

능동적 사고 방식의

java

강사 박주병

# Part01 변수

01 강사 및 강의소개

02 개발환경 구축

03 변수의 타입

04 기본 자료형과 참조형

# 01

## 강사 및 강의 소개



# PROFILE

박주병

- 1.소방,해경 구조관제시스템 고도화
- 2.서울소방 관제시스템 고도화
- 3.소방 ARS 관리 시스템 신규 개발
- 4.신고번호통합 비상대응시스템 신규 개발
- 5.곽병원 OCS,NMS,ERP개발

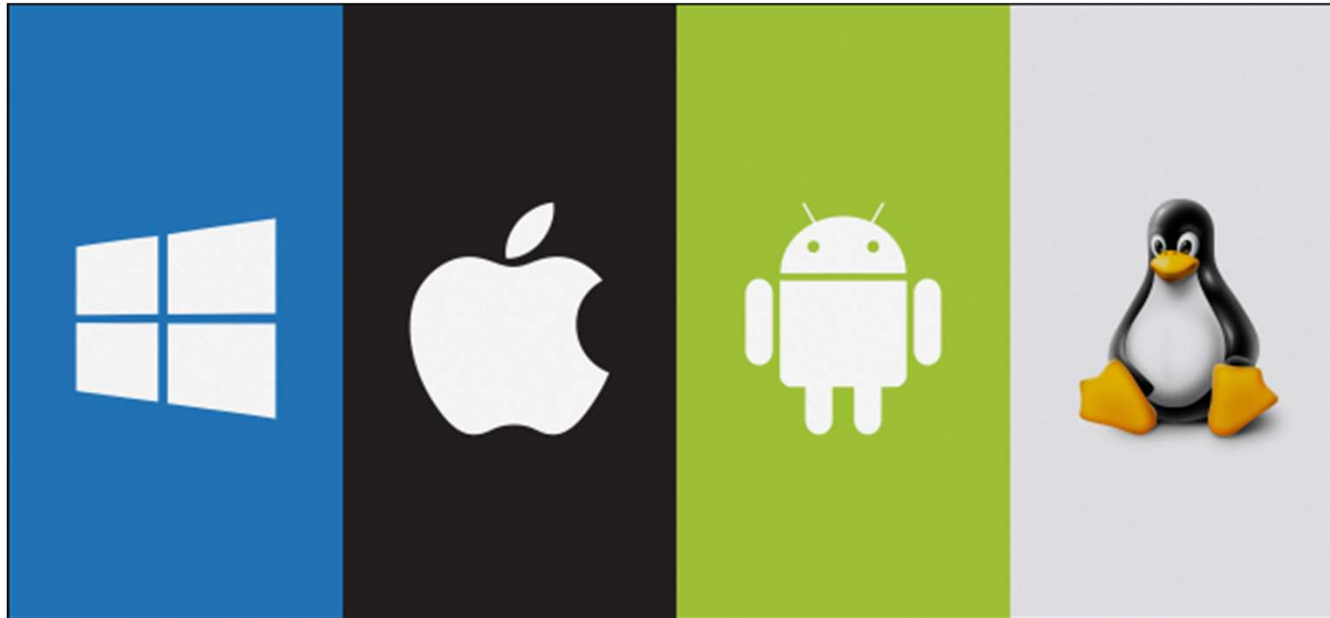
?

# Java란?

- 1.플랫폼 독립적 언어
- 2.객체지향 언어
- 3.C++의 어려운점은 감추다
- 4.현재 가장 많이 쓰이는 언어



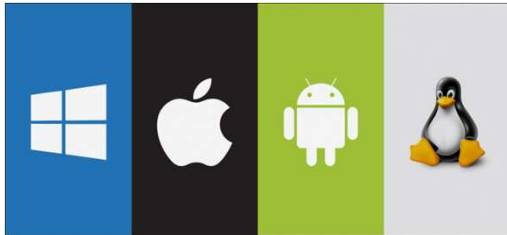
# 플랫폼



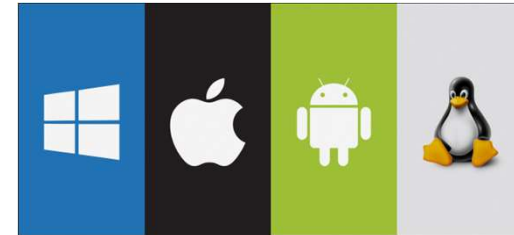
JAVA 프로그램



JVM(java virtual machine)



일반 프로그램





## JVM(java virtual machine)

- 1.속도가 상대적으로 느리나 최근엔 컴파일러 개선 및 최적화로 개선되었다.
- 2.Class 파일을 실행한다.
- 3.가비지 컬렉션으로 메모리 관리를 한다.

## 자바 개발도구 (JDK)

1. java.exe
2. Javac.exe
3. Javap.exe
4. Jar.exe
5. JRE (JVM+API)

컴파일러(javac.exe)

.java    >>>    .class

실행(java.exe)

>>>    바이너리파일

# — 02

개발환경 구축

## N | 이클립스 다운로드

통합 VIEW 이미지 지식iN 인플루언서 동영상 쇼핑 뉴스 여학사전

download.eclipse.org

Eclipse Downloads | The Eclipse Foundation

The Eclipse Foundation - home to a global community, the Eclipse IDE, Jakarta EE and over 415 c projects, including runtimes, tools and frameworks.



The Eclipse Installer 2022-12 R now includes a JRE for macOS, Windows and Linux.



## Get Eclipse IDE 2022-12

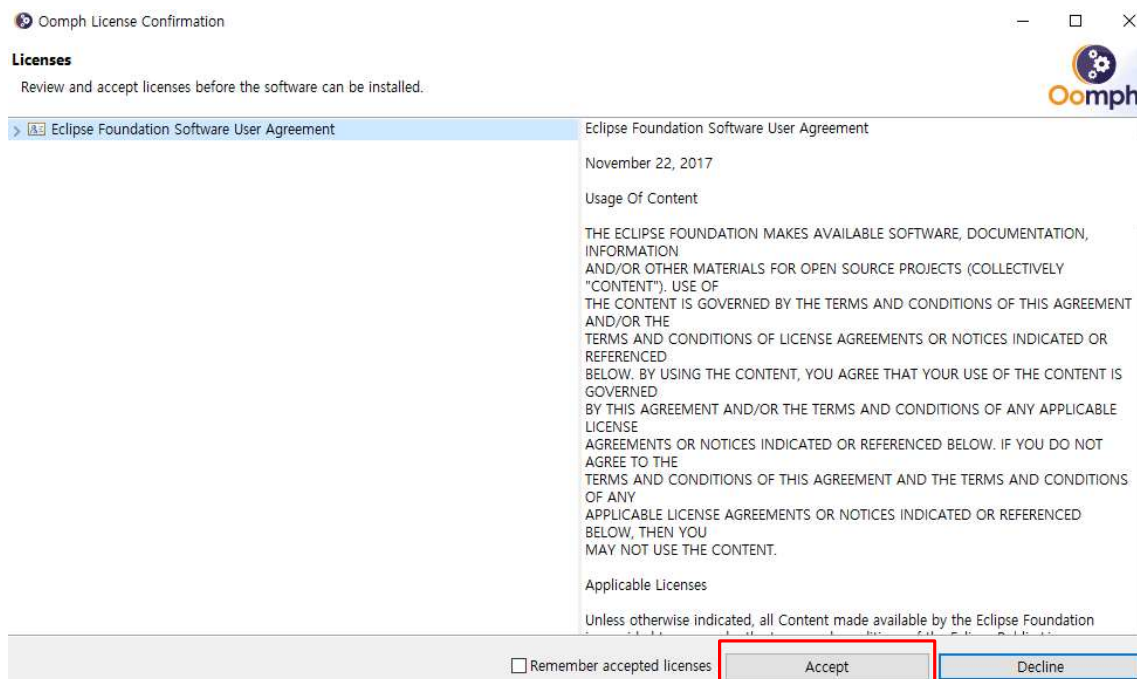
Install your favorite desktop IDE packages.

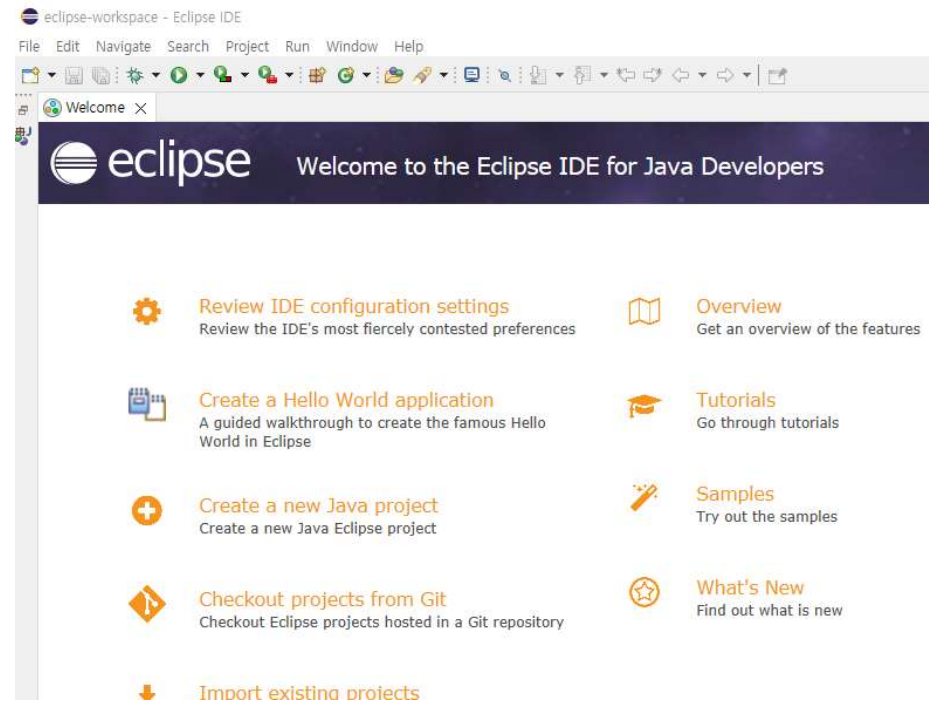
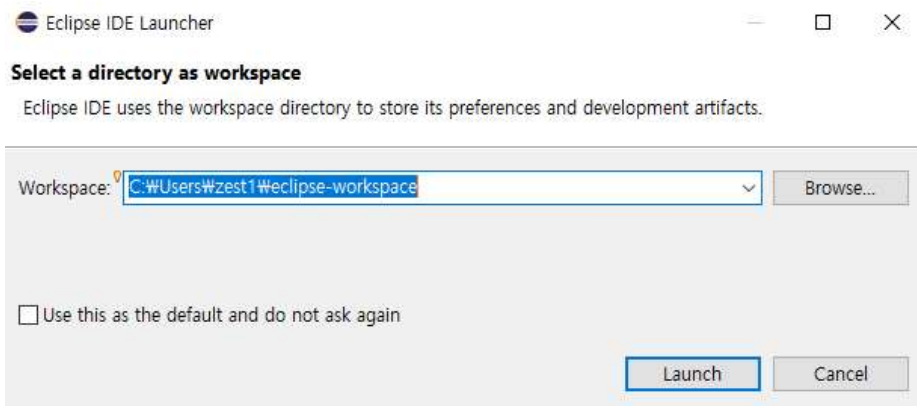
Download x86\_64

Download Packages | Need Help?

ParkJuBye

</

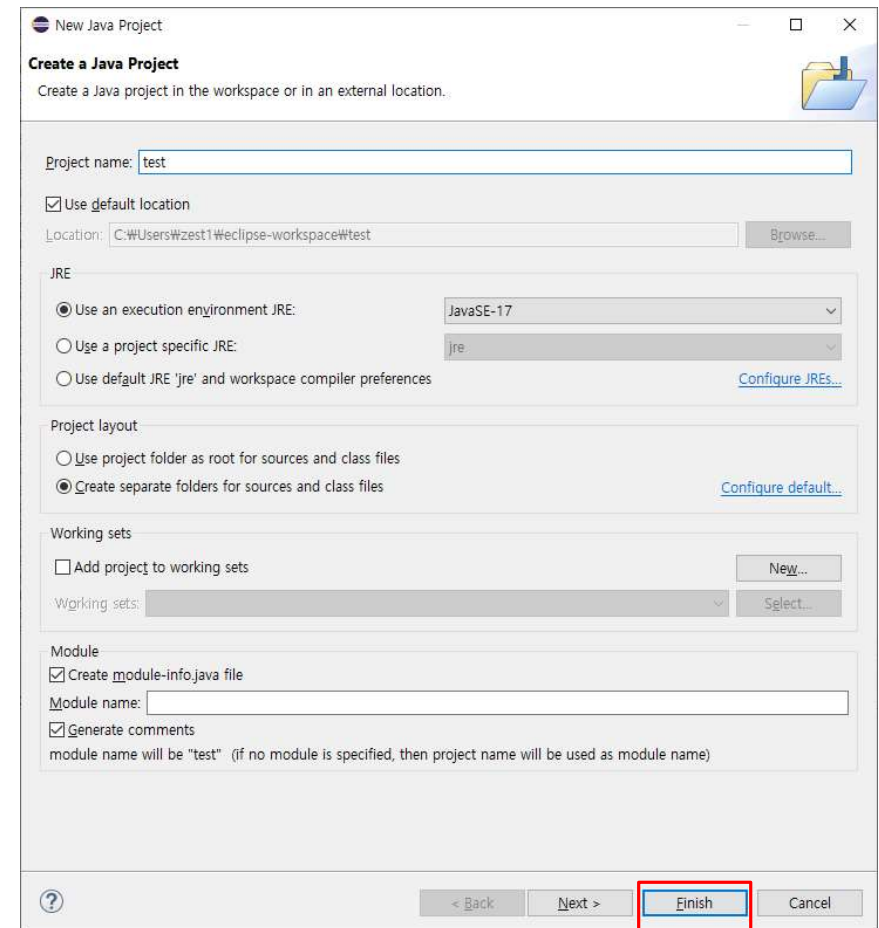
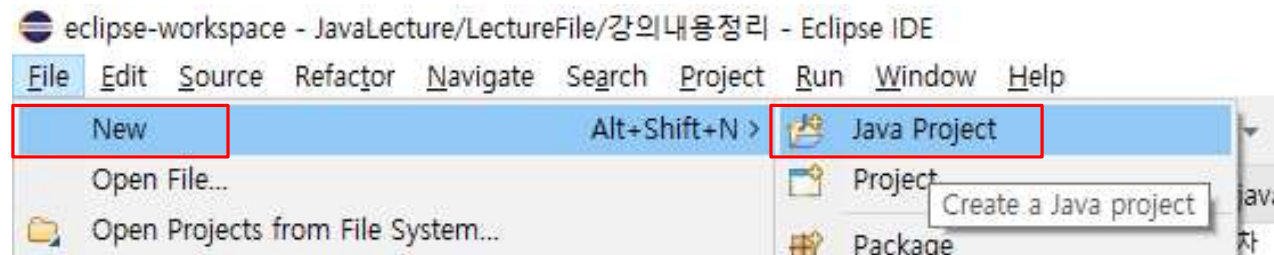


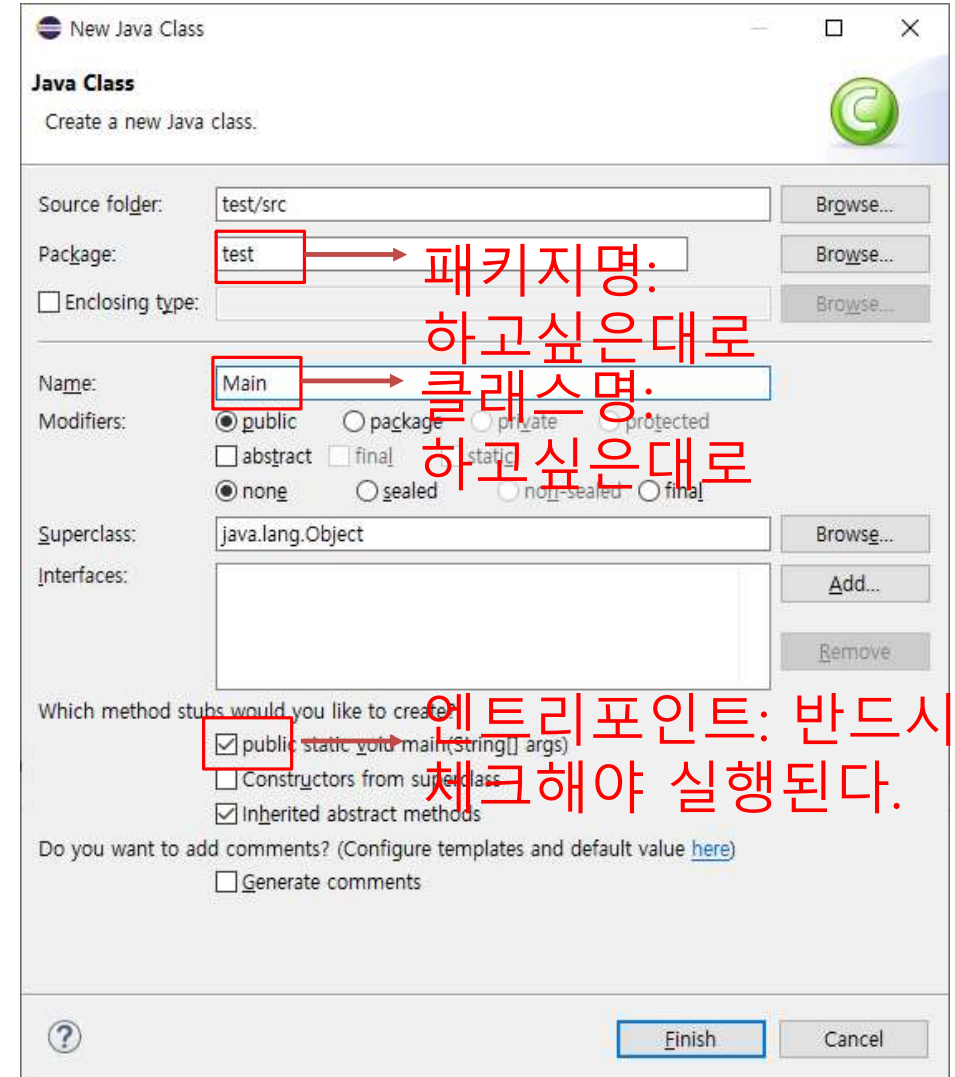
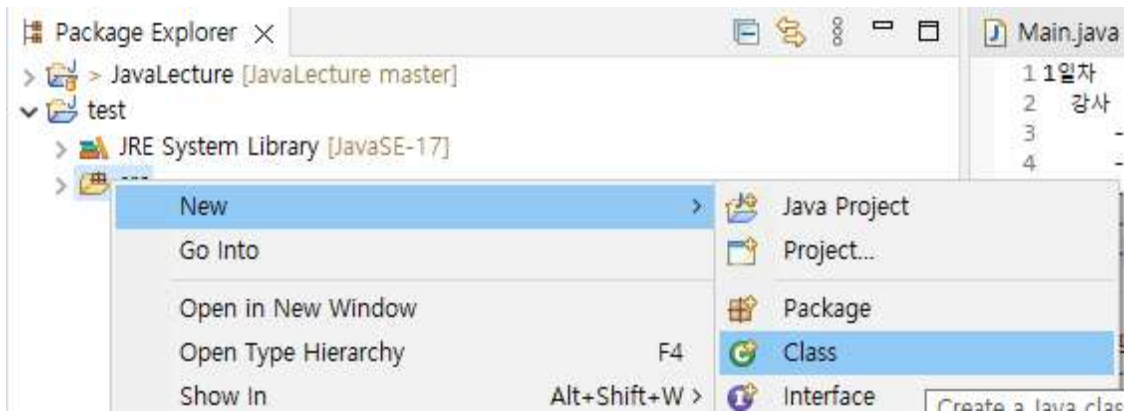




# 실습

Hello World! 를 출력하는 프로그램을 만들어 보자





```
1 package joo;
2
3 public class Main {
4
5     public static void main(String[] args) {
6
7         System.out.println("Hello World!");
8
9
10
11     }
12
13 }
14
```



실행(F11)



```
1 package joo;
2
3 public class Main {
4
5     public static void main(String[] args) {
6
7         System.out.println("Hello World!");
8
9     }
10
11 }
12
13
14
```

패키지명: 클래스를 분류하는 폴더 같은 역할이다.

클래스명: 객체지향언어의 핵심(파일명과 동일해야함)

엔트리포인트: 프로그램의 시작  
지점

모니터로 데이터를 표시해주는 표준 입출력 함수

# 03

## 변수의 타입

$$Y = x + 1$$

변수 : 하나의 값을 저장하는  
공간

```
int age;
```

변수타입

변수명

세미콜론: 문장의 끝

```
int age = 20;
```

대입연산자

값

age 20



```
int a, b, c ;  
int d=0, e=10, f=13;
```

한번에 여러 개의 변수 선언 및 초기화도 가능하다.

```
int t;  
t=30;
```

선언과 초기화를 따로 해줘도 된다.

```
int a=10;  
System.out.println(a);
```

변수에 담긴 값을 화면에 출력할수 있다.

# 실습

age 변수를 선언하여 나이를 저장후

System.out.println();

을 활용하여 나이를 출력해보자.

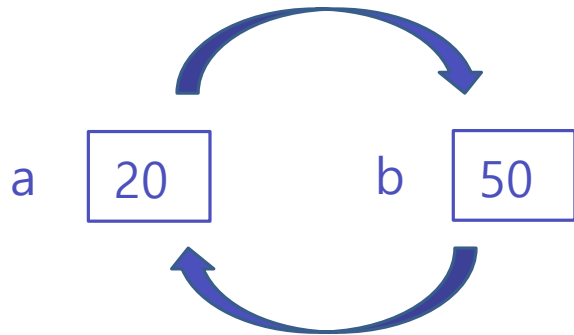
```
int a=10;  
  
int b= a+30;  
  
System.out.println(b);
```

1. a의 값을 읽어온다.
2. 10과 30을 더한다
3. 더한결과를 b에 저장한다.

# 실습문제

a , b 변수를 선언하고 서로의 값을  
바꾼뒤 출력하여 보자

\*변수는 추가로 더 선언하여도 된다.



```
int a=10,b=50, temp;
```

```
temp = a;
```

```
a=b;
```

```
b=temp;
```

```
System.out.println(a);
```

```
System.out.println(b);
```

```
System.out.println("a: " + a + " b: " + b);
```

+ 연산자를 이용해 한번에 붙여서 출력 가능하다.

## 변수명 규칙

1. 대소문자가 구분된다.
2. 자바에서 미리 지정한 예약어는 사용할 수 없다. (예약어 : 문법적 의미를 지닌 단어들)
3. 숫자로 시작할 수 없다.
4. 특수문자는 \_ 와 \$ 만 사용 가능하다



## 변수명 권장사항

1. 단어의 첫글자는 대문자로 한다.

Age , Name, NodeList

2. 상수는 모두 대문자로 한다.(상수: 값이 고정된 변수)

PI, MAX, MIN

# 주석

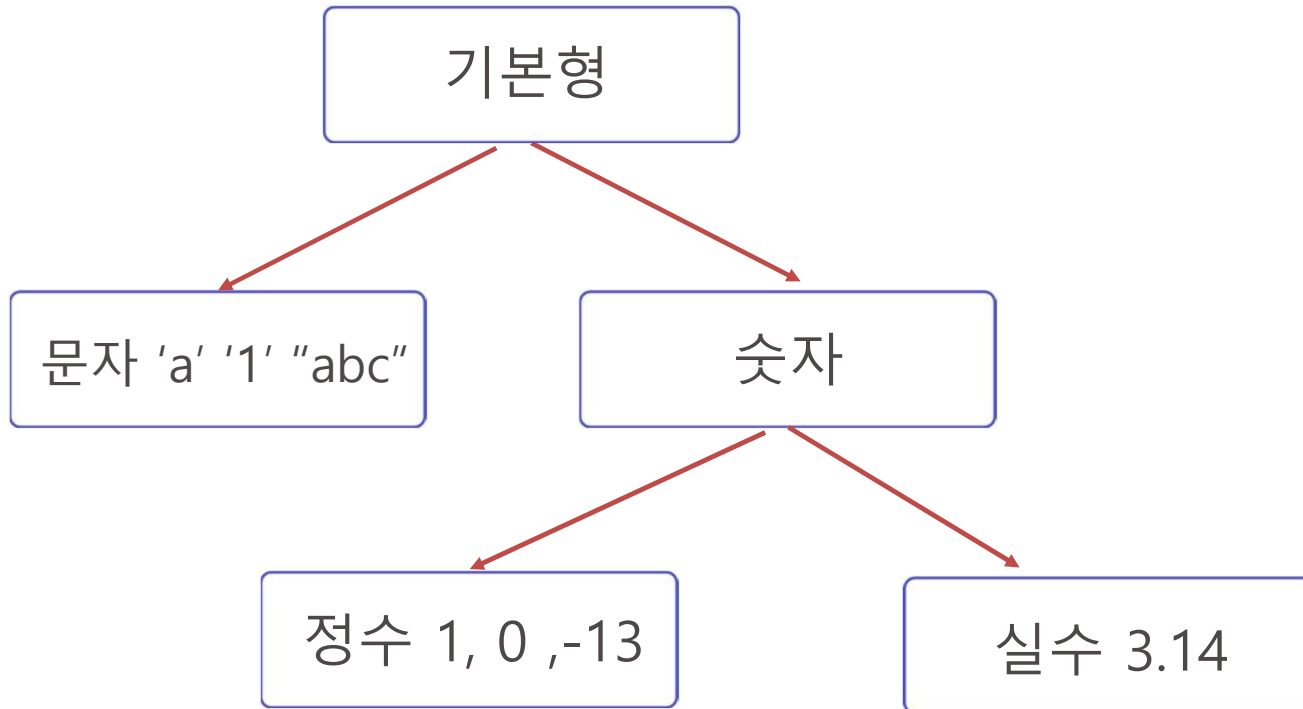
```
1 package joo;
2
3 /*
4  *
5  * Java를 처음 시작하는 예제
6  *
7  */
8 public class Main {
9
10     public static void main(String[] args) {
11
12         int a=10,b=50, temp;
13         |
14         temp = a;
15         a=b;
16         b=temp;
17
18         System.out.println("a: "+a + " b: " + b);
19
20
21
22
23
24     }
25
26 }
```

```
1 package joo;
2
3 //자바를 처음시작하는 예제
4 public class Main {
5
6     public static void main(String[] args) {
7
8         int Age=10; //나이
9
10         //System.out.println("a: "+a + " b: " + b);
11
12
13         System.out.println(Age);
14
15
16 }
```

```
int age ;
```

변수타입

# 변수의 타입



메모리주소	값
0x000A	0x000E
0x000B	0x000K
0x000C	
0x000D	
0x000E	박주병
0x000F	100
0x000G	객체의정보...
0x000H	

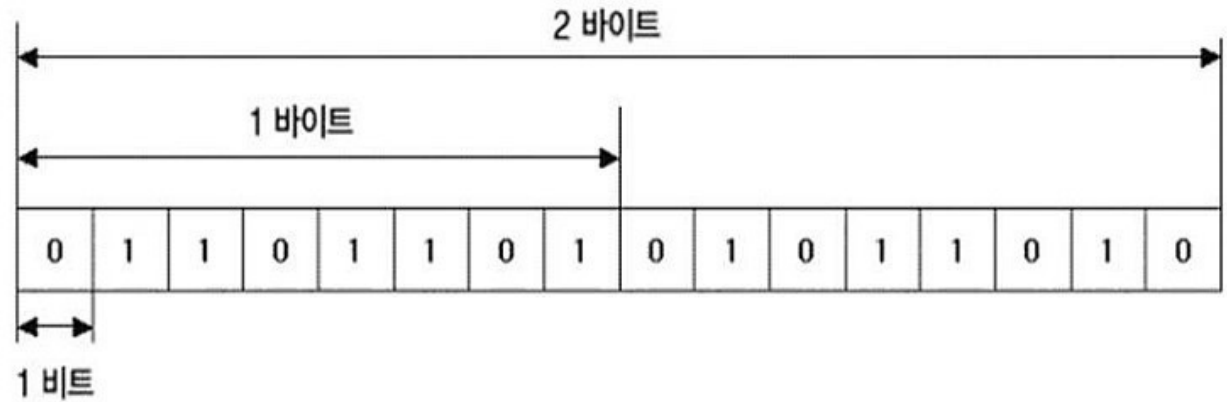
# 기본형 데이터 타입

종류	이름	값
논리형	boolean	True, false
문자형	char	'a', '가', '1'
정수형	byte, short, int, long	12, 0, -3
실수형	float, double	3.14

```
boolean isEqual = true;  
char a = '가';  
int age = 30;  
long age2 = 32;  
double PI = 3.141592;
```

# 데이터 타입별 메모리 크기

데이터타입	크기
Boolean, byte	1 바이트
char, short	2바이트
int , float	4바이트
long, double	8바이트



## 문제

Int 형은 2의 몇승까지 저장할수 있는가?

## 상수와 리터럴

1. 상수는 관례상 대문자로 작성한다.
2. 선언과 동시에 초기화 해야 한다.
3. 한번 값이 정해지면 변경할수 없다.

```
final double PI = 3.14;
```

상수를 만드는  
예약어

상수

리터럴



## 왜 사용 하는걸까?

1. 코드의 가독성이 좋아진다.
2. 값 변경시 한번에 수정할수 있다.

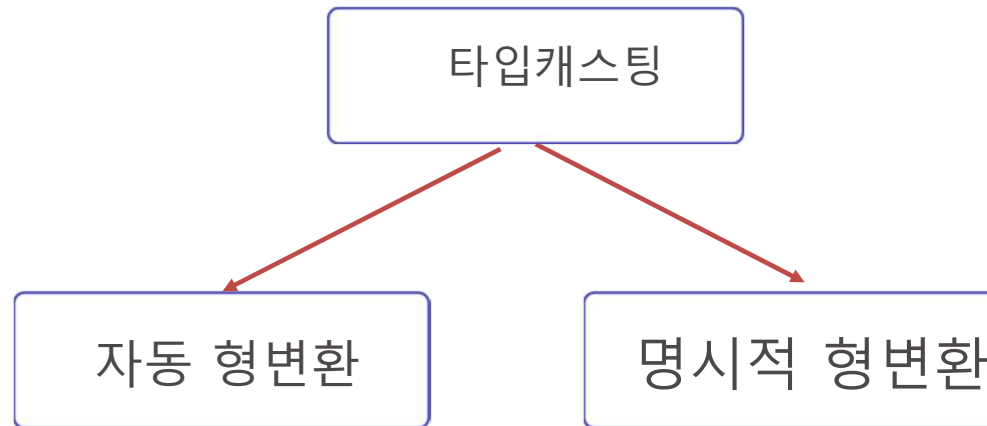
```
final int HEIGHT = 20;  
final int WIDTH = 30;  
  
int rectangle = HEIGHT * WIDTH;  
  
int triangle = 20 * 30 / 2;
```

가능할까?

```
float test = 13;
```

# 형 변환(타입캐스팅)

데이터 타입을 임의로 변경하는것



## 명시적 형변환

```
float test = (float)13;
```

## 자동 형변환

리터럴 상수 13을 자동으로 float  
형태로 변환하여 저장한다.

```
float test = 13;
```

## 리터럴의 타입

float 타입의 변수를 하나 선언하고  
소수를 저장하여보자.

```
11  
12  
13 float WIDTH = 30.5;  
14  
15  
16
```

리터럴 역시 데이터 이기에 그자체의 데이터 타입이 있다

```
final double PI = 3.14;
```

리터럴 소수는 기본적으로  
double형이다

```
float width = 30.5f;
```

```
float a = (float) 3.5;
```

Float  
타입으로  
변경

명시적타입캐스팅



## 타입 불일치

```
char a = 110;
```

Char 는 내부적으로 유니코드를 저장

```
int b = 'a';
```

리터럴은 char형이고 자동타입캐스팅  
으로 a 문자에 대한 유니코드로  
변환되어 숫자가 저장된다.

```
int c = (int)3.14;
```

명시적 타입캐스팅 소수점은 사라지고  
3이 저장된다.

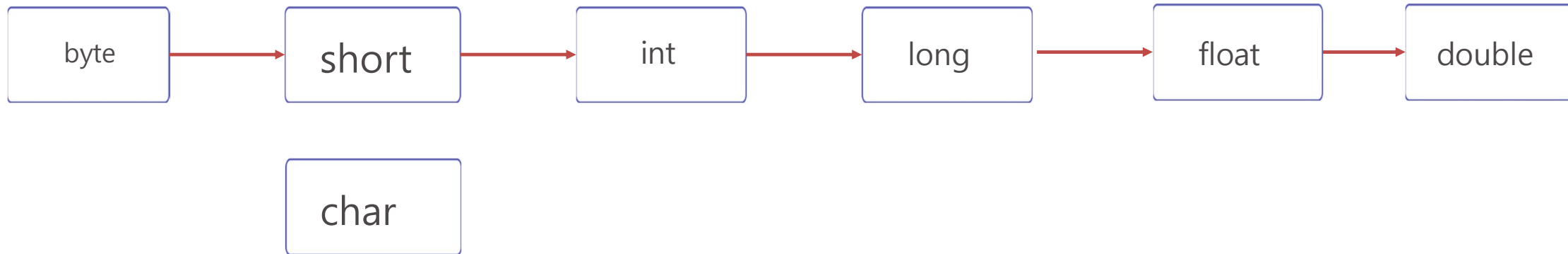
```
double d = 3.14f;
```

접미사 타입캐스팅

```
int e = "abc";
```

문자열은 int형 캐스팅 불가능

작은것에서 큰것으로는 자동캐스팅되며 데이터 손실이 없다.



# 04

## 참조형

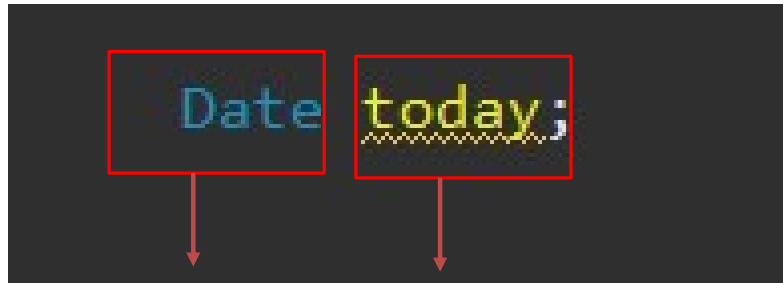
## 클래스



## 객체(인스턴스)

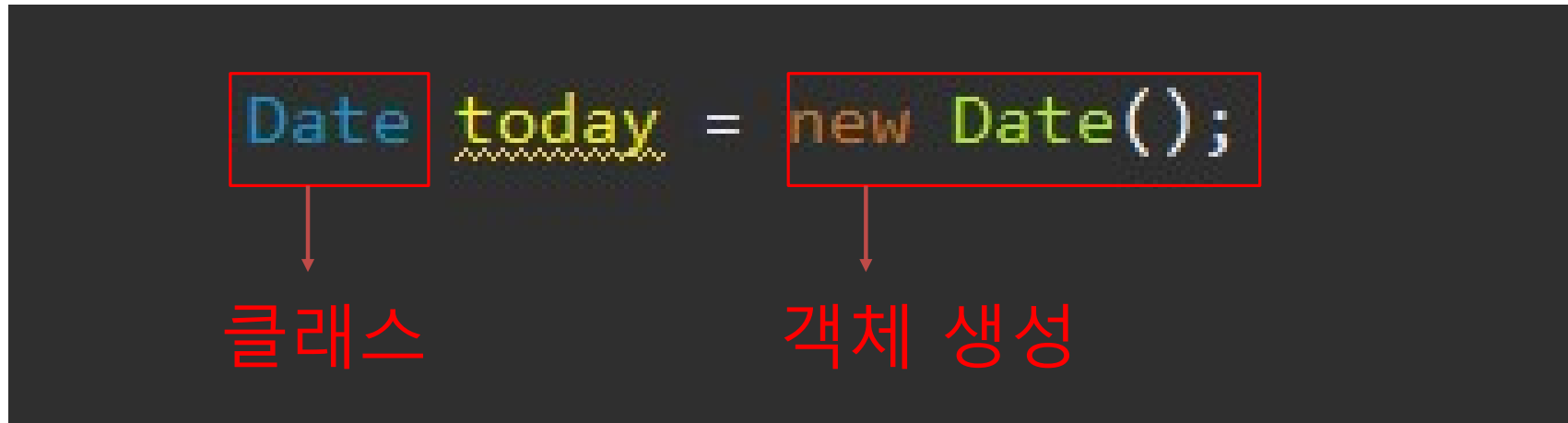


```
Date today;
```



클래스 참조형 변수 (초기값이 없어 null을 가지고 있다.)

```
Date today = new Date();
```



클래스                      객체 생성

---

# THANK YOU

강사 박주병