑤ 7 + "7" → "7" + "7" → "77"
- ⑥ 7 + 7 + "" → 14 + "" → "14" + "" → "14"
- ⑦ " " + 7.5 → " " + "7.5" → " 7.7"
- ⑧ true + "" + 7 → "true" + "" + 7 → "true" + "7" → "true7"
- ⑨ null + "" + 7 → "null" + "" + 7 → "null" + "7" → "null7"

문장 예제 14

- 다음 중 형 변환을 생략할 수 있는 것은? (모두 고르시오)

```
byte b = 10;  
char ch = 'A';  
int i = 100;  
long l = 1000L;
```

- ① b = (byte) i; ② ch = (char) b;
③ short s = (short) ch; ④ float f = (float) l;
⑤ i = (int) ch;

- ① : int(4 byte) → byte(1 byte)이므로 반드시 형변환 필요
② : byte(1 byte) → char(2 byte)이지만 범위가 달라서 형변환 필요
③ : char, short은 2 byte이지만 범위가 달라서 형변환 필요
④ : float(4 byte)의 범위가 long(8 byte)보다 커서 생략 가능
⑤ : char(2 byte) → int(4 byte)이므로 생략 가능

문장 예제 15

- 다음 프로그램을 컴파일 했을 때 결과를 추측해보시오.

```
public static void main(String[] args) {  
    byte b = 127;  
    byte c = 126;  
    byte d = b + c;  
  
    System.out.println("b " + b);  
    System.out.println("c " + c);  
    System.out.println("d " + d);  
}
```

+ 연산을 하게 되면 int 형으로 자동 형변환되는데 int형을 byte형으로 변환하지 못하기에 컴파일에러가 발생

문장 예제 16

■ 다음 프로그램의 결과를 설명하여라.

```
public static void main(String[] args) {  
    char ch1, ch2;  
    ch1 = '1';  
    ch2 = 1;  
  
    System.out.printf("ch1 = %c, ch1 = %d\n", ch1, ch1);  
    System.out.printf("ch2 = %c, ch2 = %d\n", ch2, ch2);  
}
```

문장 예제 17

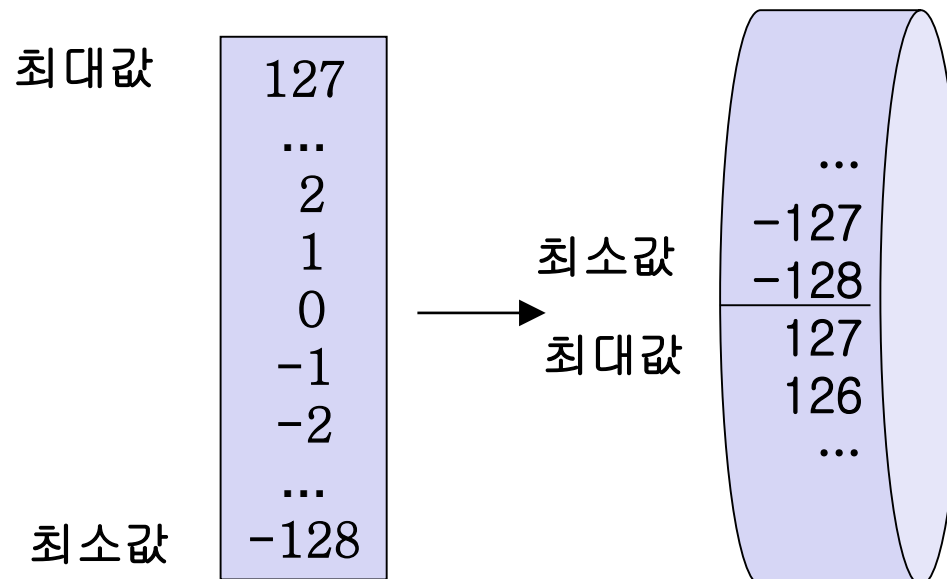
- 다음 프로그램의 결과를 설명하여라.

```
public static void main(String[] args) {  
    byte StartNumber = 120;  
    byte SmallNumber;  
    SmallNumber = StartNumber;  
  
    for (int index = 0; index < 16; index++) {  
        System.out.println("StartNumber(" + StartNumber + ") +  
                            " + LargeNumber(" + index + ") = " +  
                                SmallNumber++);  
    }  
}
```

문장 예제 18

- 길이를 초과하는 값을 저장할 때 처리되는 결과
- Overflow 발생
 - 결과 -128의 값이 출력

```
byte b = 128;           //에러  
byte b = 127;  
b = b + 1;              // b에 저장된 값을 1증가
```



문장 예제 19

```
public static void main(String[] args) {  
    int num = 10; //10진수  
    int bnum = 0B1010; //2진수  
    int onum = 012; //8진수  
    int xnum = 0XA; //16진수  
  
    System.out.printf("num = %d\n", num);  
    System.out.printf("bnum = %d\n", bnum);  
    System.out.printf("onum = %o\n", onum);  
    System.out.printf("xnum = %x\n", xnum);  
}
```

10진수는 평소 쓰던 숫자처럼 입력해주고 2진수는 0B를 숫자 앞에 붙여주고, 8진수는 0을 숫자 앞에 붙여주고 16진수는 0X를 숫자 앞에 붙여 주면 됨

Program 연습 0

- 다음과 같이 출력되는 Program을 작성하세요.



- 변화할 수 있는 부분은 ?

Program 연습 0

■ 상수를 이용하는 방법

[illegible]

Program 연습 0

■ 변수를 이용하는 방법

```
public class Sample3 {  
    public static void main(String[] args) {
```

Program 연습 1

- 다음과 같이 출력하는 프로그램을 작성하여라



- 1) 출력 내용에서 변수를 찾아보아라
- 2) 변수의 형을 이야기 해보자

Program 연습 1

```
public class DataTest1 {  
    public static void main(String[] args) {  
        System.out.printf("사과의 종류는 %s이고, 개수는 %d개 입니다.\n",  
                           "부사", 4);  
        System.out.printf("%s님의 키는 %.1f Cm 입니다.\n",  
                           "홍길동", 170.5);  
    }  
}
```

int 정수형 상수

문자열 상수

double 실수형 상수

Program 연습 2

- 다음과 같이 출력되는 Program을 작성하여라



Program 연습 2

- 다음과 같이 출력되는 Program을 작성하여라

```
public static void main(String[] args) {  
    char c;  
  
    c = 'a';  
    System.out.println(c);  
    c = '가';  
    System.out.println(c);  
  
    String s = "Hello World!";  
    System.out.println(s);  
}
```

프로그램 연습 3

- 다음과 같이 출력 하는 Program을 작성하여라



1) Data Type을 구분하여 보자

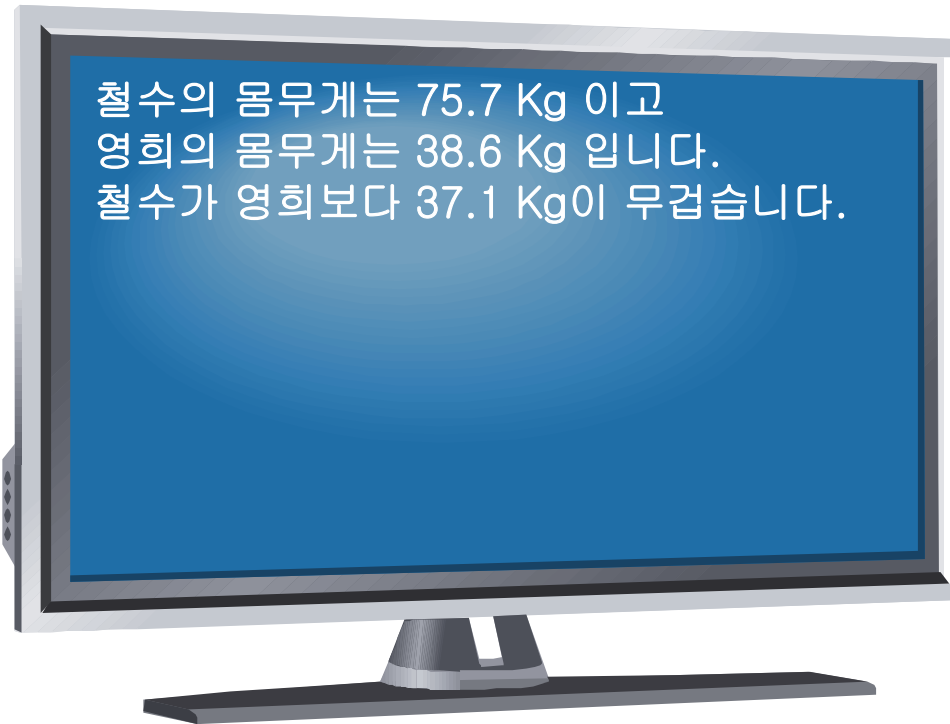
Program 연습 4

- 다음과 같이 출력 하는 Program을 작성하여라



Program 연습 5

- 다음과 같이 출력하는 Program을 작성하여라.



- 1) Data를 상수로 표현 하여라
- 2) Data를 변수로 표현 하여라
- 3) 무겁다, 가볍다는 정보이다

Program 연습 6

- 다음과 같이 출력하는 Program을 작성하여라.



Program 연습 7

- 다음 Program의 Error의 원인은 무엇인가요?

```
public static void main(String[] args) {  
    int num;  
    num = 17.5;  
    System.out.println(num);  
}
```

변수의 데이터형과 상수(리터럴)의 데이터 타입은
일치하여야 함

Program 연습 8

- 다음 Program의 Error의 원인은 무엇인가요?

```
public static void main(String args[]) {  
    int num;  
  
    System.out.println(num);  
}
```

지역 변수에 아무 값도 초기화하지 않고 사용하는 것은 오류이다

Program 연습 9

- 다음 Program의 Error의 원인은 무엇인가요?

```
public static void main(String[] args) {  
    System.out.println(num1);  
    double num2 = 12.75;           // num2 변수의 선언문  
    System.out.println(num2);  
    char ch = 'A';                 // ch 변수의 선언문  
    System.out.println(ch);  
    short num1 = 12;               // num1 변수의 선언문  
}
```

지역 변수의 선언문은 메소드 내부에서 사용되기 전에
있어야 된다

Program 연습 10

- 다음 Program의 Error의 원인은 무엇인가요?

```
public static void main(String[] args) {  
    {  
        int num = 10;  
    }  
    System.out.println(num);  
}
```

블록 안에 선언된 변수는 블록 밖에서 사용할 수 없다

Program 연습 11

- 다음 Program의 Error의 원인은 무엇인가요?

```
public static void main(String[] args) {  
    int num = 10;  
    {  
        int num = 30;  
    }  
    System.out.println(num);  
}
```

블록 밖에 선언된 변수를 블록 안에서 사용하는 것은 가능
하다

도입 실습

- 자동차 연비를 계산하여라.



연비란 연료 1L로 주행 가능한 거리를 말하는 것

도입 실습

- 단계 0> 문제 영역(도메인 지식) 분석
 - 자동차 연비란 ?
 - 데이터와 정보를 구분
 - 데이터는 변수와 상수로 구분
 - 변수의 데이터 형(type) 생각
 - 데이터(변수)는 입력 받는다.
 - 정보는 계산한다.

도입 실습

- 단계 1> 문제 분석
- 목표(출력): 연비계산 값을 출력
- 정보(입력): 자동차의 실제 연료 소모량
자동차의 주행 거리
- 계산(처리): 사전 정보로부터 얻은 실제 값을 연비 공식에 대입하여 계산

$$\text{자동차 연비} = \frac{\text{실제 주행 거리(km)}}{\text{실제 연료 소모량(liter)}}$$

도입 실습

■ 단계 2> 프로그램 기본 구조

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

사용할 변수명과 데이터 형 선언

입력 부문

처리 부문

출력 부문

```
}  
}
```

도입 실습

■ 단계 2> 프로그램 기본 구조

사용할 변수 명과 데이터 형 선언	두 개의 값 (거리와 소모량)을 입력으로 하고, 공식에 의해 계산될 결과 연비를 저장해야 하 므로 변수는 모두 3개가 필요, 공식에 나눗셈이 들어 가므로 소수점을 포함하는 실수로 나타내 야 한다.
입력 부문	프롬프트(prompt)를 만들어라. System.out.print(“ 운행 거리 입력 “); Scanner 클래스를 사용한다
처리 부문	연비 계산 공식을 사용
출력 부문	출력 형태 생각

도입 실습

■ 단계 3> 프로그램 작성

```
import java.util.Scanner;  
public class Mileage {  
    public static void main(String[] args) {  
        double km, liter, result;  
        Scanner keyboard = new Scanner(System.in);
```

```
        System.out.print(" 주행 거리 입력 : ");  
        km = keyboard.nextDouble();  
        System.out.print(" 연료 소모량 입력 : ");  
        liter = keyboard.nextDouble();
```

```
        result = km / liter;
```

```
        System.out.printf(" 주행 거리 : %.2f Km\n", km);  
        System.out.printf(" 연료 소모량 : %.2f liter\n", liter);  
        System.out.printf(" 연비 : %.2f \n", result);
```

```
    }
```

```
}
```

도입 실습

■ 단계 4> 프로그램 결과 확인

ASCII Code

- Keyboard에서 한 문자를 입력하면 그 ASCII Code 값을 출력하는 Program을 작성하여라



ASCII Code

- 문제 분석

- Data는 무엇이고, Information는 무엇인가?

- 입력

- 하나의 문자 입력

- Scanner 입력 방법도 있지만, System 클래스의 read() 메소드를 사용해보자

- 출력

- ASCII Code 출력 : %h

- 계산 방법

ASCII Code

```
public static void main(String[] args) {  
    char ch;  
  
    try {  
        System.out.print("문자 입력 : ");  
        ch = (char) System.in.read();  
        System.out.printf("입력 문자 : %c이고, ASCII 코드 : %hWn", ch, ch);  
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
}
```

Arbeit 비용

- 나의 지갑에 돈이 25,000이 있다. 오늘 저녁에는 아르바이트 한 일주일 치 급여를 받는 날이다.
- 급여는 시간당 5,800원이며 하루 5시간 일하고, 일주일간 일했다.
- 자 ! 이제 여러분이 급여를 받으면 지갑에 얼마가 있게 되는가 ?
- Program을 작성하고 결과를 출력하라

Arbeit 비용

- 문제 분석

- Data는 무엇이고, Information는 무엇인가?

- 입력

- 지갑(wallet)에 25,000원

- 출력

- 지갑(wallet)의 금액

- 계산 방법

- Arbeit 비용 = 5시간 * 5800원 * 7일

Arbeit 비용

```
public static void main(String[] args) {  
    int wallet = 25000;  
    int arbiet = 5 * 5800 * 7;  
  
    System.out.println("지갑 : " + wallet);  
    System.out.println("Arbiet : " + arbiet);  
    System.out.println("총 금액 : " + (wallet + arbiet));  
}
```



Futuristic Innovator

京福大學校
KYUNGBOK UNIVERSITY

Light-year

- 빛이 진공에서 1년(365.25일)동안 진행하는 거리의 단위를 말함



Light-year

■ 문제 분석

- Data는 무엇이고, Information는 무엇인가?

■ 입력

- 빛의 속도 : 300,000 Km/S – 정수형 (int)

- 1년 : 365.25일 – 실수형 (double)

■ 출력

- 거리(distance) : 정수형 (long)

■ 계산 방법

- 1일 초 = 24시간 * 60분 * 60초 = 86,400 초

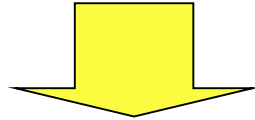
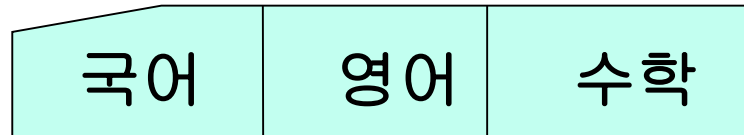
- 거리 = 365.25 * 86,400초 * 300,000 Km

Light-year

```
public static void main(String[] args) {  
    final long LIGHT = 300000;  
    long distance;  
    int day = 24 * 60 * 60;  
    distance = (long) (365.25 * day * LIGHT);  
  
    System.out.printf("빛이 1년동안 가는 거리 : %,d Km\n", distance);  
}
```


성적 계산

- 3과목(국어, 영어, 수학)의 점수를 입력 받아, 총점과 평균을 계산하는 프로그램을 만들어보자



국어	영어	수학	총점	평균점수
----	----	----	----	------

성적 계산

■ 문제 분석

- Data는 무엇이고, Information는 무엇인가?

■ 입력

- 국어(kor) – 정수형 (int 형)
- 영어(eng) – 정수형 (int 형)
- 수학(math) – 정수형 (int 형)

■ 출력

- 총점(total) – 정수형 (int 형)
- 평균(avg) – 실수형 (double)

■ 계산 방법

- 총점 = 국어 + 영어 + 수학
- 평균 = 총점 / 3

성적 계산

```
public class Main {  
    public static void main(String[] args) {  
        Scanner keyboard = new Scanner(System.in);  
        int kor;  
        int eng;  
        int math;  
        int total;  
        double avg;  
  
        System.out.print("국어 성적 입력 : ");  
        kor = keyboard.nextInt();  
        System.out.print("영어 성적 입력 : ");  
        eng = keyboard.nextInt();  
        System.out.print("수학 성적 입력 : ");  
        math = keyboard.nextInt();  
    }  
}
```

성적 계산

```
total = kor + eng + math;  
avg = total / 3.0;
```

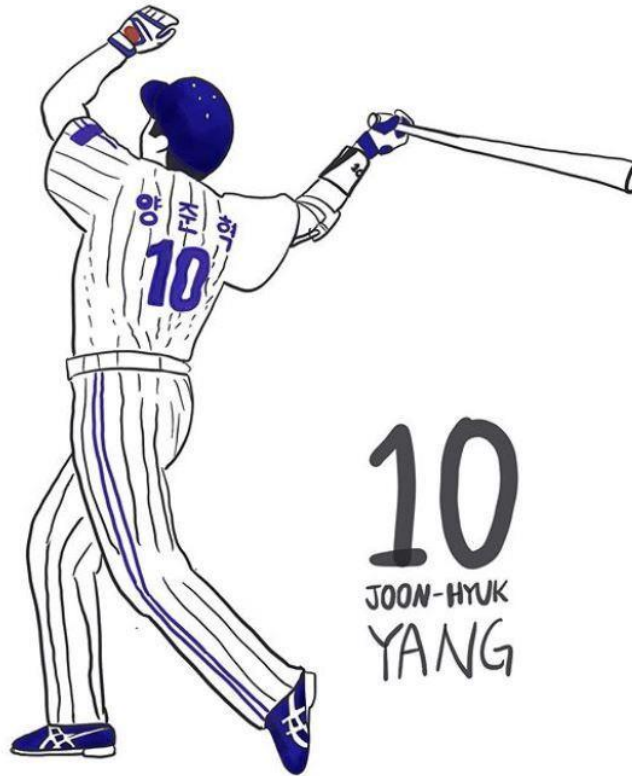
```
System.out.println("국어 점수 : " + kor);  
System.out.println("영어 점수 : " + eng);  
System.out.println("수학 점수 : " + math);  
System.out.println("총점 : " + total);  
System.out.printf("평균 : %.2f", avg)
```

```
}
```

```
}
```

Batting Average

■ 야구 선수들의 타율을 계산해보자



Batting Average

■ 문제 분석

- Data는 무엇이고, Information는 무엇인가?
 - 문제에서 Data와 정보가 명확하지 않음
 - 상식을 동원해서 찾아야 함

■ 입력

- 선수 이름(name) : 문자열 (String)
- 오늘의 타수(bat) : 정수형 (int)
- 오늘의 안타(hit) : 정수형 (int)
 - 타수를 넘을 수 없음

■ 출력

- 타율(avg) : 실수형 (double)

■ 계산 방법

- 타율 = 안타수 / 타수

Batting Average

```
public static void main(String[] args) {  
    Scanner keyboard = new Scanner(System.in);  
    String name;  
    int bat;  
    int hit;  
    double avg;  
  
    System.out.print("선수 이름 입력 : ");  
    name = keyboard.nextLine();  
    System.out.print("오늘의 타수 : ");  
    bat = keyboard.nextInt();  
    System.out.print("오늘의 안타수 : ");  
    hit = keyboard.nextInt();  
  
    avg = (double) hit / bat;
```



Batting Average

```
System.out.printf("선수 이름 : %s\n", name);  
System.out.printf("오늘의 타수 : %d\n", bat);  
System.out.printf("오늘의 안타수 : %d\n", hit);  
System.out.printf("오늘의 타수 : %.2f\n", avg);  
}
```


큰 수 더하기

- 34억과 4억을 더하는 Program을 작성하여라

```
public static void main(String[] args) {  
    long value1 = 3400000000L;  
    int value2 = 400000000;  
  
    long result = value1 + value2;  
  
    System.out.printf("%,d + %,d = %,d\n", value1, value2, result);  
}
```

BigInteger

- JAVA의 64비트 정수형인 long 형으로도 겨우 9,223,372,036,854,775,807 (구백이십이경...)까지의 숫자만 나타낼 수 있음
 - 그 이상의 정수가 필요하다면 BigInteger 클래스를 사용해야 함
- BigInteger는 숫자의 크기에 제한이 없다. 사실상 무한대의 정수를 저장할 수 있음
 - BigInteger는 기본 자료형과 달리 사칙연산이 $+$ $-$ $/$ $*$ 이런 기호로 되지는 않고, 다음과 같은 메소드를 사용하여 연산을 해야 함

더하기 : add()
빼기 : subtract()
곱하기 : multiply()
나누기 : divide()

BigDecimal

- BigInteger는 정수를 다루지만, BigDecimal은 실수
- Computer의 특성으로 인해 float이나 double은 실수 연산 시 정확한 결과가 나오지 않음
- BigDecimal을 이용하면 계산은 조금 번거롭지만 정확한 결과를 볼 수 있음

Integer와 int의 차이

■ Primitive 자료형 - Wrapper 클래스 관계

int	primitive 자료형 (long, float, double ...)
	산술 연산이 가능
	null로 초기화 할 수 없음
Integer	Wrapper 클래스 (객체)
	Unboxing을 하지 않으면 산술 연산이 불가능 하지만, null 값을 처리할 수 있음
	null 값 처리가 용이하기 때문에 SQL과 연동할 경우 처리가 용이함
	DB에서 자료형이 정수형이지만 null 값이 필요한 경우 VO에서 Integer를 사용할 수 있음