

{ JAVA }

선린인터넷고
소프트웨어과

프로그래밍
(J A V A)

O. T.

프로그래밍(JAVA)

자료구조 정보통신 컴시일 컴구조



2학년

정보통신(웹프로그래밍)

[NCS] 게임콘텐츠제작

[NCS] 스마트앱콘텐츠 제작

JAVA Programming

1. JAVA 언어의 이해 및 기초문법...
2. Objected-Oriented Programming 개념
3. Class 개요
4. 상속(Inheritance), 다형성(Polymorphism)
5. 추상클래스(Abstract-Class), Interface
6. 패키지(Package) 개요 및 활용
7. 기타 등등...

참고하세요.....

JDK 설치

생활 코딩

<https://www.opentutorials.org/module/516/5556>

이클립스 설치

생활코딩

<https://www.opentutorials.org/module/516/5567>

참고사이트

1. 생활코딩
2. 이것이 자바다
<http://cafe.naver.com/thisisjava>
3. 자바의 정석
<http://cafe.naver.com/javachobostudy>



{ 1부 }

JAVA 기본문법 개요

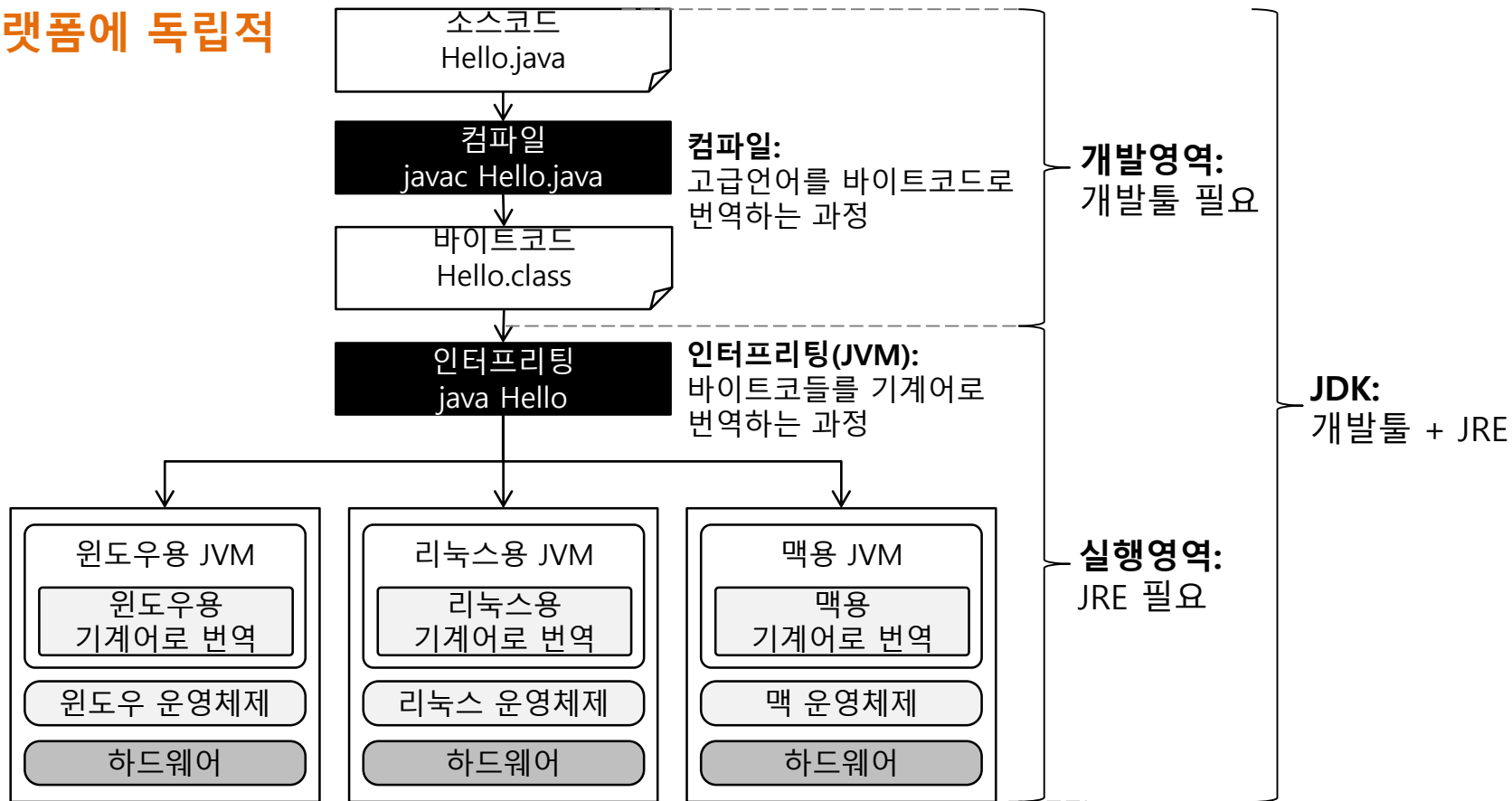
{ INDEX }

1. 자바 시작하기
2. 자바 개발환경
3. 자바 프로그램작성
4. 예약어와 식별자
5. 변수와 자료형
6. 상수와 형변환
7. 연산자
8. 조건문
9. 반복문
10. 배열
- * 콘솔 입력과 출력

1. 자바 시작하기

❖ 자바 언어의 특징-1

■ 플랫폼에 독립적

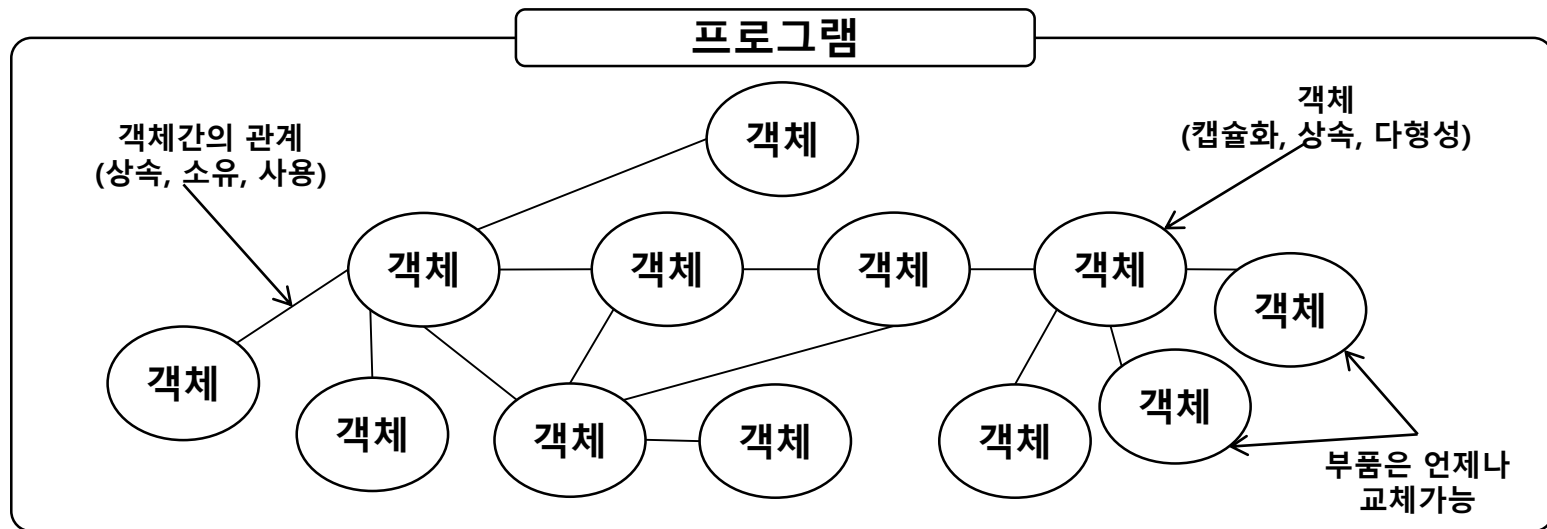


1. 자바 시작하기

❖ 자바 언어의 특징-2

■ 객체지향 프로그래밍(Object-Oriented Programming: OOP)

- 100% 객체지향 언어
- 서로 협동하는 다수의 객체들로 이루어진 프로그램
- 캡슐화, 상속, 다형성 기능을 지원



1.자바 시작하기

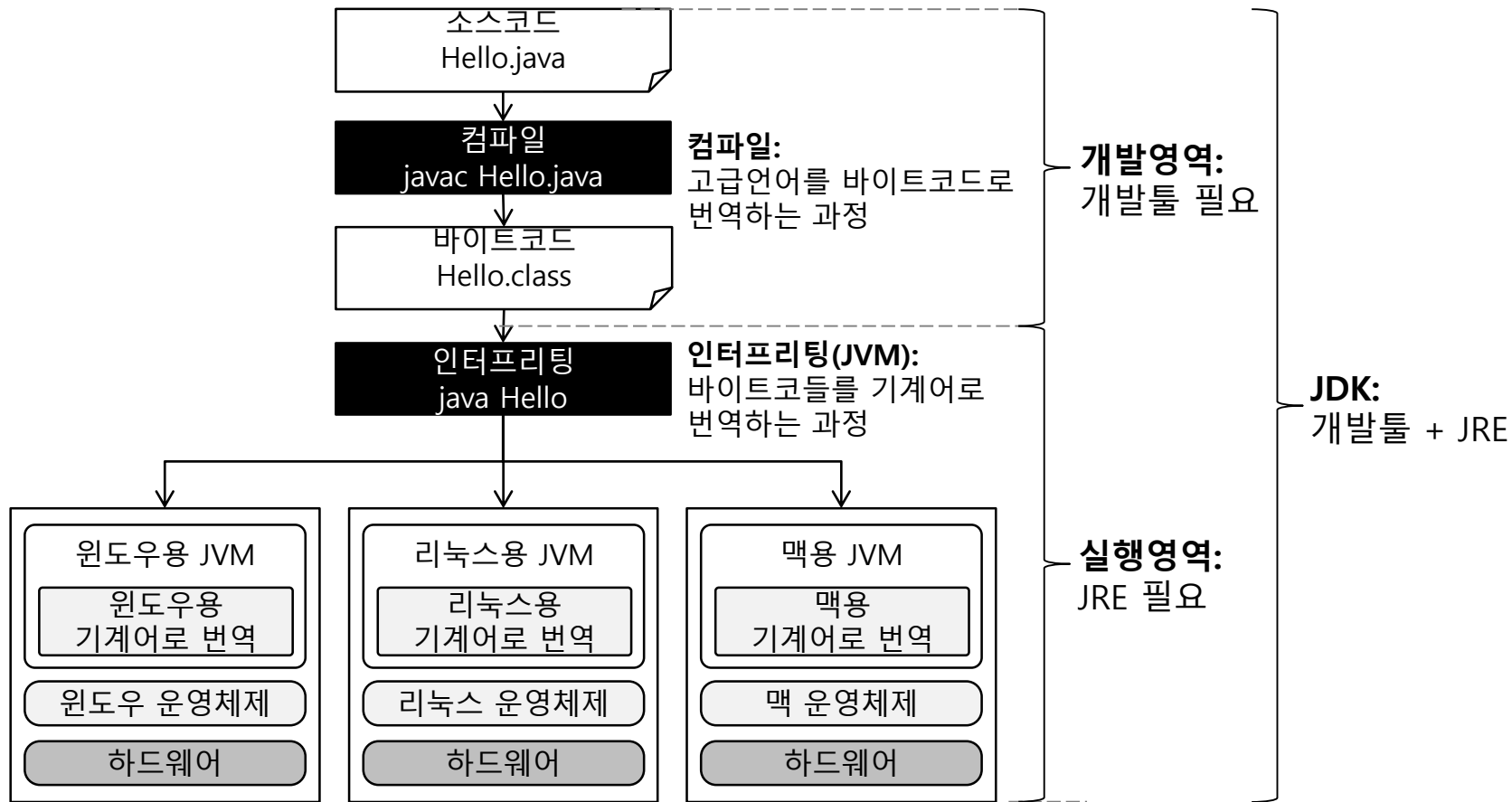
❖ 자바 언어의 특징-3

■ 단순하고 상대적으로 배우기 쉬움

- 자바 언어는 C/C++ 언어의 구문과 C++ 언어의 객체지향성을 채택
포인터, 다중상속 등의 복잡한 기능은 제외시킴
- 동적으로 할당되는 메모리를 프로그래머가 신경 쓸 필요 없이
Garbage Collector를 통해 자동으로 메모리를 관리해줌
- 임의의 데이터베이스와 쉽게 연결하기 위한
JDBC(Java Database Connectivity) 제공

1. 자바 시작하기

❖ 자바 프로그램 동작방식



{ INDEX }

1. 자바 시작하기
- 2. 자바 개발환경**
3. 자바 프로그램작성
4. 예약어와 식별자
5. 변수와 자료형
6. 상수와 형변환
7. 연산자
8. 조건문
9. 반복문
10. 배열
- * 콘솔 입력과 출력

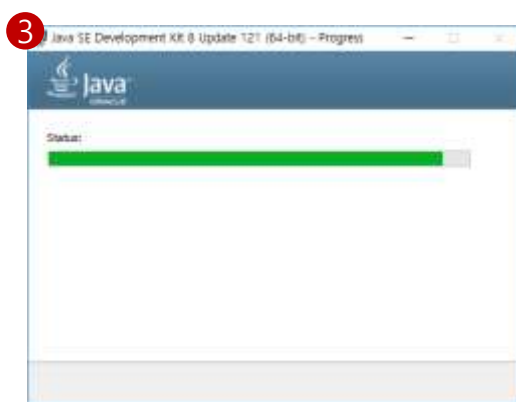
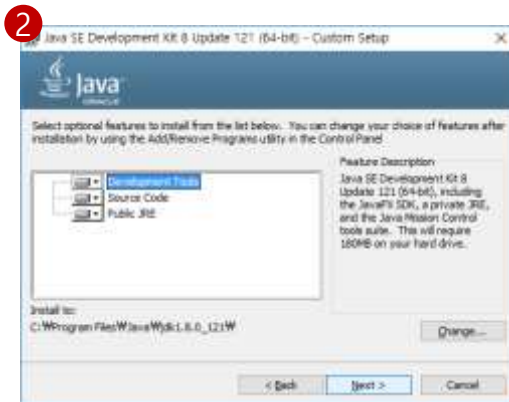
2. 자바 개발 환경

❖ JDK 다운로드 및 설치

- JDK(JAVA Development Kit)
 - 자바 프로그램의 개발을 위한 소프트웨어와 라이브러리의 모음
 - <http://www.oracle.com> (또는 <http://java.sun.com>)에서 무료로 다운로드
- JAVA의 분류
 - **JAVA SE(Standard Edition)**
 - 일반 응용 프로그램 개발용 자바
 - JAVA EE(Enterprise Edition)
 - 엔터프라이즈 개발용 자바
 - JAVA ME(Micro Edition)
 - 소형 디바이스 개발용 자바

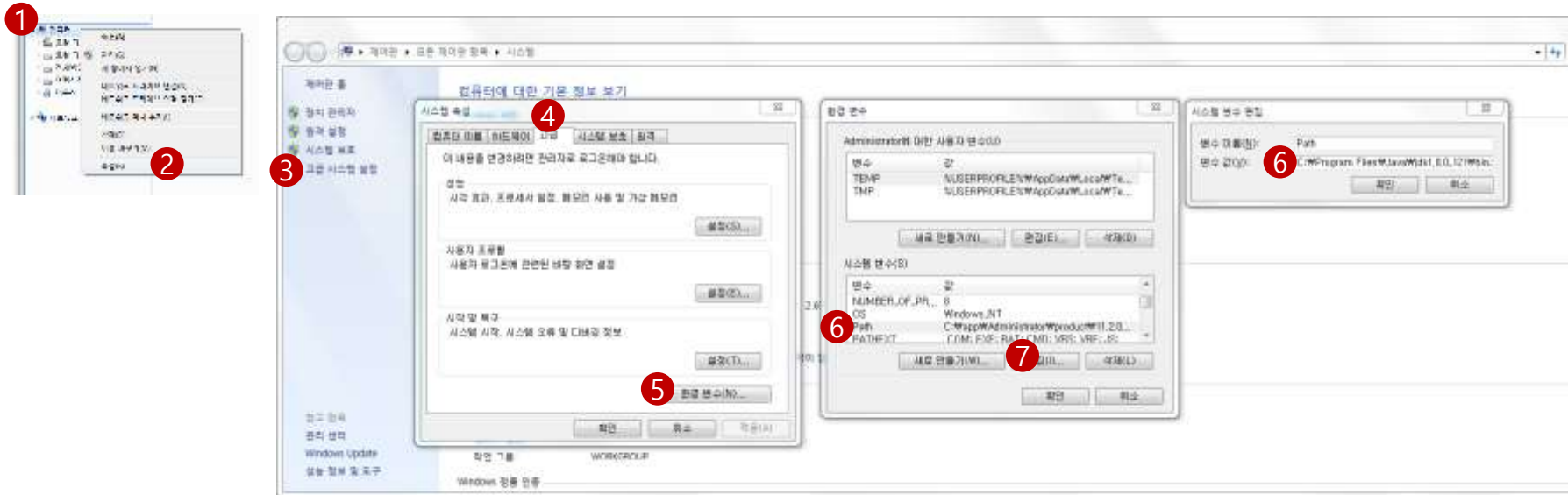
❖ JDK 다운로드 및 설치

■ 설치



2.자바개발환경

❖ 환경변수 설정- Windows 7



C:\Program Files\Java\jdk1.8.0_121\bin ; C:\Program Files (x86)\ATI Technologies\ATI.ACE\Core-Static

숫자는 다를 수 있음

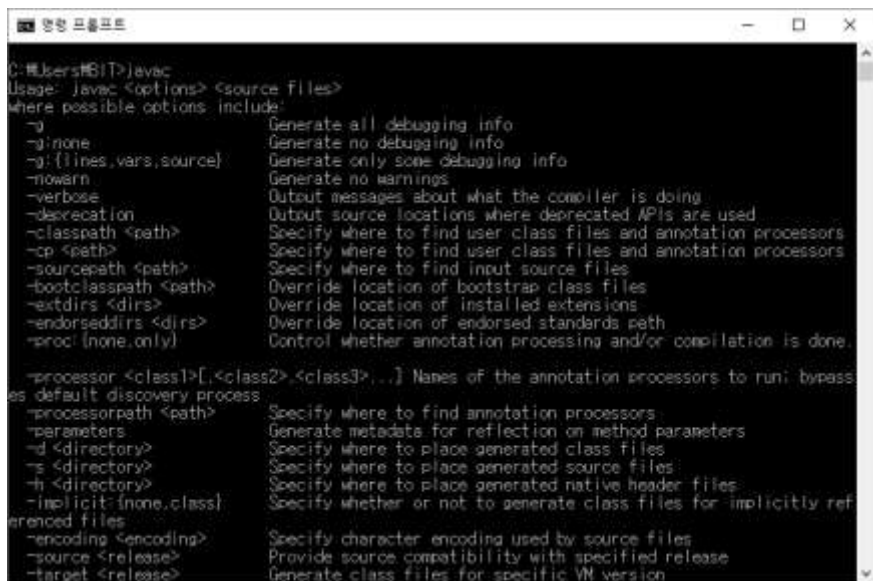
❖ 환경변수 설정- Windows 8



C:\Program Files\Java\jdk1.8.0_121\bin

❖ 환경변수 설정 확인

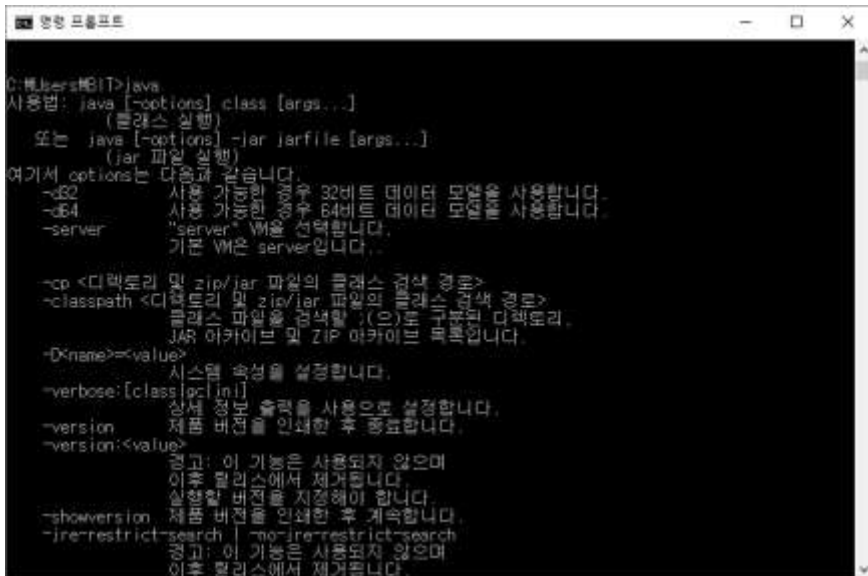
javac 입력



```
C:\Users\HBIT>javac
Usage: javac <options> <source files>
where possible options include:
  -g               Generate all debugging info
  -g:none          Generate no debugging info
  -g:{lines,vars,source} Generate only some debugging info
  -nowarn          Generate no warnings
  -verbose         Output messages about what the compiler is doing
  -deprecation     Output source locations where deprecated APIs are used
  -classpath <path> Specify where to find user class files and annotation processors
  -cp <path>       Specify where to find user class files and annotation processors
  -sourcepath <path> Specify where to find input source files
  -bootclasspath <path> Override location of bootstrap class files
  -extdirs <dirs>   Override location of installed extensions
  -endorseddirs <dirs> Override location of endorsed standards path
  -proc:{none,only} Control whether annotation processing and/or compilation is done.

  -processor <class1>[,<class2>,<class3>,...] Names of the annotation processors to run; bypass
  es default discovery process
  -processorpath <path> Specify where to find annotation processors
  -parameters      Generate metadata for reflection on method parameters
  -d <directory>    Specify where to place generated class files
  -s <directory>    Specify where to place generated source files
  -h <directory>    Specify where to place generated native header files
  -implicit:{none,class} Specify whether or not to generate class files for implicitly ref
  erenced files
  -encoding <encoding> Specify character encoding used by source files
  -source <release>  Provide source compatibility with specified release
  -target <release> Generate class files for specific VM version
```

java 입력



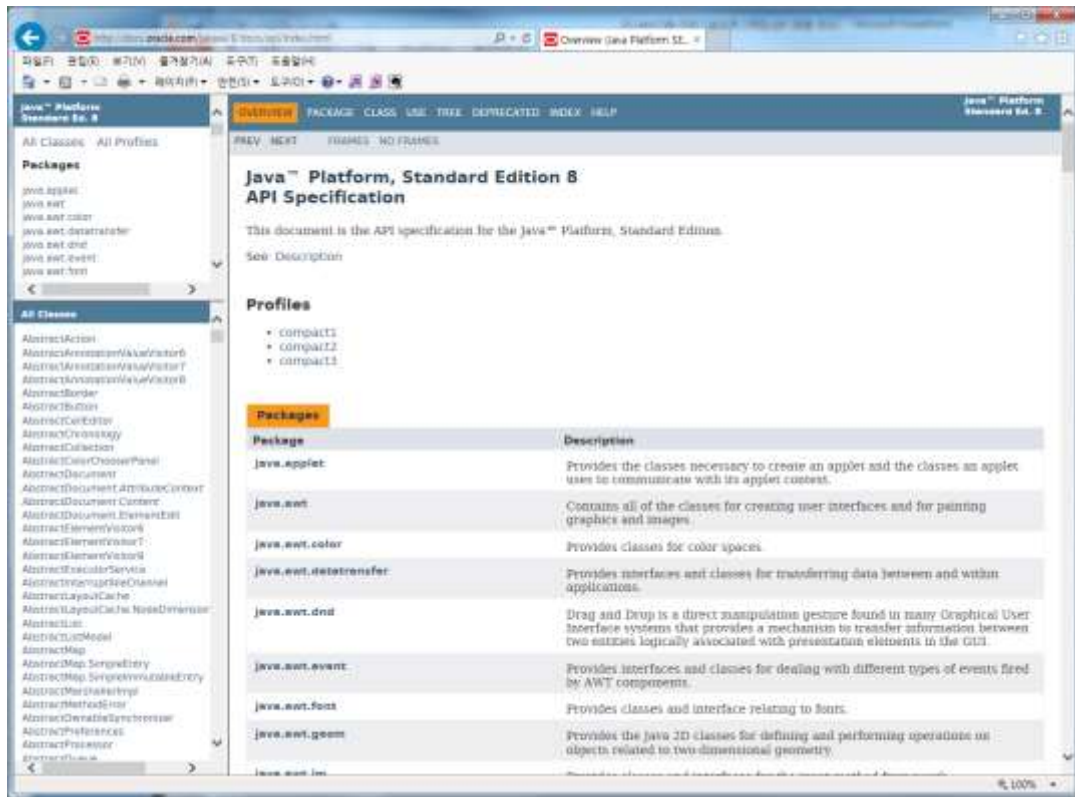
```
C:\Users\HBIT>java
사용법: java [-options] class [args...]
        (클래스 실행)
또는 java [-options] -jar jarfile [args...]
        (jar 파일 실행)
여기서 options는 다음과 같습니다.
  -d32             사용 가능한 경우 32비트 데이터 모델을 사용합니다.
  -d64             사용 가능한 경우 64비트 데이터 모델을 사용합니다.
  -server          "server" VM을 선택합니다.
                  기본 VM은 server입니다.

  -cp <디렉토리 및 zip/jar 파일의 클래스 검색 경로>
  -classpath <디렉토리 및 zip/jar 파일의 클래스 검색 경로>
                  클래스 파일을 검색할 (으)로 구분된 디렉토리,
                  JAR 아카이브 및 ZIP 아카이브 목록입니다.
  -D<name>=<value> 시스템 속성을 설정합니다.
  -verbose:[class|oc|ini] 상세 정보 출력에 사용으로 설정합니다.
  -version         제품 버전을 인쇄한 후 종료합니다.
  -version:<value> 경고: 이 기능은 사용되지 않으며
                  이후 릴리스에서 제거됩니다.
                  실행할 버전을 지정해야 합니다.
  -showversion     제품 버전을 인쇄한 후 계속합니다.
  -jre-restrict-search | -no-jre-restrict-search
                  경고: 이 기능은 사용되지 않으며
                  이후 릴리스에서 제거됩니다.
```

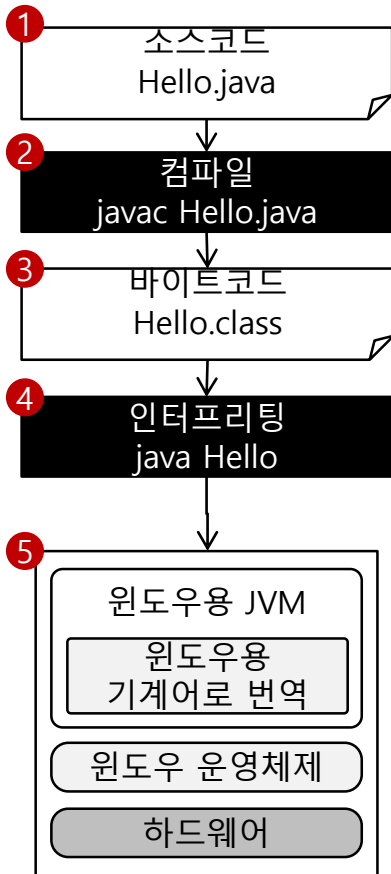

2. 자바 개발 환경

❖ JDK Document (JAVA API)

- <http://docs.oracle.com/javase/8/docs/api/index.html>



❖ Hello World 작성해보기



(1)소스코드작성

- 메모장을 이용해서 아래의 소스코드를 작성하고 "HelloWorld.java" 로 저장
- 저장위치: D:\javaStudy\workspace

```
public class HelloWorld{  
  
    public static void main(String[] args){  
        System.out.println("Hello World!");  
    }  
}
```

(2)컴파일

- 명령프롬프트 실행
- HelloWorld.java 파일이 저장된 디렉토리로 이동 (D:\javaStudy\workspace)
- 명령프롬프트에 다음과 같이 입력 **javac HelloWorld.java**

(3)바이트 코드생성 확인

- D:\javaStudy\workspace 에서 확인 →파일명: HelloWorld.class

(4)실행(인터프리팅)하여 결과확인

- 명령프롬프트에 다음과 같이 입력 **java HelloWorld**

2. 자바 개발 환경

❖ 이클립스(Eclipse)

■ 이클립스 특징

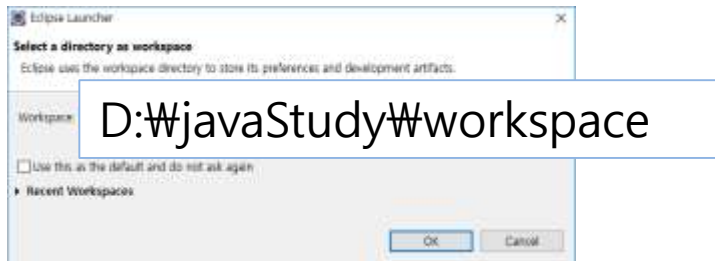
- 강력한 기능과 깔끔한 인터페이스를 갖춘 자바개발에 가장 많이 사용하는 툴
- IBM, 래쇼널 소프트웨어, 레드햇 등 여러 업체의 컨소시엄으로 개발되어 발전하고 있는 오픈소스
- 개발의 편의를 위해 다양한 플러그인을 제공하고 있다.

■ 이클립스 설치

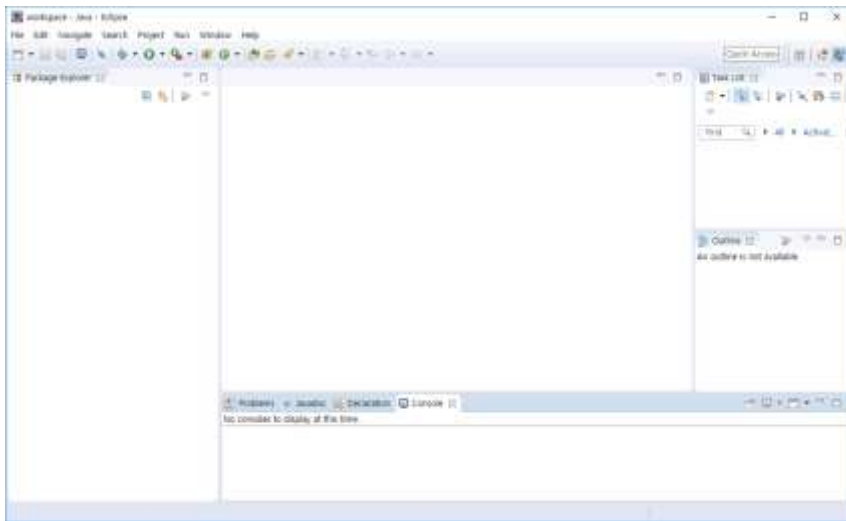
- <https://www.eclipse.org/> 에서 자신의 OS에 맞는 이클립스를 다운로드 한다
- D:\javaStudy 디렉토리에 압축을 푼다 → D:\javastudy\weclipse 가 생성된다. (확인)

■ 이클립스 실행

- workspace 디렉토리선택



- 이클립스 실행화면



{ INDEX }

1. 자바 시작하기
2. 자바 개발환경
- 3. 자바 프로그램작성**
4. 예약어와 식별자
5. 변수와 자료형
6. 상수와 형변환
7. 연산자
8. 조건문
9. 반복문
10. 배열
- * 콘솔 입력과 출력

3.자바 기본 프로그래밍

❖ 프로그래밍 실습

(1)프로젝트 만들기

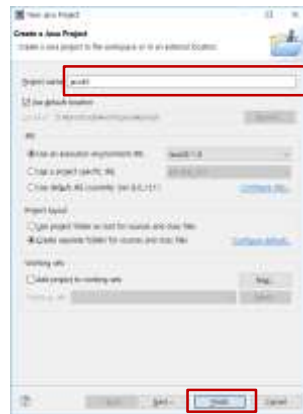
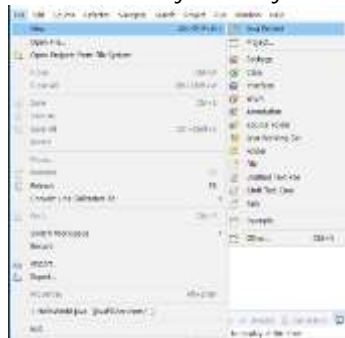
(2)패키지만들기

(3)클래스만들기

(4)소스코드작성

(5)실행하기

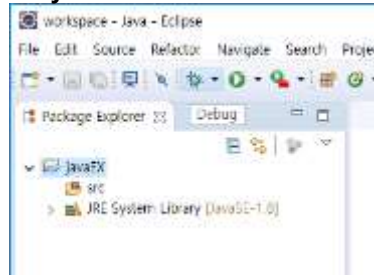
File > New > java Project 선택



javaEx 입력

finish 버튼 클릭

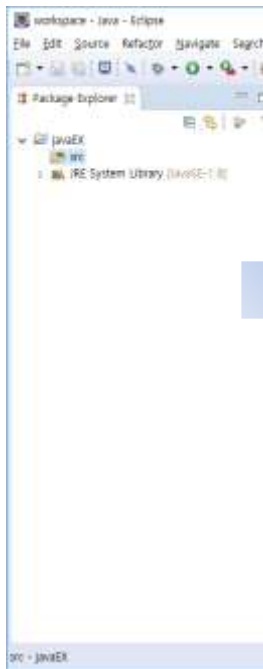
Project 생성 확인



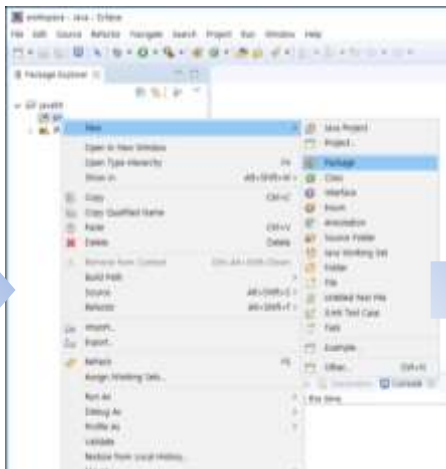
❖ 프로그래밍 실습

- (1)프로젝트 만들기
- (2)패키지만들기
- (3)클래스만들기
- (4)소스코드작성
- (5)실행하기

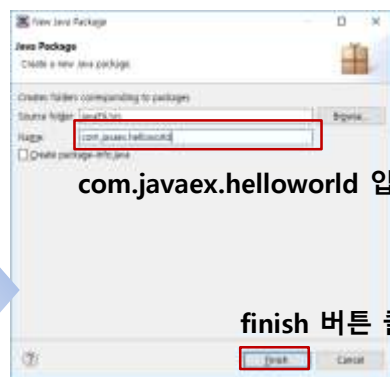
src 폴더 선택 후
오른쪽 마우스 클릭



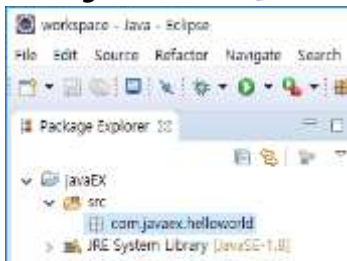
New > Package 선택



Package명 입력
*Package명은 소문자사용



Package 생성확인



3.자바 기본 프로그래밍

❖ 프로그래밍 실습

(1)프로젝트 만들기

(2)패키지만들기

(3)클래스만들기

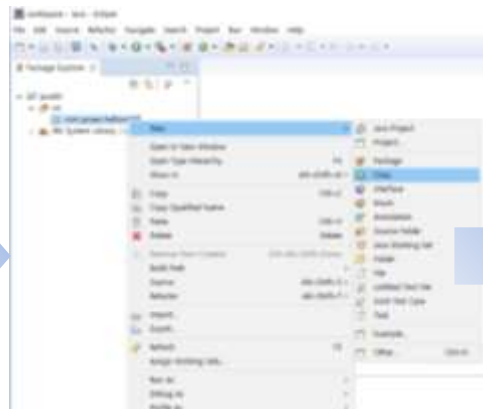
(4)소스코드작성

(5)실행하기

com.javaex.helloworld
패키지 선택 후
오른쪽 마우스 클릭



New > Class 선택



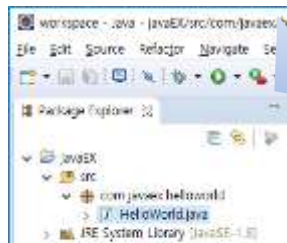
Class명 입력

*Class 명은 첫글자를 대문자로 사용

HelloWorld 입력



Class 생성확인



finish 버튼 클릭

❖ 프로그래밍 실습

(1)프로젝트 만들기

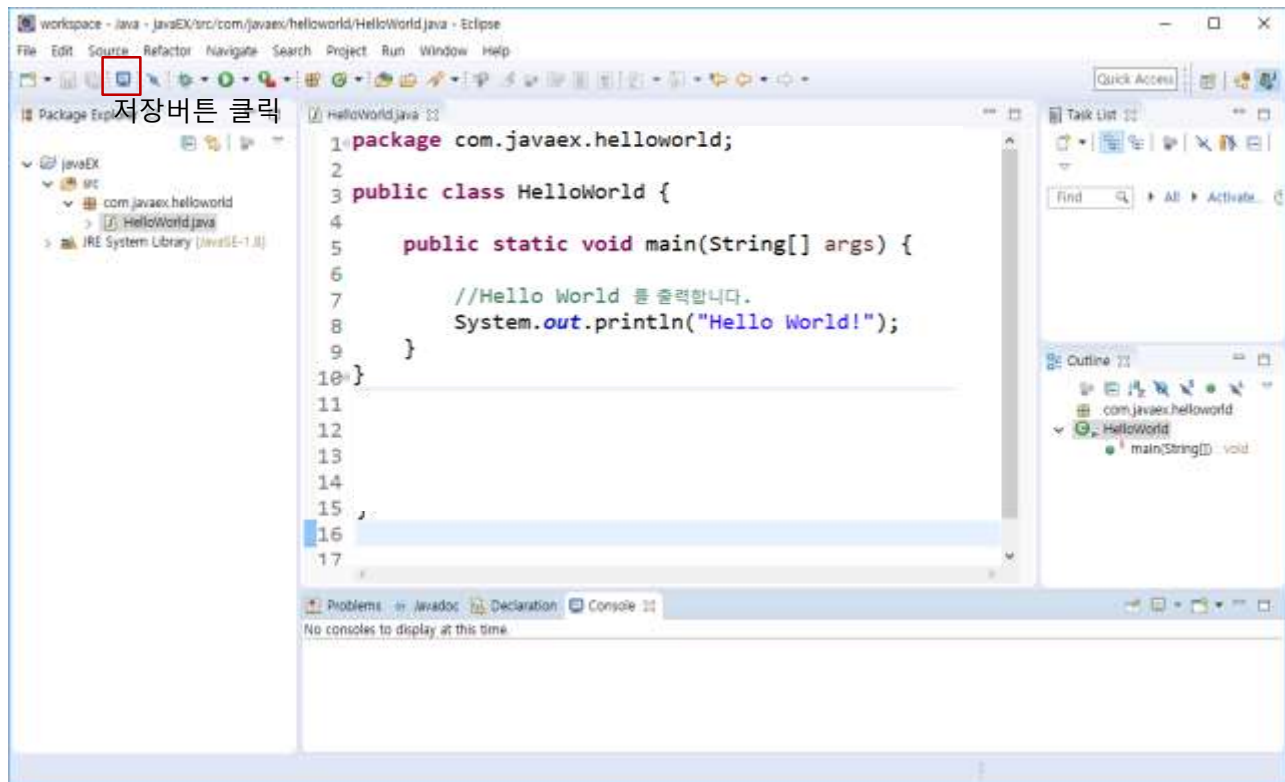
(2)패키지만들기

(3)클래스만들기

(4)소스코드작성

(5)실행하기

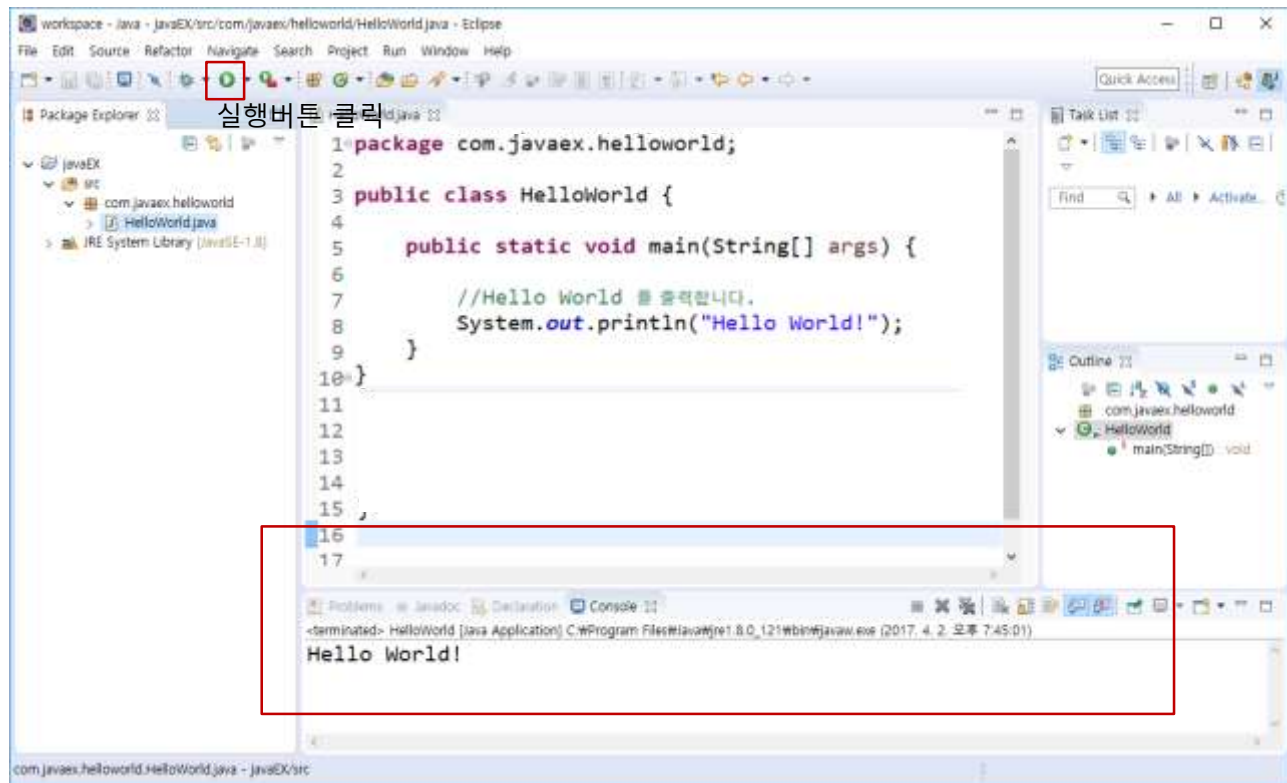
아래와 같이 소스코드를 작성 후 저장 버튼 클릭



❖ 프로그래밍 실습

- (1)프로젝트 만들기
- (2)패키지만들기
- (3)클래스만들기
- (4)소스코드작성
- (5)실행하기

실행버튼 클릭 후 console창 결과 확인



3.자바 기본 프로그래밍

❖ 코드살펴보기

```

1  /*
   작성자:홍길동
   작성일:2018.08.XX.
   설  명:Hello World 출력하기
2  */
   package com.javaex.helloworld;

3
   public class HelloWorld {
4
5       //Hello World 를 출력합니다.
6       System.out.println("Hello World!");7
   }

```

- (1) 여러줄 주석문
- (2) 패키지경로, 패키지명
패키지경로및 패키지명은 소문자로 작성
- (3) 클래스영역
클래스명의 첫글자는 대문자로 작성
소스코드 파일의 이름과 클래스이름은 동일
- (4) 메소드영역
자바 인터프리터는 자바 프로그램의 시작 지점으로
main 메소드를 찾음
메소드 내의 명령을 순차적으로 실행
- (5) 한줄주석문
- (6) **System.out.println** 메소드
표준 출력으로 명시된 문자열을 화면에 출력하는 역할을 함
- (7) 문장의 끝은 ; (세미콜론)로 표시

*주석문

- 주석문은 프로그램을 설명하기 위해 사용하는 문장으로 프로그램을 수행하는데 아무 영향을 미치지 못함.

3. 자바 기본 프로그래밍

❖ 프로그래밍 실습 과제 1 : **작성, 컴파일, 실행** (프로젝트, 패키지: welcome)

- **아래의 프로그램을 작성한 후 컴파일하고 실행해봅시다.**

```
// 실습 : Welcome.java
```

```
// 환영 메시지를 보여준다.
```

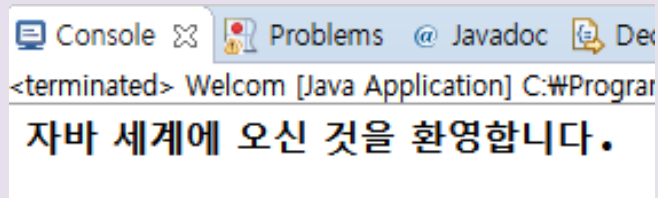
```
public class Welcome {
```

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

```
        System.out.println("자바 세계에 오신 것을 환영합니다.");
```

```
    }
```

```
}
```

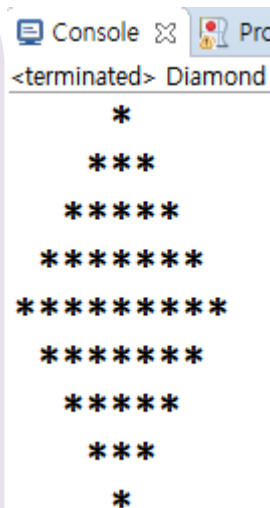


색깔은 신경쓰지 말자

❖ 프로그래밍 실습 과제 2 : **다이아몬드 모양 출력** (프로젝트,패키지: diamond)

- **아래의 프로그램을 작성한 후 컴파일하고 실행해봅시다.**

```
// 실습 : Diamond.java
// 다이아몬드 모양을 표준 출력 장치(모니터)에 출력한다.
public class Welcome {
    public static void main(String[] args){
        // 다이아몬드 모양을 여러 개의 출력문들을 사용해 출력함
        System.out.println("  *");
        System.out.println("   ***");
        // 여기에 적절한 코드를 삽입해 작성해봅시다.
    }
}
```



Console [X] [Run] Pro
<terminated> Diamond

```
  *
 ***
*****
*****
*****
*****
  ***
  *
```

❖ 특수문자(제어문자)

특수문자	표시방법
tab	\t
new line	\n
역슬래쉬 (\)	\\
작은따옴표 (')	'
큰따옴표 (")	"

3.자바 기본 프로그래밍

❖ 프로그래밍 실습 과제 3 : Esc.java (프로젝트,패키지: esc)

- 자기이름 출력
- 출신중학교 이름, 학년을 띄어쓰기 와 줄바꿈 출력하기 : `Wt` 또는 `Wn` 이용하기
- 'Hello'
- "Hello"
- "!@#\$%^&*()"
- "C:WDownloadWHello.java"

3.자바 기본 프로그래밍

❖ 프로그래밍 실습 과제 3 : Esc.java (프로젝트,패키지: esc)

도움코드

```
1 package esc;
2 public class Esc {
3     public static void main(String[] args) {
4         System.out.println("김선린");
5         System.out.println("인터넷중학교\t3학년\n졸업");
6         System.out.println("\'Hello\'");
7         System.out.println("\"Hello\"");
8         System.out.println("!@#$%^&*()");
9         System.out.println("C:\\Download\\Hello.java");
10    }
11 }
```

<terminated> Esc [Java Application] C:\Program Fil

김선린

인터넷중학교

3학년

졸업

'Hello'

"Hello"

!@#\$%^&*()

C:\Download\Hello.java

3.자바 기본 프로그래밍

❖ 프로그래밍 실습 과제 4 : Face.java (프로젝트,패키지: face)

■ 내용

- 텍스트문자를 이용해서 콘솔화면에 얼굴을 출력하는 프로그램을 작성하자.
(단, 최소한 아래보다는 멋있어야 한다.)
- 파일이름: Face.java -----> sunrinjava@gmail.com 제출

■ 예시

```
/////
|  o o  |
(|   ^   |)
|  [ - ]  |
-----
```

3. 자바 기본 프로그래밍

❖ 자바프로그래밍 공부할 때 참고할 만한 사이트

참고사이트

1. 서울위즈

<https://www.youtube.com/watch?v=MD3YGrofnDY&list=PLieE0qnqO2kTyzAlsvxzoulHVISvO8zA9>

2. 생활코딩 <https://www.opentutorials.org/module/516>

3. 이것이 자바다 <http://cafe.naver.com/thisisjava>

4. 자바의 정석 <http://cafe.naver.com/javachobostudy>

5. 윤성우의 프로그래밍 스터디 <https://cafe.naver.com/cstudyjava>

{ INDEX }

1. 자바 시작하기
2. 자바 개발환경
3. 자바 프로그램작성
- 4. 예약어와 식별자**
5. 변수와 자료형
6. 상수와 형변환
7. 연산자
8. 조건문
9. 반복문
10. 배열
- * 콘솔 입력과 출력

4.예약어와 식별자

❖ 식별자

프로그래머가 직접 만들어줘야 하는 이름
예) 변수명, 클래스명, 메소드명 등

- 알파벳 대 소문자, 숫자, `_`, `$` 를 사용 가능
- 첫 문자는 숫자 사용 불가
- `$`와 `_` 이외의 다른 특수문자는 사용 불가
- JAVA 예약어 사용불가
- 영문 대소문자를 구분
- 길이의 제한이 없음?!

```
/*
  작성자:홍길동
  작성일:2018.08.XX.
  설  명:Hello World 출력하기
 */
package com.javaex.helloworld;

public class HelloWorld {

    public static void main(String[] args) {

        //Hello World 를 출력합니다.
        System.out.println("Hello World!");
    }
}
```

4.예약어와 식별자

❖ 예약어

문법적인 기능을 수행하기 위해 미리 정의된 단어 : 16쪽 [표 1-3]

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

```
/*
  작성자:홍길동
  작성일:2018.08.XX.
  설  명:Hello World 출력하기
 */
package com.javaex.helloworld;

public class HelloWorld {

    public static void main(String[] args) {

        //Hello World 를 출력합니다.
        System.out.println("Hello World!");
    }
}
```

4.예약어와 식별자

❖ 사용 가능한 예

```
int    name;
char   student_ID;           // '_' 사용 가능
void   $func() { }           // '$' 사용 가능
class  Monster3 { }           // 숫자 사용 가능 (단, 맨 앞 위치는 안됨)
int     whatsyournamemynameiskitae; // 길이 제한 없음?!
int     barChart;  int barchart; // 대소문자 구분. barChart와 barchart는 다름
int     가격;                 // 한글 이름 사용 가능 (단, 가급적 사용 자제)
```

❖ 잘못된 예

```
int     3Chapter;           // 식별자의 첫문자로 숫자 사용 불가
class   if { }               // 자바의 예약어 if 사용 불가
char     false;              // false 사용 불가
void     null() { }           // null 사용 불가
class    %calc { }           // '%'는 특수문자
```

{ INDEX }

1. 자바 시작하기
2. 자바 개발환경
3. 자바 프로그램작성
4. 예약어와 식별자
- 5. 변수와 자료형**
6. 상수와 형변환
7. 연산자
8. 조건문
9. 반복문
10. 배열
- * 콘솔 입력과 출력

5. 변수와 자료형

❖ 변수(Variable)

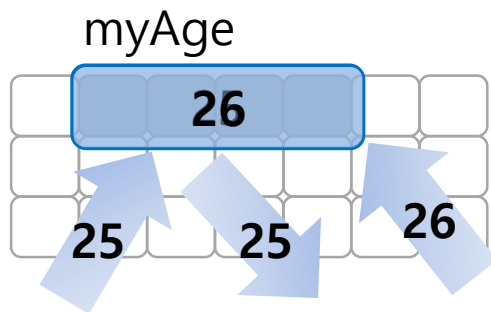
하나의 값(데이터)을 저장할 수 있는 메모리 공간
값을 저장하고, 확인하고, 변경하고 할 수 있어야 함

내 나이를 저장할래

내 나이는 25

내 나이는?

나이를 26 으로 변경



```
int myAge;    //선언
```

```
myAge = 25;   //초기화
```

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

```
myAge = 26;
```


5. 변수와 자료형

❖ 자료형(Data Type, 데이터형)

- 변수에 저장할 데이터의 종류
 - 자료형의 종류

자료형	키워드	크기	표현 범위	사용 예
논리형	boolean	1byte	true 또는 false (0과 1이 아니다)	boolean isFun = true;
문자형	char	2byte	모든 유니코드 문자	char c = 'f';
정수형	byte	1byte	-128 ~ 127	byte b = 89;
	short	2byte	-32,768 ~ 32,767	short s = 32760;
	int	4byte	-2147483648 ~ 2147483647	int x = 59;
	long	8byte	-9223372036854775808 ~ 9223372036854775807	long big = 345678912345L;
실수형	float	4byte	-3.4E38 ~ 3.4E38	float f = 32.5F
	double	8byte	-1.7E308 ~ 1.7E308	double d = 23.34

5. 변수와 자료형

❖ 자료형

■ 논리형(boolean type)

✓ 논리적인 표현을 위한 두 가지 약속

- true '참'을 의미하는 키워드
- false '거짓'을 의미하는 키워드

✓ 키워드 true와 false에 대한 좋은 이해

- 숫자의 관점에서 이해하려 들지 말자.
- 자바에서의 true와 false는 그 자체로 저장이 가능한 데이터이다. (소문자 !)
- true와 false의 저장을 위한 자료형 boolean

❖ 예제 : 논리형(boolean) 타입 예제

```
public class BooleanExam{    // BooleanExam.java
    public static void main( String[] args ) {
        boolean b1;        // 변수 선언
        b1 = true;          // 변수 초기화

        boolean b2 = false; // 변수 선언 + 초기화

        System.out.println("b1 = " + b1);
        System.out.println("b2 = " + b2);
    }
}
```

5. 변수와 자료형

❖ 자료형

■ 문자형

- 문자 하나를 **2바이트**로 표현하는 **유니코드** 기반으로 표현
- 유니코드는 전 세계의 문자를 표현할 수 있는 코드 집합
- 문자는 작은 따옴표로 감싸줘서 표현한다.
- 문자는 char형 변수에 저장한다. 저장 시 실제로는 유니코드 값 저장

```
char ch1='A';  
char ch2='한';
```

변환 발생



```
char ch1=65;           // 65는 16진수로 0x41  
char ch2=54620;        // 54620은 16진수로 0xD55C
```

■ * 문자열

```
String str="안녕하세요";
```

5. 변수와 자료형

❖ 예제 : 문자형(char) 타입 예제

```
public class CharExam{    // CharExam.java
    public static void main( String[] args ) {
        char ch1;        // 변수 선언
        ch1 = 'A';        // 변수 초기화

        char ch2 = 'Z';   // 변수 선언 + 초기화

        System.out.println("ch1 = " + ch1);
        System.out.println("ch2 = " + ch2);
    }
}
```

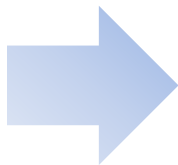
5. 변수와 자료형

❖ 자료형

■ 정수형(양의정수)

- 가장 왼쪽 비트인 MSB(Most Significant Bit)는 부호를 나타낸다.
- MSB를 제외한 나머지는 크기를 나타낸다.
- 바이트 크기의 차이는 표현범위의 차이로 이어진다.

byte 일때 → 1byte = 8bit



- MSB 0 양 수
- 0011001 25

양의 정수 25

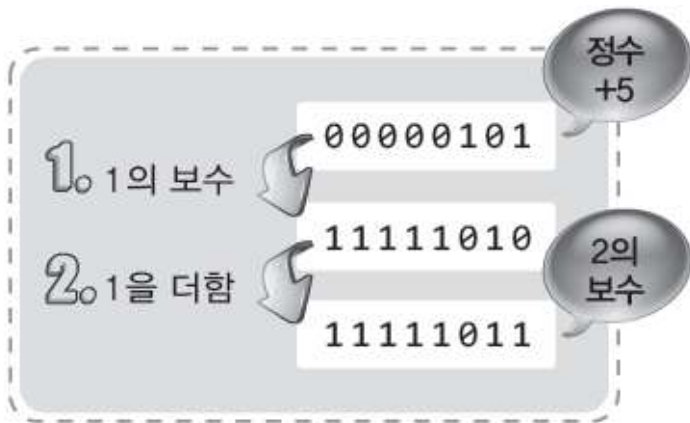
✓ 그럼 -25는??

5. 변수와 자료형

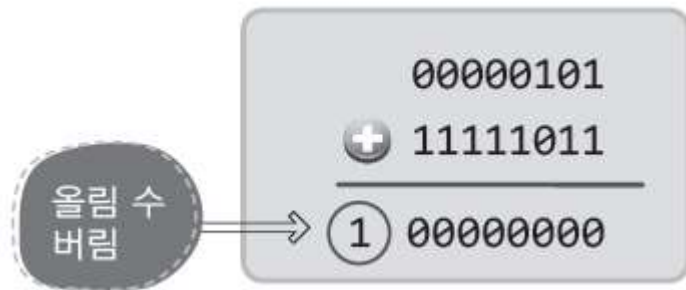
❖ 자료형

■ 정수형(음의정수)

- 양의 정수 표현방식과 다르다.
- 양의 정수와의 합이 0이 되는 구조로 정의
- 2의 보수가 음의 정수 표현방식



-5의 2진수 표현



검 증

5. 변수와 자료형

❖ 예제 : 정수형(int) 타입 예제

```
public class Variable{    // Variable.java
    public static void main( String[] args ) {
        int num1;        // 변수 선언
        num1 = 10;        // 변수 초기화

        int num2 = 20;    // 변수 선언 + 초기화

        int num3 = num1 + num2;


        System.out.println("num1 + num2 = " + num3);
    }
}
```


❖ 실습 : 자신의 나이를 정수형 변수를 이용해서 출력해봅시다.

```
public class Exam1{    // Exam1.java
    public static void main( String[] args ) {
        // 변수 선언

        // 변수 초기화

        // 변수 출력
    }
}
```

 Problems @ Java

<terminated> ex [Java

내 나이 : 17

5. 변수와 자료형

❖ 실습 : 자신의 나이를 정수형 변수를 이용해서 출력해봅시다.

도움코드

```
1 public class Exam1 {  
2     public static void main(String[] args) {  
3         int age; // 변수 선언  
4         age = 17; // 변수 초기화(초기값 대입)  
5         System.out.println("내 나이: "+age); // 출력  
6     }  
7 }
```

 Problems @ Java

<terminated> ex [Java

내 나이 : 17

5. 변수와 자료형

❖ 자료형

■ 실수형

✓ 실수표현의 문제점

- 0과 1사이의 실수만 해도 그 수가 무한대
- 단순히 몇 바이트 정도로 모든 실수의 표현은 불가능하다.

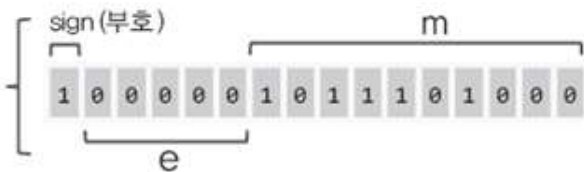
✓ 해결책

- 정밀도를 포기하고, 표현할 수 있는 값의 범위를 넓히자

실수의
표현을 위한 수식

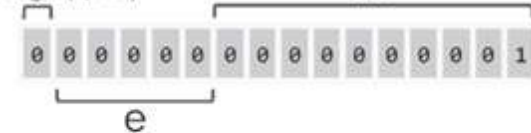
$$\pm (1.m) \times 2^{e-127}$$

수식에
반영하는 비트 구성



실수표현을 위한 수식의 도입

sign (부호)



$$+ (1.1) \times 2^{-127}$$

표현의 예

❖ 예제 : 실수형(double) 타입 예제

```
public class DoubleExam{    // DoubleExam.java
    public static void main( String[] args ) {
        double d1;          // 변수 선언
        d1 = 1.0000001;      // 변수 초기화

        double d2 = 2.0000001; // 변수 선언 + 초기화

        double d3 = d1 + d2;

        System.out.println( d3 );
    }
}
```

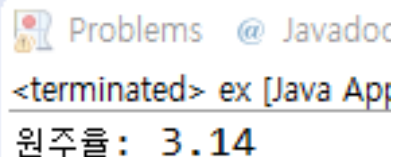
5. 변수와 자료형

- ❖ 실습 : 원주율값을 실수형 변수를 이용해 출력해봅시다.(단, 원주율을 3.14라 하자)

```
public class Exam2{    // Exam2.java
    public static void main( String[] args ) {
        // 변수 선언

        // 변수 초기화

        // 변수 출력
    }
}
```




Problems @ Javadoc
<terminated> ex [Java App
원주율 : 3.14

5. 변수와 자료형

❖ 실습 : 원주율값을 실수형 변수를 이용해 출력해봅시다.(단, 원주율을 3.14라 하자)

도움코드

```
1 public class Exam2 {  
2     public static void main(String[] args) {  
3         double pi;  
4         pi = 3.14;  
5         System.out.println("원주율: "+pi);  
6     }  
7 }
```

 Problems @ Javadoc
<terminated> ex [Java Applet]
원주율: 3.14

5. 변수와 자료형

❖ 생각해봅시다....

- 괄호 안에 적절한 데이터 타입을 기술하라.

```
(    ) number;           // 학번
(    ) name;             // 이름
(    ) isEnrolled;       // 등록 여부
(    ) grade;            // 평점
(    ) address;          // 주소
(    ) major;            // 전공
(    ) unit;             // 이수 학점
(    ) haveMinor;        // 부전공 여부
(    ) juminNo;          // 주민번호(-없이 13자리숫자)
(    ) cellphone;        // 핸드폰 번호(-포함한 숫자)
(    ) age;              // 나이
(    ) email;            // 이메일주소
```

5. 변수와 자료형

❖ 생각해봅시다.... 도움자료

- 괄호 안에 적절한 데이터 타입을 기술하라.

```
( int ) number;           // 학번
( String ) name;          // 이름
( boolean ) isEnrolled;   // 등록 여부
( double ) grade;         // 평점
( String ) address;       // 주소
( String ) major;         // 전공
( int ) unit;             // 이수 학점
( boolean ) haveMinor;    // 부전공 여부
( long ) juminNo;         // 주민번호(-없이 13자리숫자)
( String ) cellphone;     // 핸드폰 번호(-포함한 숫자)
( int ) age;              // 나이
( String ) email;         // 이메일주소
```


{ INDEX }

1. 자바 시작하기
2. 자바 개발환경
3. 자바 프로그램작성
4. 예약어와 식별자
5. 변수와 자료형
- 6. 상수와 형변환**
7. 연산자
8. 조건문
9. 반복문
10. 배열
- * 콘솔 입력과 출력

6.상수와 형변환

❖ 상수(Constant)

변경될 수 없는 고정된 데이터

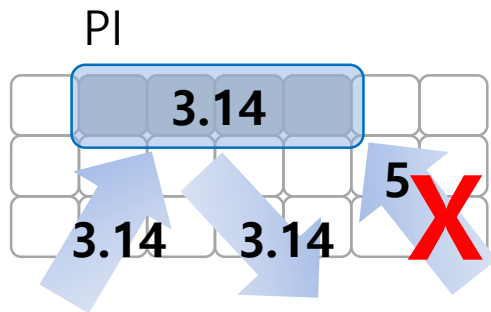
값은 초기화할 때 단 한번 저장할 수 있으며, 그 이후 변경할 수 없다.

코드의 이해와 변경이 쉬움.

원주율은 3.14

원주율은?

원주율을 5로 변경



final double PI = 3.14;

System.out.println(PI);

PI = 5; **//변경안됨**

java.lang

Class Math

java.lang.Object

java.lang.Math

PI

```
public static final double PI
```

The double value that is closer than any other to *pi*, the ratio of the circumference of a circle to its diameter.

6.상수와 형변환

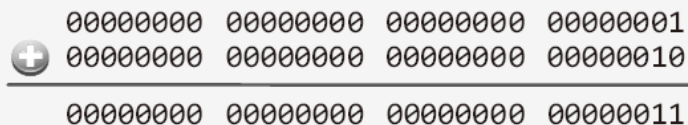
❖ 형변환

- 암묵적 형변환=자동타입변환

자료의 범위가 좁은 자료형에서 넓은 자료형으로의 변환은 시스템이 자동으로 행함

✓ 다른 자료형 끼리 연산을 할 경우

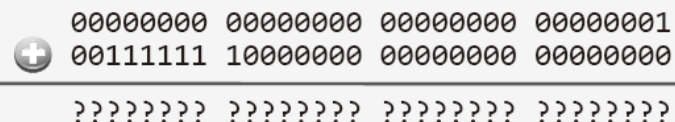
```
int num1 = 1;
int num2 = 2;
int result = num1 + num2;
```



00000000	00000000	00000000	00000001
+	00000000	00000000	00000010
<hr/>			
00000000	00000000	00000000	00000011

[그림 3-1 : 1 더하기 2는 3]

```
int num1 = 1;
double num2 = 1.0;
???? result = num1 + num2;
```



00000000	00000000	00000000	00000001
+	00111111	10000000	00000000
<hr/>			
????????	????????	????????	????????

[그림 3-2 : 1 더하기 1.0은?]

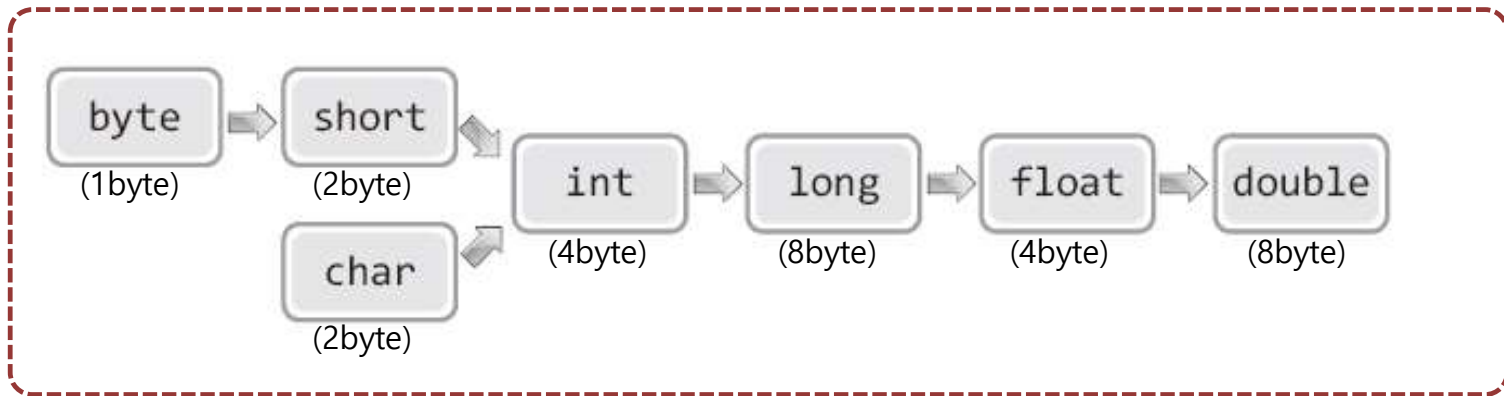
이 경우 **1.0** + 1.0 으로 자동 형변환 되어 2.0 으로 계산됨
 → int형이 **double**형으로 변경되어 계산됨

6.상수와 형변환

❖ 형변환

■ 형변환 규칙

값의 표현 범위



```
int num1 = 1;  
double num2 = 1.0;  
???? result = num1 + num2;
```

double

2.0

6. 상수와 형변환

❖ 형변환

- 명시적 **형변환**=강제타입변환

자료의 범위가 넓은 자료형에서 좁은 자료형으로의 변환은 시스템에서 자동으로 수행할 수 없으며 프로그래머가 강제로 변환해야 함

```
int intValue = 103029770;  
byte byteValue = (byte)intValue; //강제타입변환
```

00000110 00100100 00011100 00001010 ➔ 10

```
int intValue = 10;  
byte byteValue = (byte)intValue; //강제타입변환
```

00000000 00000000 00000000 00001010 ➔ 10

6. 상수와 형변환

❖ 예제

```
public class CastExam{    // CastExam.java
    public static void main( String[] args ) {
        char ch1 = 'A';
        int num = ch1;        // 자동타입 변환
        char ch2 = (char)num; // 강제타입 변환

        System.out.println("ch1 = " + ch1 );
        System.out.println("num = " + (char)num ); // 유니코드 확인, num
        System.out.println("ch2 = " + ch2 );
    }
}
```

{ INDEX }

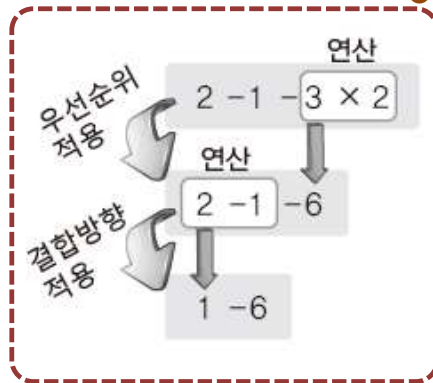
1. 자바 시작하기
2. 자바 개발환경
3. 자바 프로그램작성
4. 예약어와 식별자
5. 변수와 자료형
6. 상수와 형변환
7. 연산자
8. 조건문
9. 반복문
10. 배열
- * 콘솔 입력과 출력

7. 연산자

❖ 연산자 우선순위

연산기호	결합방향	우선순위
[], .	→	1(높음)
expr++, expr--	←	2
++expr, --expr, +expr, -expr, ~, !, (type)	←	3
*, /, %	→	4
+, -	→	5
<<, >>, >>>	→	6
<, >, <=, >=, instanceof	→	7
==, !=	→	8
&	→	9
^	→	10
	→	11
&&	→	12
	→	13
? expr : expr	←	14
=, +=, -=, *=, /=, %=, &=, ^=, =, <<=, >>=, >>>=	←	15(낮음)

연산의 과정



$48 \div 2(9+3) = 2 \text{ or } 288 ??$

❖ 대입연산자(=)와 산술 연산자(+, -, *, /, %)

연산자	연산자의 기능	결합방향
=	연산자 오른쪽에 있는 값을 연산자 왼쪽에 있는 변수에 대입한다. 예) val = 20;	←
+	두 피연산자의 값을 더한다. 예) val = 4 + 3;	→
-	왼쪽의 피연산자 값에서 오른쪽의 피연산자 값을 뺀다. 예) val = 4 - 3;	→
*	두 피연산자의 값을 곱한다. 예) val = 4 * 3;	→
/	왼쪽의 피연산자 값을 오른쪽의 피연산자 값으로 나눈다. 예) val = 7 / 3;	→
%	왼쪽의 피연산자 값을 오른쪽의 피연산자 값으로 나눴을 때 얻게 되는 나머지를 반환한다. 예) val = 7 % 3	→

7. 연산자

❖ 대입연산자(=)와 산술 연산자(+, -, *, /, %)

✓ / 연산자와 % 연산자의 연산방식

- 피연산자가 정수면 정수형 연산진행
- 피연산자가 실수면 실수형 연산진행, 단 % 연산자 제외!

❖ 예제

```
public class Arith{    // Arith.java
    public static void main( String[] args ) {
        int n1=7;
        int n2=3;
        int result = n1+n2;
        System.out.println( result );    // 덧셈
        System.out.println( n1 - n2 );    // 뺄셈
        System.out.println( n1 * n2 );    // 곱셈
        System.out.println( n1 / n2 );    // 나눗셈
        System.out.println( n1 % n2 );    // 나머지 연산
    }
}
```

❖ 예제

```
public class Div{    // Div.java
    public static void main( String[] args ) {
        System.out.println( 7 / 3 );    // 정수형 나눗셈
        System.out.println( 7.0 / 3.0 ); // 실수형 나눗셈
        System.out.println( 7 % 3 );    // 정수형 나머지
        System.out.println( 7.2 % 2.0 ); //실수형 나머지-수학적으로 문제있음
    }
}
```

- ❖ 실습 : 국어,영어,수학 및 그 평균 점수에 대한 적절한 변수를 이용해 출력해봅시다.
(단, 국어는 70점, 영어는 75점, 수학은 75점 이라고 가정하자.)

```
public class Exam3 {    // Exam3.java
    public static void main( String[] args ) {
        // 국어 점수 변수 선언 및 초기화
        // 영어 점수 변수 선언 및 초기화
        // 수학 점수 변수 선언 및 초기화
        // 평균 점수 변수 선언

        // 평균 점수를 구하여 변수에 저장

        // 평균 점수 출력
    }
}
```



7. 연산자

- ❖ 실습 : 국어, 영어, 수학 및 그 평균 점수에 대한 적절한 변수를 이용해 출력해봅시다.
(단, 국어는 70점, 영어는 75점, 수학은 75점 이라고 가정하자.)

도움코드

```
1 public class Exam3 {  
2     public static void main(String[] args) {  
3         int kor=70, eng=75, math=75;  
4         double avg;  
5         avg = (70+75+75)/3.;  
6         System.out.println("평균점수: "+avg);  
7     }  
8 }
```

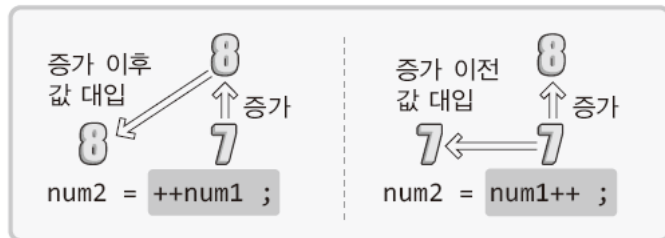
Problems @ Javadoc Declaration Console
<terminated> Exam3 [Java Application] C:\Program File
평균점수: 73.33333333333333

7. 연산자

❖ 증가, 감소 연산자

연산자	연산자의 기능	결합방향
++ (prefix)	피연산자에 저장된 값을 1 증가 예) val = ++n;	←
-- (prefix)	피연산자에 저장된 값을 1 감소 예) val = --n;	←

연산자	연산자의 기능	결합방향
++ (postfix)	피연산자에 저장된 값을 1 증가 예) val = n++;	←
-- (postfix)	피연산자에 저장된 값을 1 감소 예) val = n--;	←



❖ 예제

```
public class ExamPre{    // ExamPre.java
    public static void main( String[] args ) {
        int num1=7;
        int num2,num3;
        num2 = ++num1;
        System.out.println("++num1 : " + num2 );
        num1=7;
        num3 = num1++;
        System.out.println("num1++ : " + num3 );
    }
}
```

```
++num1 : 8
num1++ : 7
```


7. 연산자

❖ 관계 연산자

n1 = 7, n2 = 3

연산자	연산자의 기능	결합방향
<	예) n1 < n2 n1이 n2보다 작은가? false	→
>	예) n1 > n2 n1이 n2보다 큰가? true	→
<=	예) n1 <= n2 n1이 n2보다 같거나 작은가? false	→
>=	예) n1 >= n2 n1이 n2보다 같거나 큰가? true	→
==	예) n1 == n2 n1과 n2가 같은가? false	→
!=	예) n1 != n2 n1과 n2가 다른가? true	→

연산의 결과로 true or false 반환

❖ 예제 : 관계연산자 예제

```
public class Exam4 {    // Exam4.java
    public static void main( String[] args ) {

        System.out.println( 3 > 4 );

        System.out.println( 3 < 4 );

    }
}
```

false
true

7. 연산자

❖ 논리 연산자

- AND(&&), OR(||), NOT(!)
- 결과는 boolean 타입

a	b	a && b	a b	!a
false	false	false	false	true
false	true	false	true	
true	false	false	true	false
true	true	true	true	

❖ 예제 : 논리연산자 예제

```
public class Exam5 {    // Exam5.java
    public static void main( String[] args ) {
        int num1=10, num2=20;
        boolean result1 = (num1==10 && num2==20);
        boolean result2 = (num1<=12 || num2>=20);
        System.out.println("num==10 그리고 num2==20 : "+result1 );
        System.out.println("num<=12 또는 num2>=20 : "+result2 );
    }
}
```

num==10 그리고 num2==20 : true
num<=12 또는 num2>=20 : true

7.연산자

❖ 비트연산 진리표

비트 A	비트 B	비트 A & 비트 B
1	1	1
1	0	0
0	1	0
0	0	0

비트 A	비트 B	비트 A 비트 B
1	1	1
1	0	1
0	1	1
0	0	0

비트 A	비트 B	비트 A ^ 비트 B
1	1	0
1	0	1
0	1	1
0	0	0

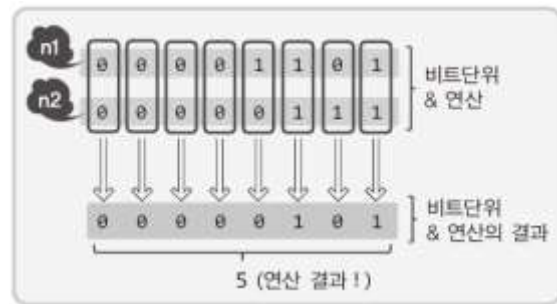
비트	~비트
1	0
0	1

~(NOT)

&(and)

|(OR)

^(XOR)



❖ 비트 쉬프트(Shift) 연산자

연산자	연산자의 기능	결합방향
<<	<ul style="list-style-type: none"> 피연산자의 비트 열을 왼쪽으로 이동 이동에 따른 빈 공간은 0으로 채움 예) $n \ll 2$; → n의 비트 열을 두 칸 왼쪽으로 이동 시킨 결과 반환 	→
>>	<ul style="list-style-type: none"> 피연산자의 비트 열을 오른쪽으로 이동 이동에 따른 빈 공간은 음수의 경우 1, 양수의 경우 0으로 채움 예) $n \gg 2$; → n의 비트 열을 두 칸 오른쪽으로 이동 시킨 결과 반환 	→
>>>	<ul style="list-style-type: none"> 피연산자의 비트 열을 오른쪽으로 이동 이동에 따른 빈 공간은 0으로 채움 예) $n \ggg 2$; → n의 비트 열을 두 칸 오른쪽으로 이동 시킨 결과 반환 	→

✓ 비트연산의 특징

- 왼쪽으로의 비트 열 이동은 2의 배수의 곱
- 오른쪽으로의 비트 열 이동은 2의 배수의 나눗셈

- 정수 2 → 00000010 → 정수 2
- $2 \ll 1 \rightarrow 00000100 \rightarrow$ 정수 4
- $2 \ll 2 \rightarrow 00001000 \rightarrow$ 정수 8
- $2 \ll 3 \rightarrow 00010000 \rightarrow$ 정수 16

{ INDEX }

1. 자바 시작하기
2. 자바 개발환경
3. 자바 프로그램작성
4. 예약어와 식별자
5. 변수와 자료형
6. 상수와 형변환
7. 연산자
8. 조건문
9. 반복문
10. 배열

* **콘솔 입력과 출력**

*.콘솔 입력과 출력

❖ 콘솔출력 : System.out.println과 System.out.print

- println 메소드는 출력 후에 개행을 한다.
- print 메소드는 출력 후에 개행을 하지 않는다.

```
public class OutEx {  
  
    public static void main(String[] args) {  
  
        System.out.print("안녕");  
        System.out.println("하세요");  
  
        System.out.println("안녕하세요");  
  
        String str01 = "안녕하세요";  
        System.out.println(str01);    //System.out.println("str01");  
  
        String str02 = "하세요";  
        System.out.println("안녕" + str02);  
    }  
}
```

안녕하세요
안녕하세요
안녕하세요
안녕하세요

*.콘솔 입력과 출력

❖ 콘솔출력 : System.out.println과 System.out.print

- 문자열 안에서 특별한 의미로 해석되는 문자를 가리켜 '이스케이프 시퀀스'라 한다.

- \n 개행
- \t 탭(Tab)
- \" 큰 따옴표(Quotation mark)
- \\ 역슬래쉬(Backslash)

\ == ₩

```
System.out.println("안녕₩n하세요");
System.out.println("₩"자바₩"수업중입니다.");
System.out.println("₩t탭(들여쓰기)입니다.");
System.out.println("₩₩역슬러시입니다.");
```

안녕
하세요
"자바"수업중입니다.
 탭(들여쓰기)입니다.
₩역슬러시입니다.

*.콘솔 입력과 출력

❖ 콘솔입력 : Scanner

❖ Scanner 클래스의 생성자!

- Scanner(File source)
- Scanner(InputStream source)
- Scanner(Readable source)
- Scanner(String source)



단, **import java.util.Scanner;** 선언이 필요합니다
⇒ package 와 class 사이 위치에 선언합니다.

Scanner 클래스는 단순히 키보드의 입력만을 목적으로 디자인된 클래스가 아니다.
스캐너 클래스는 다양한 리소스를 대상으로 입력을 받을 수 있도록 정의된 클래스이다.

```
Scanner sc = new Scanner(System.in);  
int num1 = sc.nextInt();  
System.out.println(num1);  
sc.close();
```

Scanner sc

- public boolean nextBoolean()
- public byte nextByte()
- public short nextShort()
- **public int nextInt()**
- public long nextLong()
- public float nextFloat()
- public double nextDouble()
- public String nextLine()

*.콘솔 입력과 출력

■ 예제 : Joy01.java

```
import java.util.Scanner;
public class Joy01 {
    public static void main(String arg[]) {
        String str;
        Scanner sc = new Scanner(System.in);
        System.out.print("이름은? ");
        str = sc.next();
        System.out.println(str+"님, 만나서 반가워요");
        sc.close();
    }
}
```

이름은? 김선린
김선린님, 만나서 반가워요

*.콘솔 입력과 출력

■ 예제 : Joy02.java

```
import java.util.Scanner;
public class Joy02 {
    public static void main(String arg[]) {
        int a = 39;
        double b = 170.3;
        String d= " 홍길동";

        Scanner sc = new Scanner(System.in);

        d = sc.next();          // 문자열
        a = sc.nextInt();       // 정수
        b = sc.nextDouble();    // 실수

    }
}
```

*.콘솔 입력과 출력

❖ 예제 : Ex01.java

태어난 년도를 입력받아서

현재 만 나이를 출력하는 프로그래밍을 작성하시오.

(예시)

태어난 년도? ** (입력)

현재 나이는 만 ** 살입니다.

*.콘솔 입력과 출력

❖ 예제 : Ex01.java

태어난 년도를 입력받아서

현재 만 나이를 출력하는 프로그래밍을 작성하시오.

도움코드

```
1 public class Ex01 {  
2     public static void main(String[] args) {  
3         int age;  
4         System.out.print("태어난 년도? ");  
5         age = new java.util.Scanner(System.in).nextInt();  
6         System.out.println("현재 나이는 만 "+(2018-age)  
7                               +"살입니다.");  
8     }  
9 }
```

태어난 년도? 2002
현재 나이는 만 16살입니다.

*.콘솔 입력과 출력

❖ 예제 : Ex02.java

국어,영어,수학 점수를 입력받아서

평균 출력하는 프로그래밍을 작성하시오.

(예시)

국어? ** (입력)

영어? ** (입력)

수학? ** (입력)

평균점수는 ** 점입니다.

*.콘솔 입력과 출력

❖ 예제 : Ex02.java

국어,영어,수학 점수를 입력받아서

평균 출력하는 프로그래밍을 작성하시오.

도움코드

```
1 import java.util.Scanner;
2 public class Ex02 {
3     public static void main(String[] args) {
4         int kor, eng, math, sum;
5         double avg;
6         Scanner sc = new Scanner(System.in);
7
8         System.out.print("국어? ");
9         kor = sc.nextInt();
10        System.out.print("영어? ");
11        eng = sc.nextInt();
12        System.out.print("수학? ");
13        math = sc.nextInt();
14        sum = kor + eng + math;
15        avg = (double)sum / 3;
16        System.out.print("평균점수는 ");
17        System.out.println(avg+"점입니다.");
18    }
19 }
```

국어? 90

영어? 95

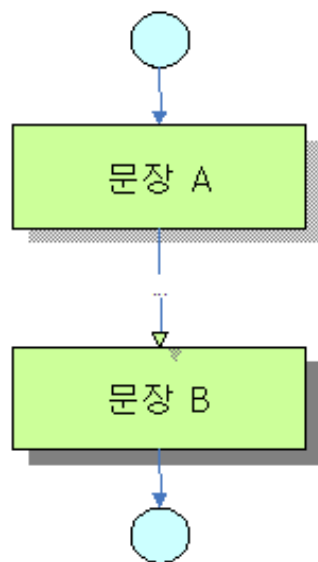
수학? 92

평균점수는 92.33333333333333점입니다.

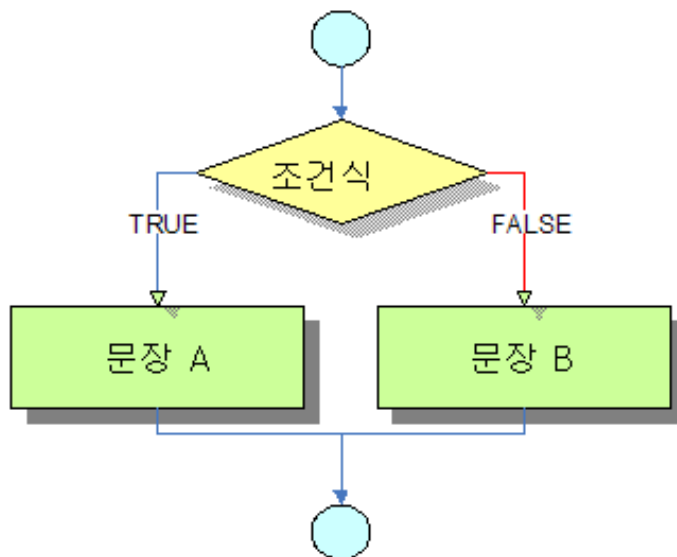
{ INDEX }

1. 자바 시작하기
2. 자바 개발환경
3. 자바 프로그램작성
4. 예약어와 식별자
5. 변수와 자료형
6. 상수와 형변환
7. 연산자
- 8. 조건문**
9. 반복문
10. 배열
- * 콘솔 입력과 출력

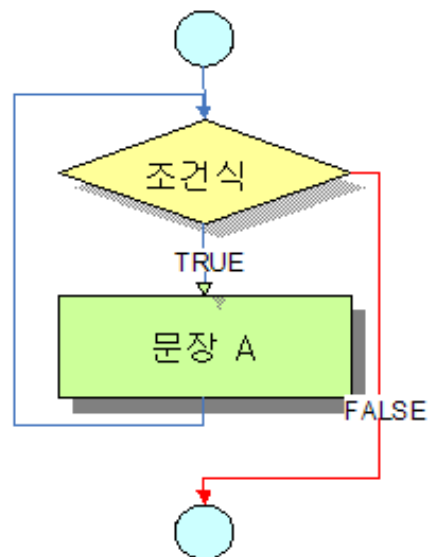
❖ 3가지 제어구조



순차 구조

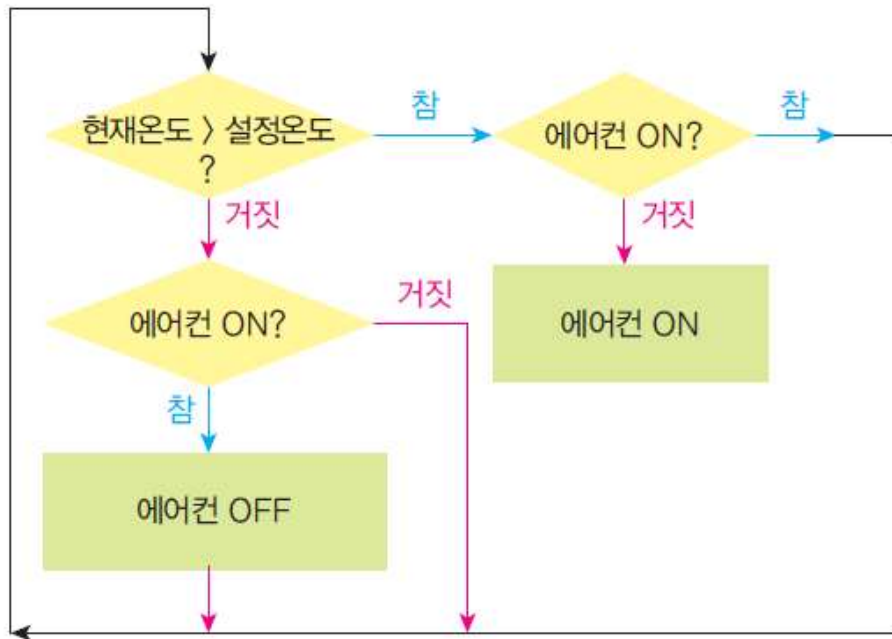


선택 구조



반복 구조

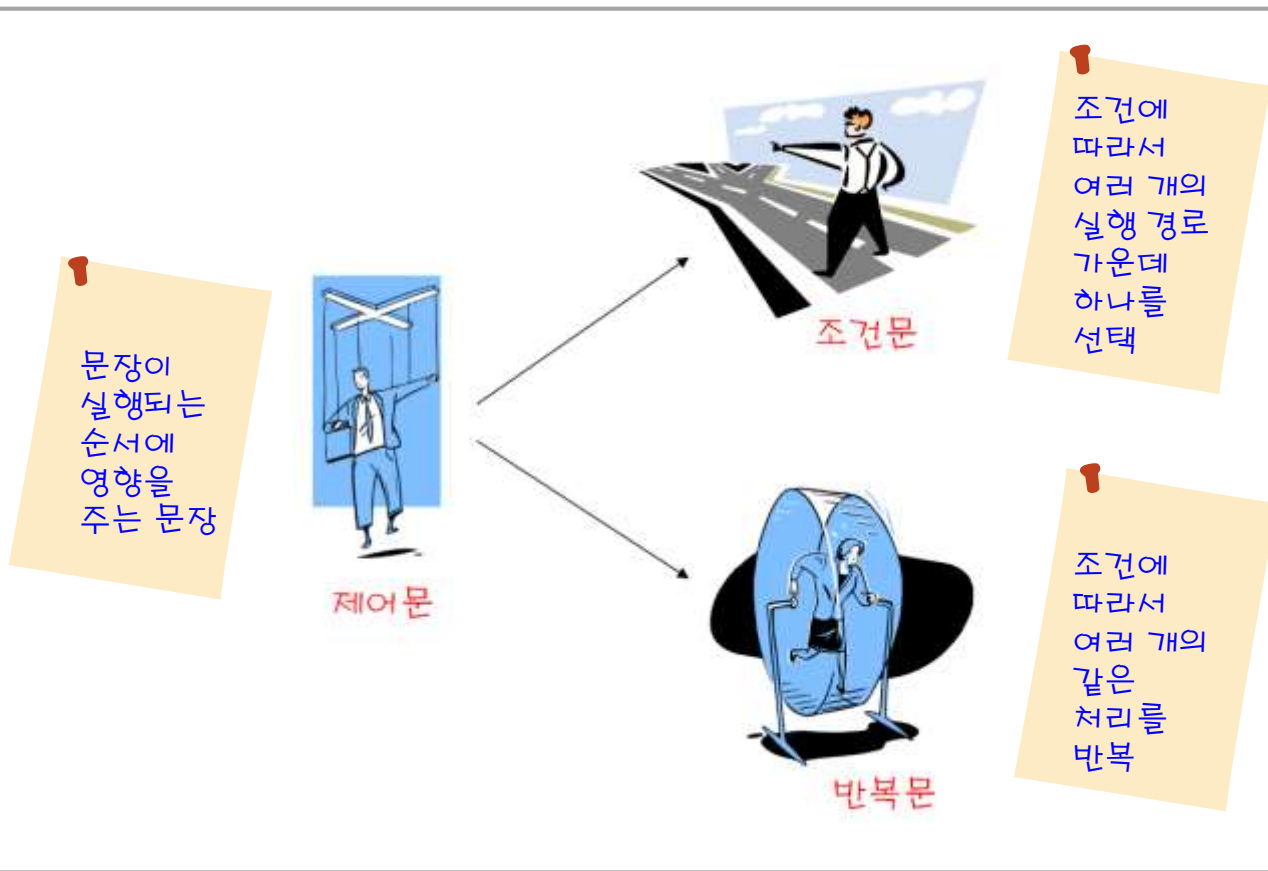
❖ 복잡한 프로그램이라면?



프로그램에서 문장은 기본적으로 순차적으로 실행됩니다. 하지만 복잡한 문제는 순차 실행만으로는 안 되고 조건문이나 반복문을 사용해야 합니다.

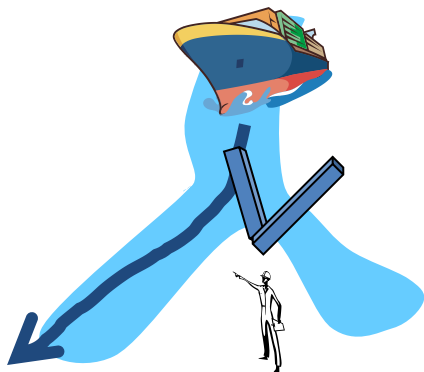


❖ 제어문 = 조건문과 반복문

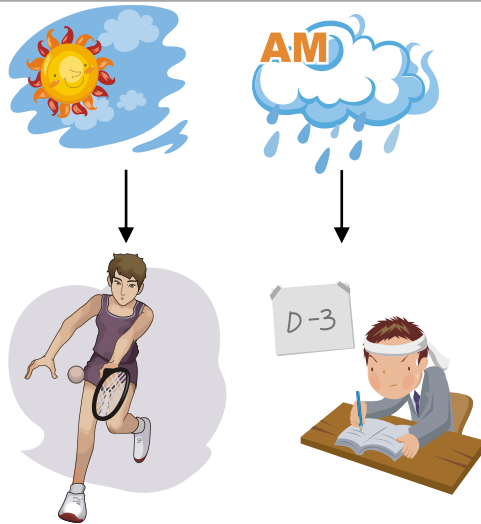
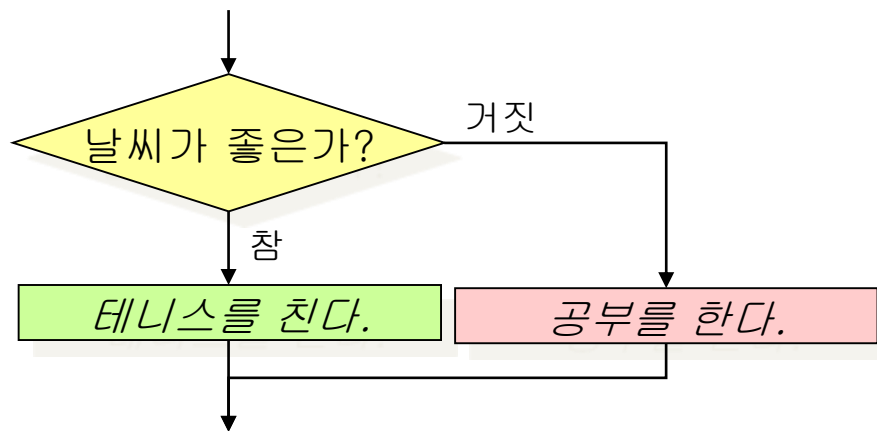


❖ 조건문 (예: 일상생활)

- 만약 비가 오지 않으면 테니스를 친다.
- 만약 결석이 1/3이상이면 F학점을 받는다.
- 만약 시간이 없는 경우에는 택시를 탄다.
- 만약 날씨가 좋고 공휴일이면 공원에 산책을 간다.
- 점수가 60점 이상이면 합격이고 그렇지 않으면 불합격이다.



❖ 조건문 (예: if – else문)



```
if( 조건식 )
```

```
    문장1;
```

```
else
```

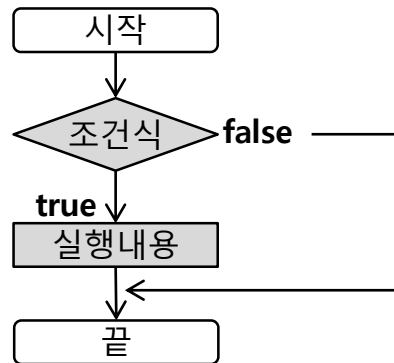
```
    문장2;
```

조건식이 참이면 실행된다.

조건식이 거짓이면 실행된다.

❖ if 조건문

```
if ( 조건식 ) {  
    /* 조건식이 true 이면 실행되는 영역 */  
}
```



■ 예제 : condition01.java

점수가 60점 이상이면 "합격입니다" 를 출력하세요

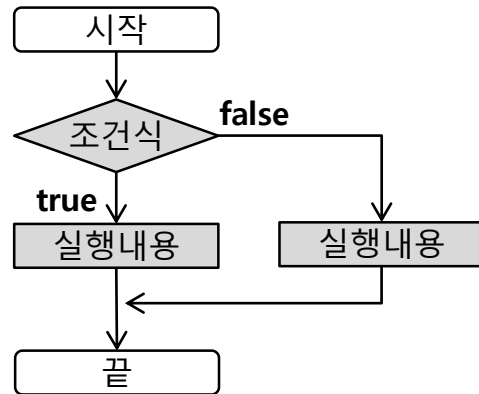
(예) 점수를 입력하세요 : 90
합격입니다.

도움코드

```
// package xx;
import java.util.Scanner;
public class Condition01 {
    public static void main(String arg[])
    {
        int num;
        Scanner sc = new Scanner(System.in);
        System.out.print("점수를 입력하세요: ");
        num = sc.nextInt();
        if(num >= 60 )
        {
            System.out.println("합격입니다");
        }
        sc.close();
    }
}
```


❖ if-else 조건문

```
if ( 조건식 ) {  
    /* 조건식이 true 이면 실행되는 영역 */  
}  
else {  
    /* 조건식이 false 이면 실행되는 영역 */  
}
```



■ 예제(계속) : condition01.java

점수가
60점 이상이면 "합격입니다."
60점 미만이면 "불합격입니다."
를 출력하세요

(예) 점수를 입력하세요 : 59
불합격입니다.

도움코드

```
// package xx;
import java.util.Scanner;
public class Condition01 {
    public static void main(String arg[])
    {
        int num;
        Scanner sc = new Scanner(System.in);
        System.out.print("점수를 입력하세요: ");
        num = sc.nextInt();
        if(num >= 60 )
        {
            System.out.println("합격입니다");
        }else {
            System.out.println("불합격입니다");
        }
        sc.close();
    }
}
```

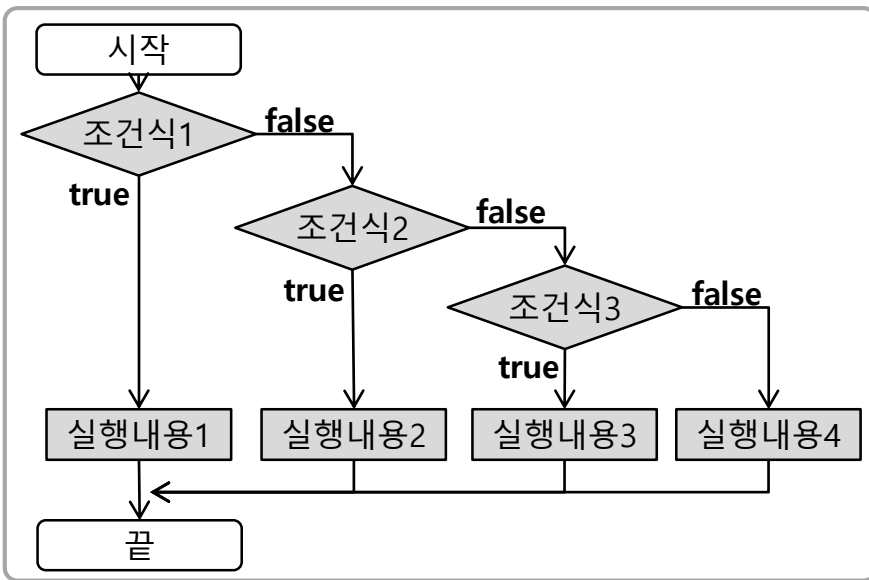
8.조건문

❖ if-else if-else 조건문

```

if ( 조건식1 ) {
    /* 조건식1 이 true 이면 실행되는 영역 */
} else if (조건식2) {
    /* 조건식2 가 true 이면 실행되는 영역 */
} else if (조건식3) {
    /* 조건식3 이 true 이면 실행되는 영역 */
} else {
    /* 위의조건이 모두해당되지 않으면 실행되는 영역 */
}

```



■ 예제 : Condition02.java

숫자가 0보다 크면 "양수", 영보다 작으면 "음수"를 출력하세요. 만약 0이면, 0을 출력하세요.

(예) 숫자를 입력해 주세요 3

양수

```
1 public class Condition02 {  
2     public static void main(String[] args) {  
3         int num;  
4         System.out.print("숫자를 입력해주세요 ");  
5         num = new java.util.Scanner(System.in).nextInt();  
6         if(num>0) System.out.println("양수");  
7         else if(num==0) System.out.println("0");  
8         else System.out.println("음수");  
9     }  
10 }
```

■ 예제 : Condition03.java

과목 code값이

1 이면 "자바 입니다."

2 이면 "C언어 입니다."

3 이면 "C++언어 입니다."

4 이면 "파이썬 입니다."

나머지는 "과목이 없습니다."

를 출력하세요

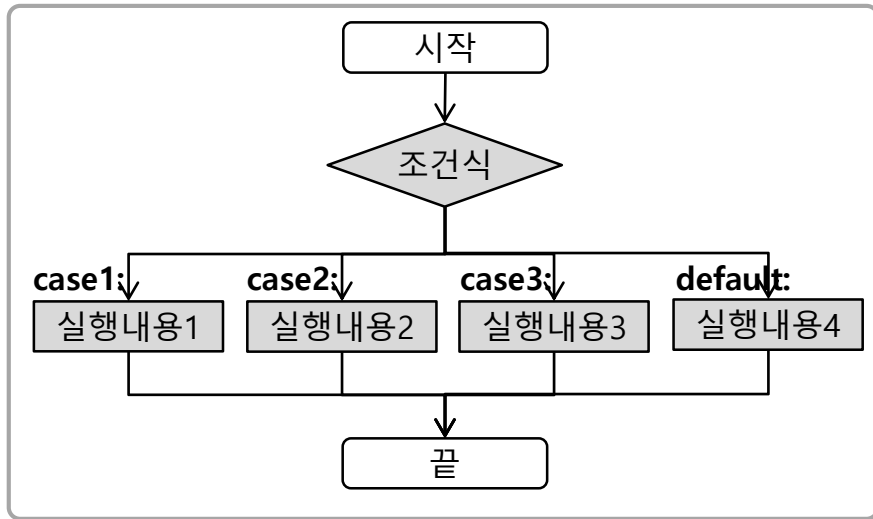
(예) 과목코드를 입력해서 과목명을 확인하세요(1.자바 2.C 3.C++ 4.파이썬) : **1**

자바 입니다.

```
1 public class Condition03 {
2     public static void main(String[] args) {
3         int inp;
4         System.out.println("과목코드를 입력해서 과목명을 확인하세요"
5                             + "(1.자바 2.C 3.C++ 4.파이썬) : ");
6         inp = new java.util.Scanner(System.in).nextInt();
7         if(inp==1) System.out.println("자바 입니다.");
8         else if(inp==2) System.out.println("C언어 입니다.");
9         else if(inp==3) System.out.println("C++언어 입니다.");
10        else if(inp==4) System.out.println("파이썬 입니다.");
11        else System.out.println("과목이 없습니다.");
12    }
13 }
```

❖ switch-case 조건문

```
switch (변수) {  
    case 값1 :  
        /* 변수값이 값1 일때 실행내용*/  
        break;  
    case 값2 :  
        /* 변수값이 값2 일때 실행내용*/  
        break;  
    case 값3 :  
        /* 변수값이 값3 일때 실행내용*/  
        break;  
    default :  
        /* 해당내용이 없을때 실행내용*/  
        break;  
}
```



■ 예제 : Switch01.java

입력한 월이
1월이면 "31일"
2월이면 "28일"
3월이면 "31일"
4월이면 "30일"
5월이면 "31일"
6월이면 "30일"
7월이면 "31일"
8월이면 "31일"
9월이면 "30일"
10월이면 "31일"
11월이면 "30일"
12월이면 "31일"
을... 출력하세요

(예) "월"을 입력해주세요 3
31일

▪ switch

```
switch (month) {  
    case 2:  
        days = 28;  
        break;  
  
    case 4:  
    case 6:  
    case 9:  
    case 11:  
        days = 30;  
        break;  
  
    default:  
        days = 31;  
        break;  
}
```

▪ if

```
if (month == 2) {  
    days = 28;  
} else if ((month == 4) || (month == 6) || (month == 9) || (month == 11)) {  
    days = 30;  
} else {  
    days = 31;  
}
```

if-else 문의 조건식 부분이 모두 == 로 표현될때 switch문을 사용

[조건문1~5] 예제: 각자 해결보세요

❖ 조건문 예제

[조건문1] 수를 입력받아 "짝수"인지 "홀수"인지 판별하기

숫자? 77
홀수

[조건문2] 수를 입력받아 "5의 배수"인지 "5의 배수가 아닌지" 판별하기

숫자? 77
5의배수 아님

[조건문3] 수를 입력받아 "3의 배수"인지 "5의 배수"인지 "3과 5의 배수 둘 다 아닌"지 판별하기

숫자? 77
3과 5의 배수 둘 다 아님

❖ 조건문 예제

[조건문4] 국어,영어,수학 점수 입력받아서 평균이 80점 이상이면 "합격", 아니면 "불합격" 출력하기

국어? 80

영어? 85

수학? 90

합격

[조건문5] 세개의 수를 입력받아 가장 큰 수 한 개를 화면에 출력하기

숫자1? 77

숫자2? 60

숫자3? 45

가장 큰수는 77입니다.

❖ 조건문 예제

[조건문6]

입력된 점수가

90점 이상~100점 이면 "A"

80점 이상~89점 이면 "B"

70점 이상~79점 이면 "C"

60점 이상~69점 이면 "D"

60점 미만이면 "F"

을 화면에 출력하세요 (단, 입력값이 0 ~ 100점 까지만 입력될 것이라고 가정합니다. – 예외처리 안함)

(예) 점수를 입력하세요 : 77

C

❖ 조건문 예제 [조건문6] 도움말

이 페이지를 보지 않고 한번 시도해보세요!

```
import java.util.Scanner;
public class Exam6 {
    public static void main(String[] args) {
        // 정수형 변수 선언

        // 프롬프트 출력 (예시) System.out.print("점수를 입력하세요: ");

        // Scanner클래스 타입 참조변수 선언 및 객체 생성
        // nextInt()메소드 호출해서 정수형 변수에 값 대입

        // 정수형변수값이 90점 이상 100점 이하이면 → "A" 출력
        // 정수형변수값이 80점 이상 90점 미만이면 → "B" 출력
        // 정수형변수값이 70점 이상 80점 미만이면 → "C" 출력
        // 정수형변수값이 60점 이상 70점 미만이면 → "D" 출력
        // 정수형변수값이 0점 이상 60점 미만이면 → "F" 출력

    }
}
```

[조건문6] 예제

```
1 import java.util.Scanner;
2 public class Exam6 {
3     public static void main(String[] args) {
4         int num;
5         System.out.print("점수를 입력하세요: ");
6         Scanner sc = new Scanner(System.in);
7         num = sc.nextInt();
8         if (num <= 100 && num >= 90)
9             System.out.println("A");
10        else if (num >= 80)
11            System.out.println("B");
12        else if (num >= 70)
13            System.out.println("C");
14        else if (num >= 60)
15            System.out.println("D");
16        else
17            System.out.println("F");
18        sc.close();
19    }
20 }
```

{ INDEX }

1. 자바 시작하기
2. 자바 개발환경
3. 자바 프로그램작성
4. 예약어와 식별자
5. 변수와 자료형
6. 상수와 형변환
7. 연산자
8. 조건문
- 9. 반복문**
10. 배열
- * 콘솔 입력과 출력

9.반복문

❖ 반복문

- while 반복문
- for 반복문
- do~while 반복문

❖ 기타

- Break
- continue

9.반복문

❖ 화면에 출력해봅시다

- project : day04
- package : day04
- 소스파일(class) : ForWhileEx.java

(생각해보기) 만약

I like Java 100 까지 계속 출력해야된다면?

```
I like Java 0
```

```
I like Java 1
```

```
I like Java 2
```

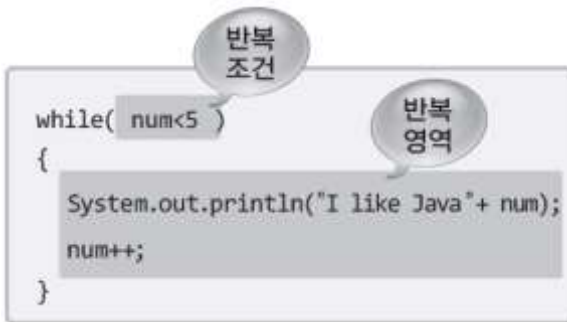
```
I like Java 3
```

```
I like Java 4
```

9.반복문

❖ while 반복문

```
while ( 조건식 ) {
    /* 조건식이 true이면 실행되는 영역*/
}
```



```
I like Java 0
I like Java 1
I like Java 2
I like Java 3
I like Java 4
```

```
package day04;
public class ForWhileEx01 {

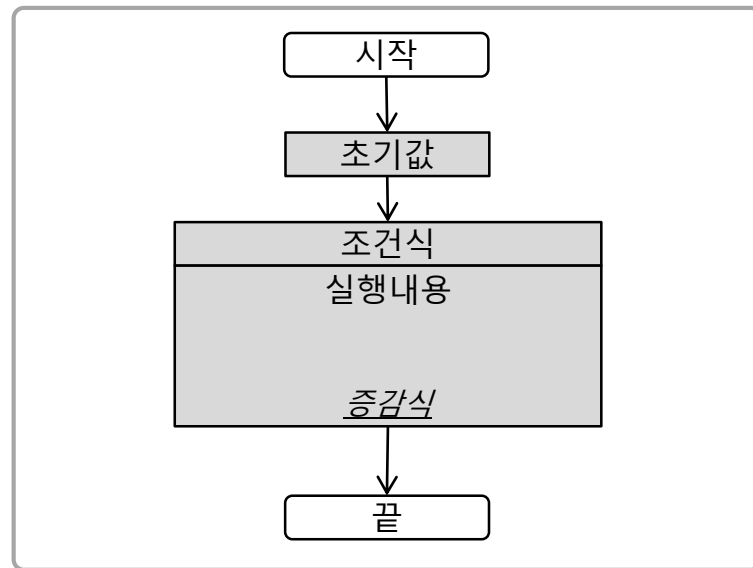
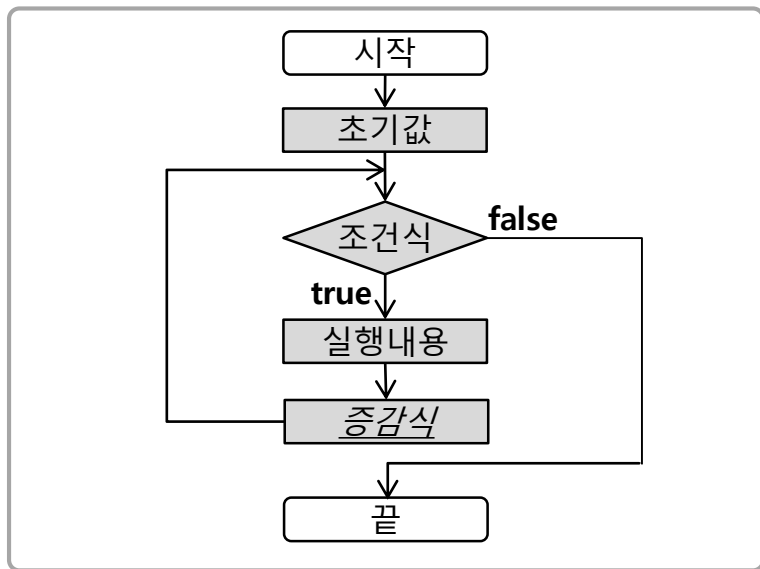
    public static void main(String[] args){

        int num=0;

        while( num<5 ) // while(num<=4)
        {
            System.out.println("I like Java" + num);
            num++;      // num = num + 1;
        }
    }
}
```

while 문은 한번도 실행되지 않을 수 있다!

❖ while 반복문



■ while 반복문 예제

숫자(정수)를 입력받아 입력한 숫자(단)의 구구단을 출력하세요.

단을 입력해주세요 3

3*1=3
3*2=6
3*3=9
3*4=12
3*5=15
3*6=18
3*7=21
3*8=24
3*9=27

```
package day04;
import java.util.Scanner;
public class ForWhileEx02 {

    public static void main(String[] args){

        Scanner kb= new Scanner(System.in);

        System.out.print("단을 입력해주세요 ");
        int dan = kb.nextInt();

        int i = 1;
        while( i <= 9 ) {

            System.out.println( dan + "*" + i + "=" + (dan*i) );

            i++;
        }
        kb.close(); // 스트림 닫기
    }
}
```

❖ while 반복문

비교

for 반복문

```
1. int num=0;
2. while( num<5 )
{
    System.out.println(num);
3. num++;
}
```

```
1. for( 1. int num=0 ; 2. num<5 ; 3. num++ )
{
    System.out.println( num );
}
```

```
for( 초기화; 조건식; 증감식 ){
    조건식이 맞을때 실행내용
}
```

1. → 반복의 횟수를 세기 위한 변수

2. → 반복의 조건

3. → 반복의 조건을 무너뜨리기 위한 연산

❖ for 반복문의 실행흐름

첫 번째 루프의 흐름

1 → 2 → 3 → 4 [i=1]

두 번째 루프의 흐름

2 → 3 → 4 [i=2]

세 번째 루프의 흐름

2 → 3 → 4 [i=3]

네 번째 루프의 흐름

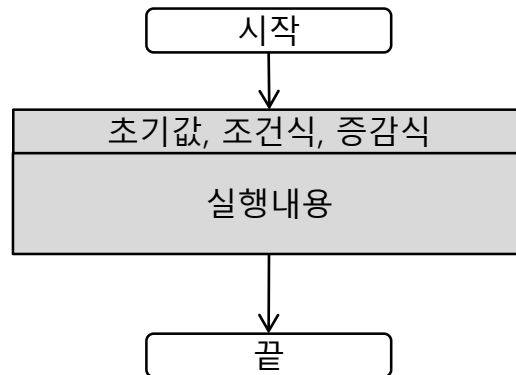
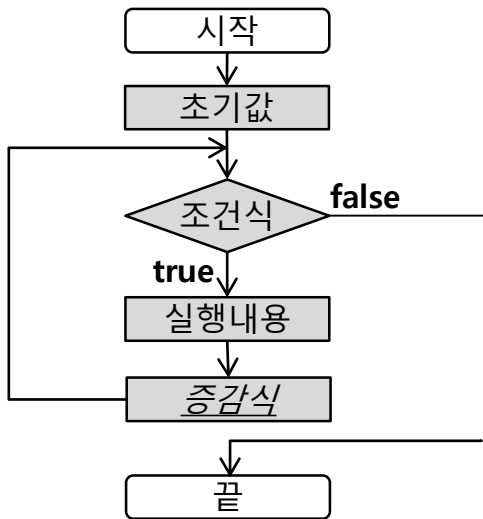
2 [i=3] 따라서 탈출!

```
for( 1.int i=0 ; 2.i<3 ; 4.i++ )  
{ 3.System.out.println("...");  
}
```

```
I like Java 0  
I like Java 1  
I like Java 2
```

❖ for 반복문

```
for( 초기화; 조건식; 증감식 ){  
    조건식이 맞을때 실행내용  
}
```



■ for 반복문 예제

숫자(정수)를 입력받아 입력한 숫자(단)의 구구단을 출력하세요.

단을 입력해주세요 3

3*1=3
3*2=6
3*3=9
3*4=12
3*5=15
3*6=18
3*7=21
3*8=24
3*9=27

```
package day04;
import java.util.Scanner;
public class ForWhileEx03 {

    public static void main(String[] args){

        Scanner kb= new Scanner(System.in);

        System.out.print("단을 입력해주세요 ");
        int dan = kb.nextInt();

        for( int i=1 ; i<=9 ; i++ ) {

            System.out.println( dan+"*"+i+"="+ (dan*i) );

        }

        kb.close();

    }
}
```


9. 반복문

do~while 문은 최소 한번은

실행이 된다!

❖ do~while 반복문

```
do
{
    System.out.println("I like Java"+ num);
    num++;
} while( num<5 );
```

반복 영역

반복 조건

ForWhileEx04.java

사용자의 숫자를 입력받아 더하는 프로그램을 작성하세요 (단, 0을 입력하면 종료하게 하자)

숫자를 입력하세요. [0이면 종료]

3

합계: 3

5

합계 : 8

7

합계 :15

0

종료

```
package day04;
import java.util.Scanner;
public class ForWhileEx04 {
    public static void main(String[] args) {

        Scanner kb= new Scanner(System.in);
        int sum =0;
        int num;

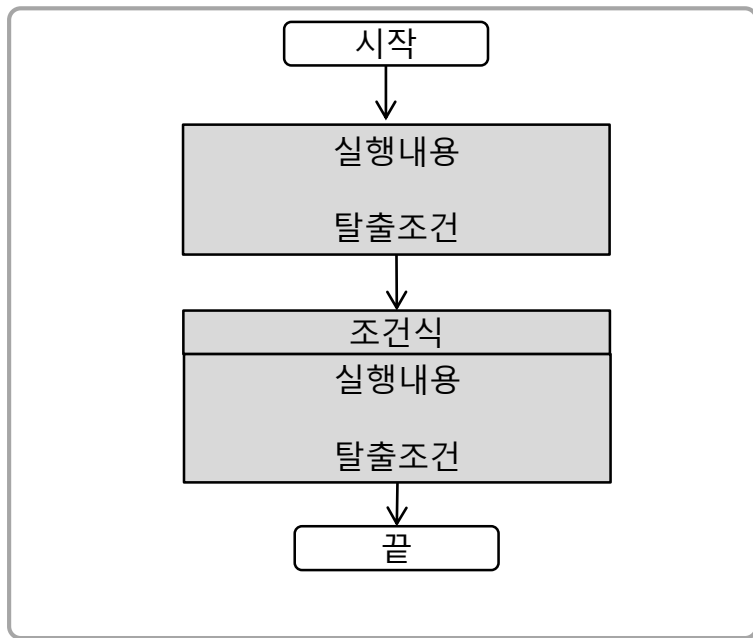
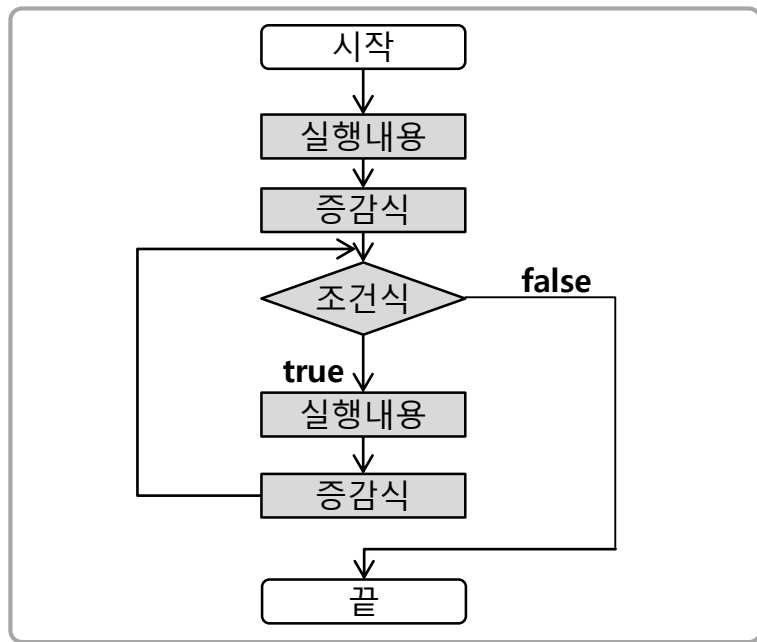
        do{
            System.out.println("숫자를 입력하세요.[0이면 종료]");
            num = kb.nextInt();
            sum = sum+num;           // sum+=num; 와 같은 뜻
            System.out.println("합계: "+sum);

        }while(num !=0);

        System.out.println("종료");
        kb.close();

    }
}
```

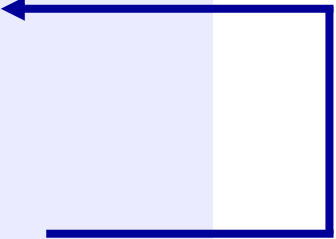
❖ do~while 반복문



❖ continue, break


아래쪽 무시하고
반복문 계속

```
while( true ) {  
    ...  
    if( x < 0 ) {  
        continue;  
    }  
    ...  
}
```



반복문 그만

```
while( true ) {  
    ...  
    if( x < 0 ) {  
        break;  
    }  
    ...  
}  
...
```



❖ while반복문 vs for반복문

반복의 횟수를 알 수 없을때 **while**반복문

반복의 횟수를 알 수 있을때 **for**반복문

4의 배수이자 6의 배수인 가장 작은 정수 찾기

```
package day04;
public class ForWhileEx05 {

    public static void main(String[] args){

        int num=1;

        while(true){

            if( num%4==0 && num%6==0 ){

                break; // 반복문 탈출(반복그만!)
            }

            num++;

        }

        Sytem.out.println(num);

    }
}
```

❖ while반복문 vs for반복문

반복의 횟수를 알 수 없을때 : while반복문

반복의 횟수를 알 수 있을때 : for반복문

아래와 같이 출력되는 구구단을 출력하세요

2*1 = 2
2*2 = 4
2*3 = 6

[중략]

9*7 = 63
9*8 = 72
9*9 = 81

아래와 같이 출력되는 프로그램을 작성하세요

*
**

[구구단] 예제
(2~9단 출력)

```
1 public class Gugudan {  
2     public static void main(String[] args) {  
3         int x, y;  
4         for(x=2;x<=9;x++) {  
5             for(y=1;y<=9;y++)  
6                 System.out.println(x+"*"+y+" = "+(x*y));  
7                 System.out.println();  
8             }  
9         }  
10 }
```

[삼각형 별찍기] 예제

```
1 public class Star {  
2     public static void main(String[] args) {  
3         int x,y;  
4         for(x=0;x<6;x++) {  
5             for(y=0;y<=x;y++)  
6                 System.out.print("*");  
7             System.out.println();  
8         }  
9     }  
10 }
```

```
*  
**  
***  
****  
*****  
*****
```

9.반복문

❖ 반복문 종합문제 : 숫자 맞추기 게임

숨겨진 숫자를 맞추는 게임을 만들어봅시다.
(단, 숨겨진 숫자는 매 게임마다 값이 달라짐)

0~99사이 임의의 숫자가 미리 숨겨져 있고,
이 숫자를 맞추는 게임입니다. (맞추면 종료)

숨겨진 숫자가 82이라고 가정해봅시다.
50이 입력되면 "더 높게",
85가 입력되면 "더 낮게"라고 출력합니다.

숫자를 맞추면 몇 번만에 맞췄는지 출력합니다.

(예시)

```
### 숫자 맞추기 게임 ###  
0~99 중 숫자 하나를 숨겼습니다. 맞춰보세요.  
  
1번째 도전>> 50  
더 높게  
2번째 도전>> 80  
더 높게  
3번째 도전>> 90  
더 낮게  
4번째 도전>> 85  
더 낮게  
5번째 도전>> 83  
더 낮게  
6번째 도전>> 82  
6번째에 맞추셨습니다.  
## 게임 종료 ##
```


9.반복문

❖ 도움말

1. 임의의 정수 생성 : Random 클래스 사용하기

```
import java.util.Random;
```

```
Random rand = new Random( );
```

```
int num = rand.nextInt(100);
```

```
// 0~99까지 임의의 정수가 생성됨.
```

2. 숫자 입력 : Scanner 클래스 사용하기

```
import java.util.Scanner;
```

```
Scanner sc = new Scanner(System.in);
```

```
int input;
```

```
input = sc.nextInt( );
```

숫자 맞추기 게임

0~99 중 숫자 하나를 숨겼습니다. 맞춰보세요.

1번째 도전>> 50

더 높게

2번째 도전>> 80

더 높게

3번째 도전>> 90

더 낮게

4번째 도전>> 85

더 낮게

5번째 도전>> 83

더 낮게

6번째 도전>> 82

6번째에 맞추셨습니다.

게임 종료

❖ 도움코드

```
import java.util.Random;
import java.util.Scanner;
public class Project02 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Random rand = new Random();
        int num = rand.nextInt(100);
        int input, cnt;

        System.out.println("### 숫자 맞추기 게임 ###");
        System.out.println("0~99 중 숫자 하나를 숨겼습니다. 맞춰보세요.");
        System.out.println();
        cnt = 1; // 도전횟수를 기억하는 정수형 변수
        while(true){
            // 적절한 코드들을 작성해 봅시다
        }
        System.out.println("## 게임 종료 ##");
    }
}
```

```
### 숫자 맞추기 게임 ###
0~99 중 숫자 하나를 숨겼습니다. 맞춰보세요.

1번째 도전>> 50
더 높게
2번째 도전>> 80
더 높게
3번째 도전>> 90
더 낮게
4번째 도전>> 85
더 낮게
5번째 도전>> 83
더 낮게
6번째 도전>> 82
6번째에 맞추셨습니다.
## 게임 종료 ##
```

도움코드

```
1 import java.util.Random;
2 import java.util.Scanner;
3 public class Project02 {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6         Random rand = new Random();
7         int num = rand.nextInt(100), input, cnt = 1;
8         System.out.println("### 숫자 맞추기 게임 ###");
9         System.out.println("0~99 중 숫자 하나를 숨겼습니다. 맞춰보세요.");
10        System.out.println();
11        while(true){
12            System.out.print(cnt+"번째 도전>> ");
13            input = sc.nextInt();
14            if(input==num) {
15                System.out.println(cnt+"번째에 맞추셨습니다.");
16                break;
17            }else if(input>num) System.out.println("더 낮게");
18            else System.out.println("더 높게");
19            cnt++;
20        }
21        System.out.println("## 게임 종료 ##");
22        sc.close();
23    }
24 }
```

{ INDEX }

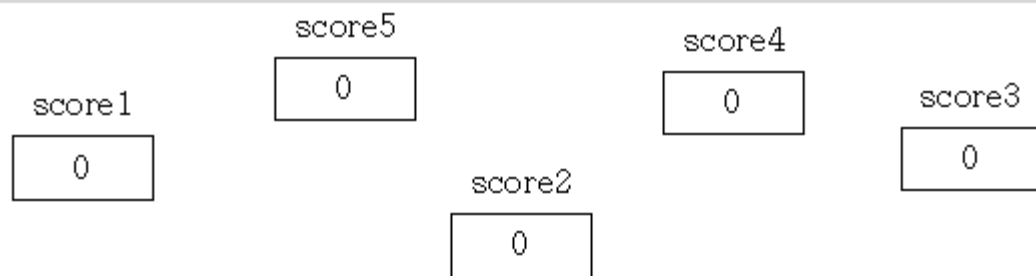
1. 자바 시작하기
2. 자바 개발환경
3. 자바 프로그램작성
4. 예약어와 식별자
5. 변수와 자료형
6. 상수와 형변환
7. 연산자
8. 조건문
9. 반복문
- 10. 배열**

배열 Array

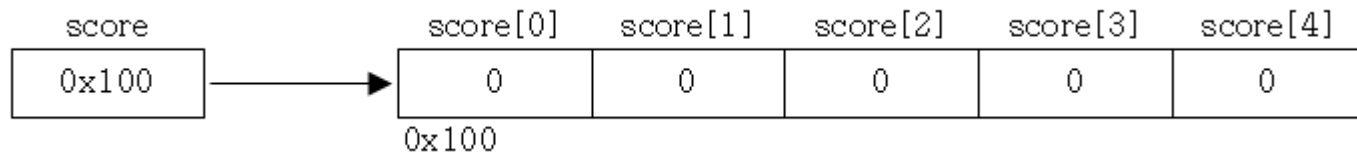
❖ 배열

서로 같은 자료형을 갖는 자료들을 묶은 하나의 집합

```
int score1=0, score2=0, score3=0, score4=0, score5=0 ;
```



```
int[] score = new int[5]; // 5개의 int 값을 저장할 수 있는 배열을 생성한다.
```



10. 배열

❖ 배열

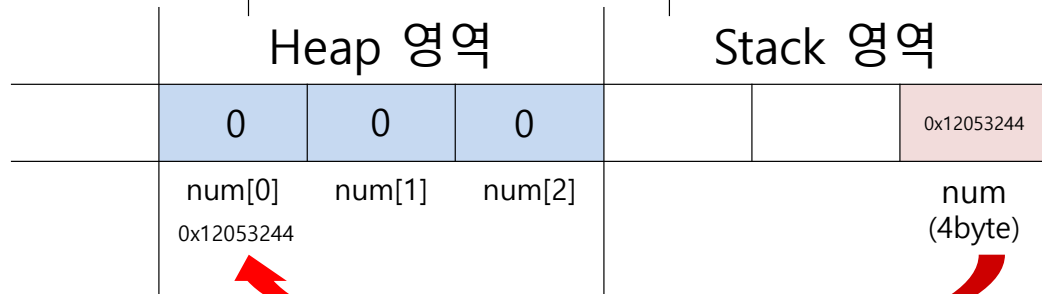
1. 배열 선언

```
int [ ]num; (or int num[ ];)
```



2. 배열 생성

```
num = new int[3];
```



3. 배열 초기화

```
num[0] = 10;
```

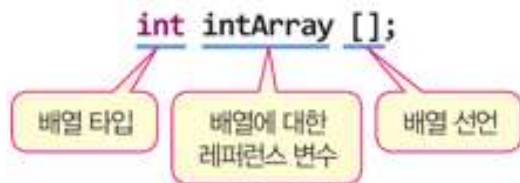


10. 배열

❖ 배열

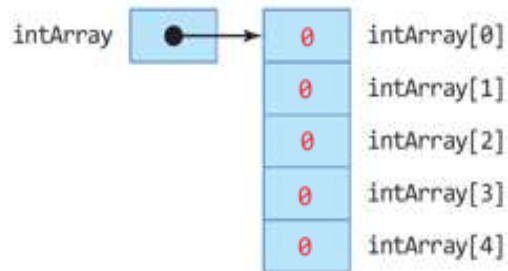
1. 배열 선언

```
int intArray[ ];
```



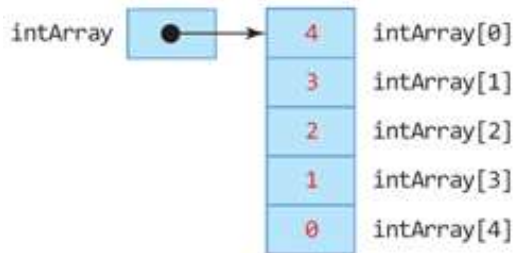
2. 배열 생성

```
intArray=new int[5];
```



3. 배열 초기화

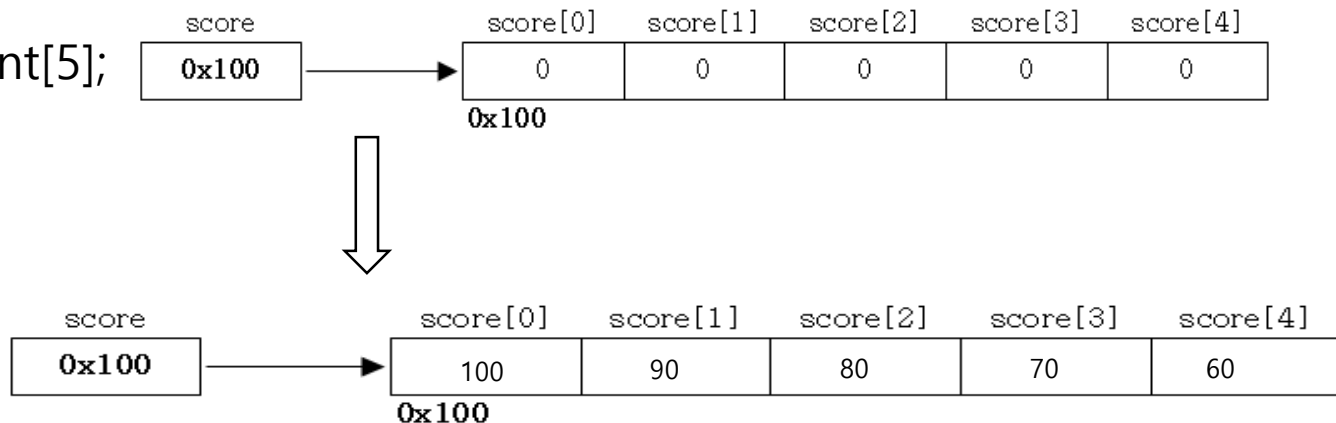
```
intArray=new int[]{4,3,2,1,0}
```



❖ 배열의 초기화

배열의 초기화

```
int[] score=new int[5];  
score[0]=100;  
score[1]=90;  
score[2]=80;  
score[3]=70;  
score[4]=60;
```



❖ 배열의 초기화

배열의 초기화

```
int[] score = { 100, 90, 80, 70, 60};           // 1번  
int[] score = new int[]{ 100, 90, 80, 70, 60}; // 2번
```

```
int[] score;  
score = { 100, 90, 80, 70, 60}; // 에러 발생!!!
```

```
int[] score;  
score = new int[]{ 100, 90, 80, 70, 60}; // OK
```

❖ 배열의 활용

배열에 값을 저장하고 읽어오기

```
score[3] = 100;          // 배열 score의 4번째 요소에 100을 저장한다.  
int value = score[3];    // 배열 score의 4번째 요소에 저장된 값을 읽어서 value에 저장.
```

‘배열이름.length’는 배열의 크기를 알려준다. (상수값)

```
int[] score = { 100, 90, 80, 70, 60, 50 };
```

```
for(int i=0; i < 6; i++) {  
    System.out.println(score[i]);  
}
```



```
for(int i=0; i < score.length; i++) {  
    System.out.println(score[i]);  
}
```

❖ 실습예제1

ArrayEx1.java

다음과 같이 배열 test를 선언 및 초기화하고,
오른쪽 예시와 같이 출력되도록
프로그램을 작성해봅시다.

	[0]	[1]	[2]	[3]	[4]
test	9	4	3	8	2

(예시)

역순출력: 2 8 3 4 9
총합 = ?
짝수의 개수 = ?
홀수의 개수 = ?

(참고)

위에서 ? 부분을 계산해서....
나타내보세요. ^_^

실습예제1

```
1 public class ArrayEx1 {  
2     public static void main(String[] args) {  
3         int[] test = new int[] {9,4,3,8,2};  
4         int oddNum = 0, evenNum = 0, sum = 0;  
5         System.out.print("역순출력: ");  
6         for(int x = test.length-1; x>=0; x--) {  
7             System.out.print(test[x]+" ");  
8             if(x%2==0) evenNum++;  
9             else oddNum++;  
10            sum += test[x];  
11        }  
12        System.out.println("\n총합 = "+sum);  
13        System.out.println("짝수의 개수 = "+evenNum);  
14        System.out.println("홀수의 개수 = "+oddNum);  
15    }  
16 }
```

역순출력: 2 8 3 4 9
총합 = 26
짝수의 개수 = 3
홀수의 개수 = 2

❖ 실습예제2

ArrayEx2.java

(예시)

1. 인원을 입력받아 배열을 생성함
2. 각 학생별 점수를 입력받아 배열에 차례대로 저장함
3. 배열에 저장된 점수의 평균을 구하여 화면에 출력함

(참고) 키보드로부터 입력받을 때...

```
Scanner sc = new Scanner(System.in);  
int num = sc.nextInt( );
```

```
인원을 입력하세요: 4  
1번째 학생 점수: 100  
2번째 학생 점수: 90  
3번째 학생 점수: 95  
4번째 학생 점수: 97  
avg: 95.5
```

실습예제2

```
1 import java.util.Scanner;
2 public class ArrayEx2 {
3     public static void main(String[] args) {
4         int array[], sum=0; double avg;
5         Scanner sc = new Scanner(System.in);
6         System.out.print("인원을 입력하세요: ");
7         array = new int[sc.nextInt()];
8         for(int x=0; x<array.length; x++) {
9             System.out.print((x+1)+"번째 학생 점수: ");
10            array[x] = sc.nextInt();
11            sum += array[x];
12        }
13        avg = (double)sum / array.length;
14        System.out.println("avg: "+ avg);
15        sc.close();
16    }
17 }
```

이차원 배열

❖ 이차원 배열

`int[][] scores = new int[2][3];` 수학의 행렬과 비슷한 자료구조

【2 x 3 행렬의 구조】

↓
열

	0	1	2
0 → 행	(0,0)	(0,1)	(0,2)
1	(1,0)	(1,1)	(1,2)

❖ 이차원 배열

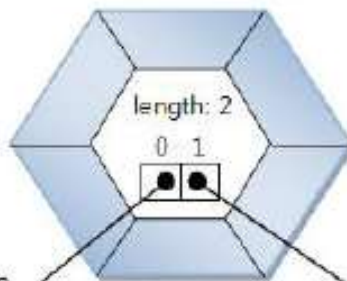
`int[][] scores = new int[2][3];` 중첩배열방식으로 구현

`int[][] scores = new int[2][3];`

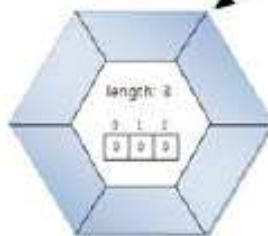


힙(heap) 영역

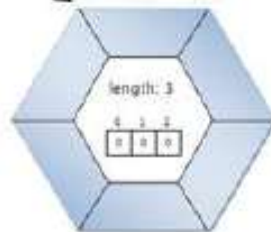
int 타입 배열 A



int 타입 배열 B



int 타입 배열 C

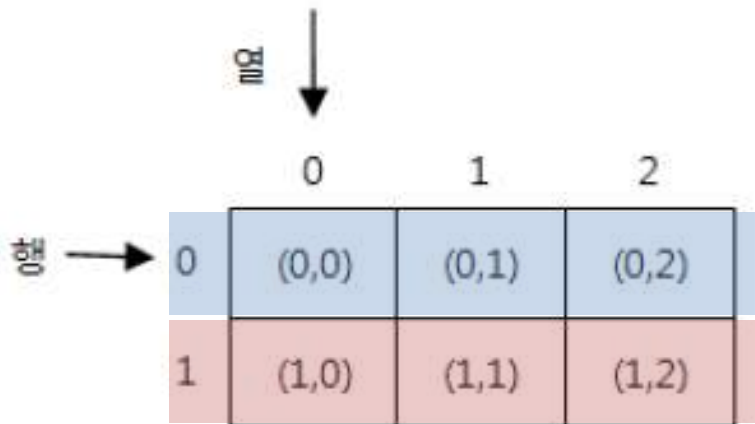


`scores.length` // 2 (배열 A의 길이)
`scores[0].length` // 3 (배열 B의 길이)
`scores[1].length` // 3 (배열 C의 길이)

❖ 이차원 배열

`int[][] scores = new int[2][3];` 수학의 행렬과 비슷한 자료구조

【2 x 3 행렬의 구조】



		0	1	2
행 →	0	(0,0)	(0,1)	(0,2)
	1	(1,0)	(1,1)	(1,2)

❖ 이차원 배열 실습예제3

```
public class ArrayEx03 {  
    public static void main(String[] args) {  
        int[][] a = new int[2][3]; // int a[][] = new int[2][3];  
        a[0][0]=1; a[0][1]=2; a[0][2]=3;  
        a[1][0]=4; a[1][1]=5; a[1][2]=6;  
        for(int x=0; x < a.length ; x++) {  
            for(int y=0; y < a[x].length ; y++)  
                System.out.print(a[x][y]+" ");  
            System.out.println();  
        }  
    }  
}
```

1	2	3
4	5	6

❖ 이차원 배열

```
public class ArrayEx03 {  
    public static void main(String[] args) {  
        int[][] a = { {1,2,3}, {4,5,6}, {7,8,9} };  
        // a[0][0]=1; a[0][1]=2; a[0][2]=3;  
        // a[1][0]=4; a[1][1]=5; a[1][2]=6;  
  
        for(int x=0; x < a.length ; x++) {  
            for(int y=0; y < a[x].length ; y++)  
                System.out.print(a[x][y]+" ");  
            System.out.println();  
        }  
    }  
}
```

1	2	3
4	5	6
7	8	9

❖ 이차원 배열

```
public class ArrayEx03 {  
    public static void main(String[] args) {  
        int[][] a;  
        a = new int[][] { {1,2,3},{4,5,6},{7,8,9} };  
        // int[][] a = { {1,2,3}, {4,5,6}, {7,8,9} };  
        // a = { {1,2,3},{4,5,6},{7,8,9} }; // → 오류  
        for(int x=0; x < a.length ; x++) {  
            for(int y=0; y < a[x].length ; y++)  
                System.out.print(a[x][y]+" ");  
            System.out.println();  
        }  
    }  
}
```

1	2	3
4	5	6
7	8	9

❖ 실습예제4

ArrayEx4.java

1. 3*4의 int형 2차원 배열을 만들자.
2. 각 원소에 0~9 범위의 정수를 랜덤하게 저장하자.
3. 2차원 배열 원소와 합계를 오른쪽 예시와 같이 출력하자.

(예시)

0	8	6	1
1	9	4	4
8	6	2	6
합계:	55		

(참고) ** 랜덤숫자 얻기 **

```
import java.util.Random; // Random 클래스는 임포트해야 쓸 수 있음.
```

```
Random rd = new Random( ); // 랜덤숫자를 얻기 위해 Random객체를 생성.
```

```
int n = rd.nextInt(10); // 0~9까지 범위의 정수 값을 랜덤으로 얻어 n에 저장.
```

❖ 실습예제4 도움말

ArrayEx4.java

```
import java.util.Random;
```

```
public class ArrayEx4 {
```

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

```
        int arr[][]; //2차원 int형 배열 선언
```

```
        arr = _____ ; // arr에 3*4 정수배열 생성
```

```
        // 12개 정수를 랜덤하게 발생하여 배열 원소에 순서대로 저장
```

```
        Random rd = new Random();
```

```
        for(int x=0; x<arr.length; x++)
```

```
            for(int y=0; y<arr[x].length; y++)
```

```
                arr[x][y] = rd.nextInt(10);
```

```
        // 이하 코드생략 . . . . .
```

```
    }
```

```
}
```

(예시)

0	8	6	1
1	9	4	4
8	6	2	6

합계: 55

실습예제4

```
1 import java.util.Random;
2 public class ArrayEx4 {
3     public static void main(String[] args) {
4         int arr[][];
5         arr = new int[3][4];
6         Random rd = new Random();
7         for(int x=0; x<arr.length; x++)
8             for(int y=0; y<arr[x].length; y++)
9                 arr[x][y] = rd.nextInt(10);
10
11         int sum = 0;
12         for(int x=0; x<arr.length; x++) {
13             for(int y=0; y<arr[x].length; y++) {
14                 System.out.print(arr[x][y]+"\\t");
15                 sum += arr[x][y];
16             }
17             System.out.println();
18         }
19         System.out.println("합계: "+sum);
20     }
21 }
```

10. 배열

❖ 이차원 배열

```
int i[ ][ ] = new int[4][ ];
```

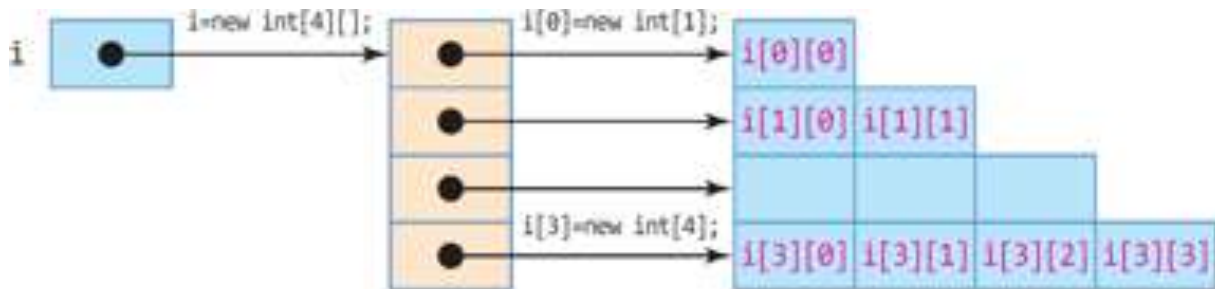
비정방형 배열

```
i[0] = new int[1];
```

```
i[1] = new int[2];
```

```
i[2] = new int[3];
```

```
i[3] = new int[4];
```



```
i.length : 4
```

```
i[0].length : 1
```

```
i[1].length : 2
```

```
i[2].length : 3
```

```
i[3].length : 4
```

(참고)

```
int i[][] = { {0}, {1,2}, {3,4,5}, {6,7,8,9} };
```

❖ 실습예제5

ArrayEx5.java

비정방형 배열을 다음과 같이 만들고 초기화하자.
오른쪽 예시와 같이 출력하는 프로그램을 작성하자.

	0	1	2
0	10	11	12
1	20	21	
2	30	31	32
3	40	41	

(예시)

```
10 11 12
20 21
30 31 32
40 41
```

(참고) 배열의 크기(길이) : length 필드(상수값)

실습예제5

```
1 public class ArrayEx5 {  
2     public static void main(String[] args) {  
3         int arr[][] = new int[4][];  
4         arr[0] = new int[3];  
5         arr[1] = new int[2];  
6         arr[2] = new int[3];  
7         arr[3] = new int[2];  
8         for(int x=0; x<arr.length; x++)  
9             for(int y=0; y<arr[x].length; y++)  
10                arr[x][y] = (x+1)*10 + y;  
11  
12         for(int x=0; x<arr.length; x++) {  
13             for(int y=0; y<arr[x].length; y++)  
14                 System.out.print(arr[x][y]+" ");  
15             System.out.println();  
16         }  
17     }  
18 }
```

10	11	12
20	21	
30	31	32
40	41	

❖ 문자열형 객체 배열 : String 객체 배열

```
public class StringArray {  
    public static void main(String[] args) {  
        String[] str = new String[3]; // 각 배열원소 초기값: null  
        str[0] = "Hello";  
        str[1] = "Java";  
        str[2] = new String("World"); // str[2] = "World";  
        for(String s : str) // for-each문 ★  
            System.out.println(s);  
        // for(int x=0; x<str.length; x++)  
        //     System.out.println(str[x]);  
    }  
}
```

Hello
Java
World

10.배열

❖ 명령행 매개 변수

main메소드는 문자열을 매개변수로 받아 실행시 필요한 정보를 프로그램에 전달.
이러한 매개변수를 **명령행 매개변수**로 함.

main메소드의 매개변수는 String 데이터형을 원소로 하는 배열.(**String객체배열**)

사용자가 명령행에서 입력한 문자열들은 순서에 따라 차례대로 **배열 args**에 저장.

* 콘솔(cmd)에서 실행하는 방법

args 0 1 2

ArguExam.java - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
public class ArguExam {
    public static void main(String[] args) {
        System.out.println("개수: "+args.length);
        for(String s: args)
            System.out.println(s);
    }
}
```

C:\tmp>java ArguExam Hello Java World

개수: 3

Hello

Java

World

명령행 매개변수

❖ 명령행 매개 변수 * eclipse ide에서 실행하는 방법

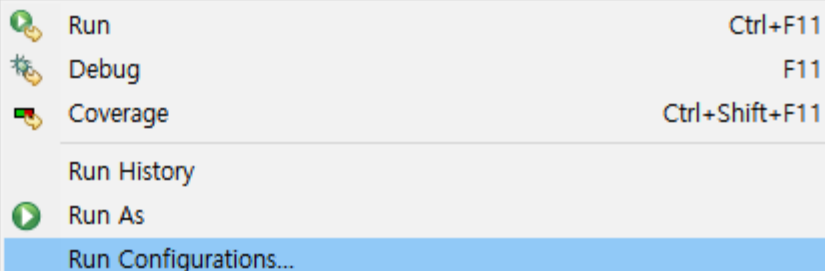
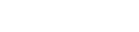
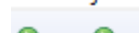
ArguExam.java

```
1 public class ArguExam {  
2     public static void main(String[] args) {  
3         System.out.println("개수: "+args.length);  
4         for(String s: args)  
5             System.out.println(s);  
6     }  
7 }
```

개수: 3
Hello
Java
World

- Eclipse IDE

Project Run



Name: ArguExam

Main (x)= Arguments

Program arguments:

Hello Java World

❖ 실습예제6

ArrayEx6.java

명령행 매개변수를 다음 예시와 같이 출력되도록 프로그램을 작성해보자.

1. 명령행 매개변수가 없을 때

명령행 매개변수를 입력하세요.

2. 명령행 매개변수가 1개일때
(JAVA 입력된 경우임)

입력된 명령행 매개변수는 JAVA입니다.

3. 명령행 매개변수가 2개일때
(JAVA World 입력된 경우임)

입력된 명령행 매개변수는 JAVA, World입니다.

4. 명령행 매개변수가 3개이상일때

명령행 매개변수는 2개까지만 입력 가능합니다.

실습예제6

```
1 public class ArrayEx6 {  
2     public static void main(String[] args) {  
3         if(args.length == 0)  
4             System.out.println("명령행 매개변수를 입력하세요.");  
5         else if(args.length==1)  
6             System.out.println("입력된 명령행 매개변수는 "  
7                 +args[0]+"입니다.");  
8         else if(args.length==2)  
9             System.out.println("입력된 명령행 매개변수는 "  
10                +args[0]+", "+args[1]+"입니다.");  
11         else  
12             System.out.println("명령행 매개변수는 2개까지만 입력"  
13                + "가능합니다.");  
14     }  
15 }
```

{ 2부 }

객체지향과 클래스

{ INDEX }

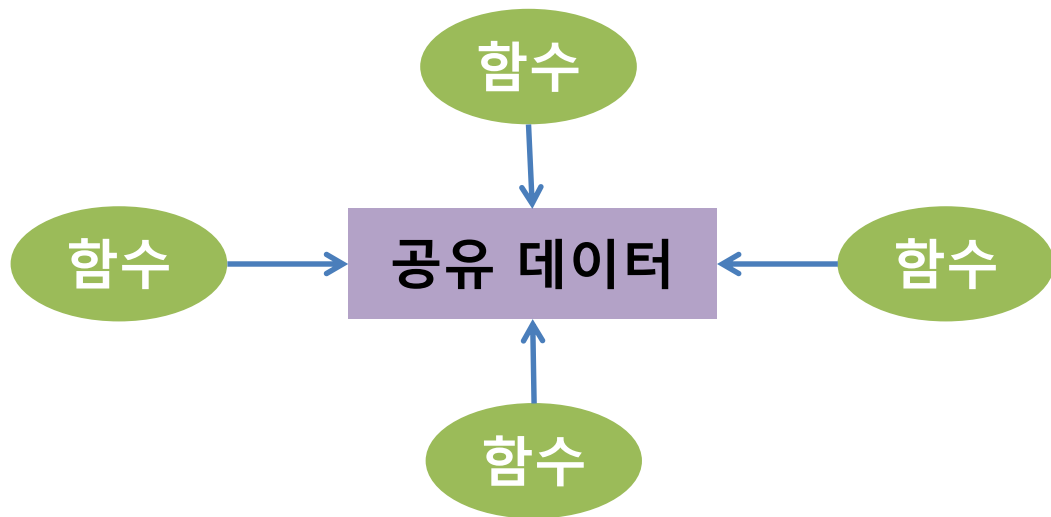
1. 객체지향의 개요
2. 클래스
3. 상속
4. 추상클래스와 인터페이스
5. 패키지의 활용

객체지향의 개요

❖ 객체지향의 개념

절차지향(Procedural-Oriented)

공유된 데이터 + 함수 → 문제해결절차 중시



공유 data의 format이
변한다면?

작성한 함수를
다른 data에 사용하려면?

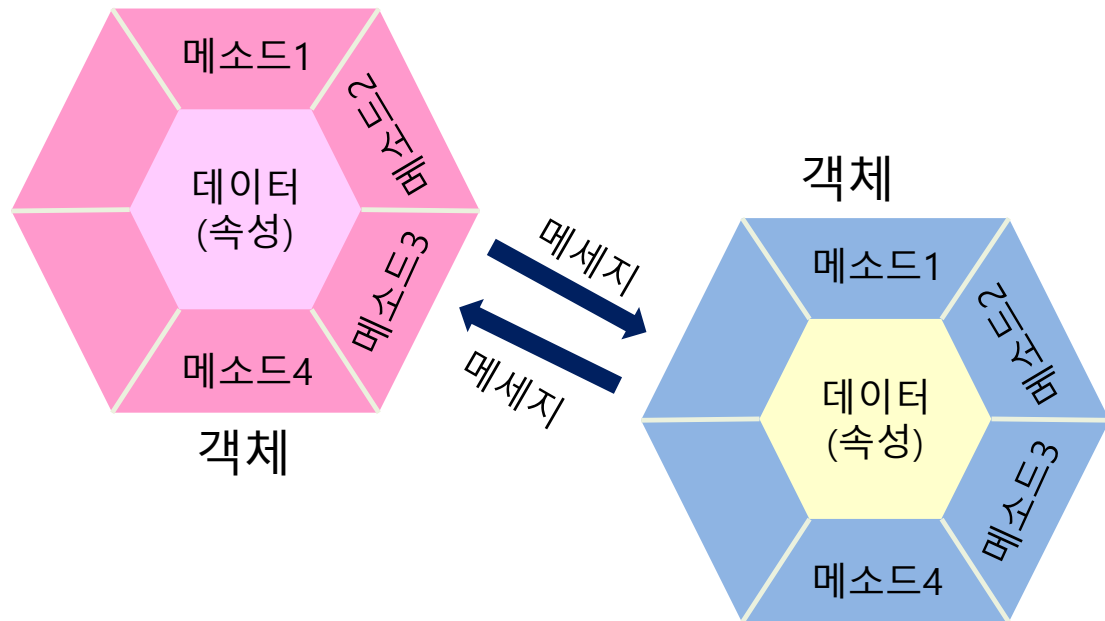
객체지향의 개요

❖ 객체지향의 개념

객체지향(Object-Oriented)

객체를 만들어서 객체들이 **메시지로 통신**함.

객체 : 속성(데이터) + 데이터를 처리할 수 있는 메소드



이 때,
프로그래밍이란?

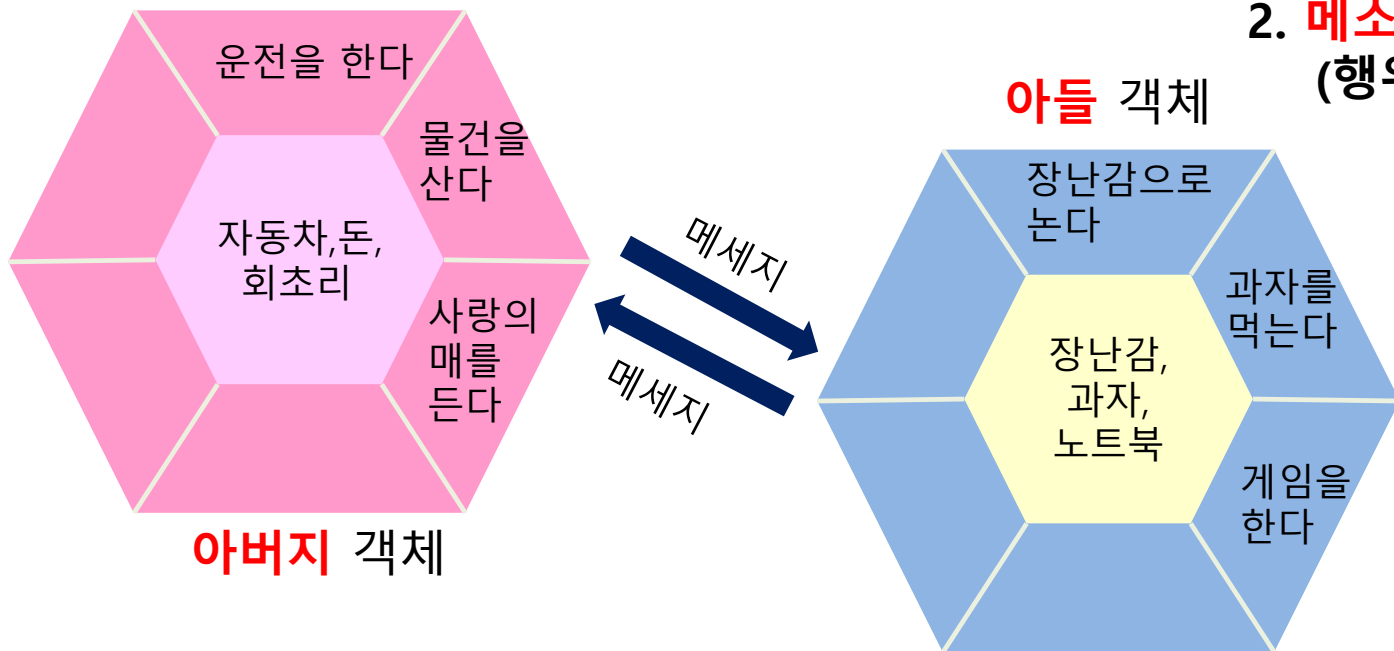
객체들의 집합

객체지향의 개요

❖ 객체지향의 개념

객체(object) : 실세계에 존재하는 모든 것 (명사 성질)
실제 프로그램에 필요한 특성 중심 모델링

- 두가지 관점 : 1. 속 성 : 객체 정적 특성
- 2. 메소드 : 객체 동적 특성 (행위 표현)



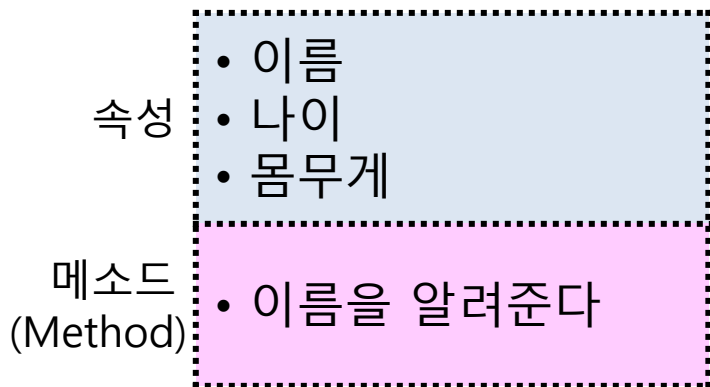
객체지향의 개요

❖ 클래스class → 객체instance

- (같은 특성 갖는 여러 개) 객체는 항상 클래스로부터 생성됨.
즉, 클래스(**붕어빵틀**)는 객체(**붕어빵**)를 만들어 내는 틀(template).
- 속성(field, 정적인 특성)과 메소드(method, 동적인 특성)을 가짐.



Person 클래스

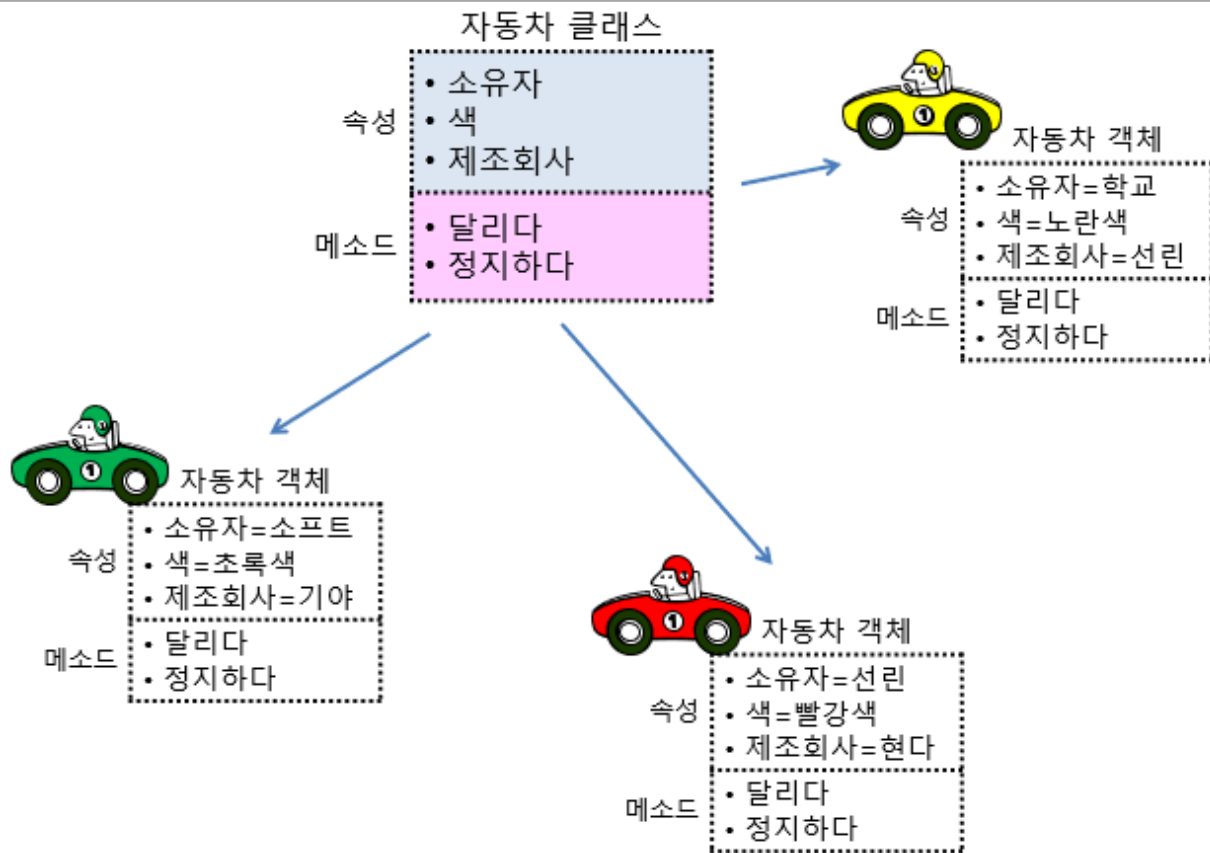


```

Class Person{
    String name;
    int age;
    double weight;
    String getName( ) {
        return name;
    }
}
  
```

객체지향의 개요

❖ 클래스 Class 정의 → 객체(instance) 생성



객체지향의 개요

❖ 객체지향 프로그래밍 3단계

1실세계 객체 모델링 -> 2클래스 정의 -> 3객체 생성 및 사용

메모리에 생성, 사용

클래스
정의

```
class 자동차 {  
    String 소유자 ;  
    String 색상 ;  
    String 제조회사 ;  
    void 달린다() {    }  
    void 정지하다() {    }  
}
```

객체의
생성
및
사용

자동차 노란자동차 = new 자동차() ;

노란자동차.소유자 = "김선린" ;
노란자동차.색상 = "노란색" ;
노란자동차.제조회사 = "(주)선린" ;
노란자동차.달린다() ;
노란자동차.정지하다() ;

❖ 객체지향 언어의 특징

- 캡슐화(Encapsulation)
- 상속(Interitance)
- 다형성(Polymorphism)
- 메시지(message)

❖ 객체지향 언어의 특징

캡슐화(Encapsulation)

- 객체를 만들 때 캡슐화하여 What만 보여주고 How는 감춘다.
- 객체를 작성할 때 숨겨야 하는 정보(private)와 공개해야 하는 정보(public)를 구분하여 작성
- 객체의 사용자는 기능만 알고 사용하며 어떻게 처리되는지는 은폐(Information Hiding)된다.



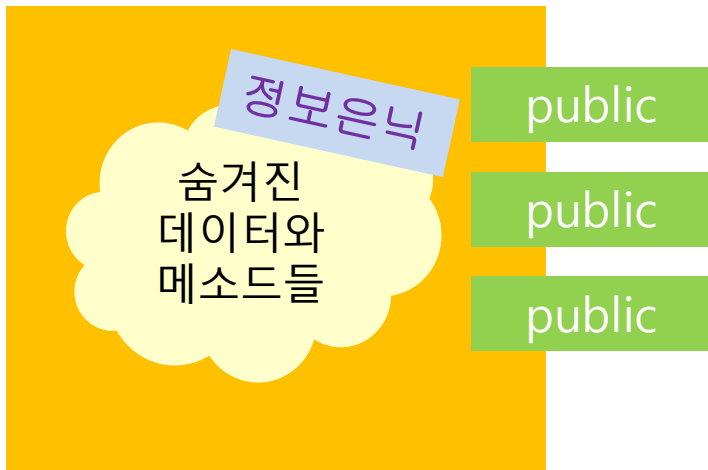
정보은닉

*접근제어자: public private protected (default)

객체지향의 개요

❖ 객체지향 언어의 특징

캡슐화(Encapsulation)



객체의 사용자들은
공개된 인터페이스를
통해서만
객체에 접근한다.

객체지향의 개요

❖ 객체지향 언어의 특징

상속(Inheritance)

Loose Coupling

- 상속 관계의 클래스들은 계층 구조를 구성할 수 있다.
- 하위 계층의 클래스는 상위 계층의 모든 요소를 상속 받고, 추가적으로 필요로 되는 새로운 자료구조와 메소드를 더 가진다.
- 하위 클래스는 상위 클래스를 확장한 개념. `extends`

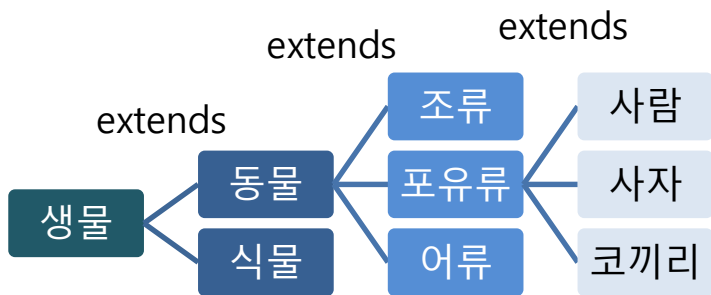
* 상위클래스: 부모클래스, 조상클래스, 슈퍼클래스

* 하위클래스: 자식클래스, 자손클래스, 서브클래스

객체지향의 개요

❖ 객체지향 언어의 특징

상속(Inheritance)



Loose Coupling

생 물 = 호흡함

동 물 = 호흡함 + 걸어다님

포유류 = 호흡함 + 걸어다님 + 젖으로 양육함

사 람 = 호흡함 + 걸어다님 + 젖으로 양육함 + 웃음

*동물: 생물의 속성을 가지고 있음(=상속)

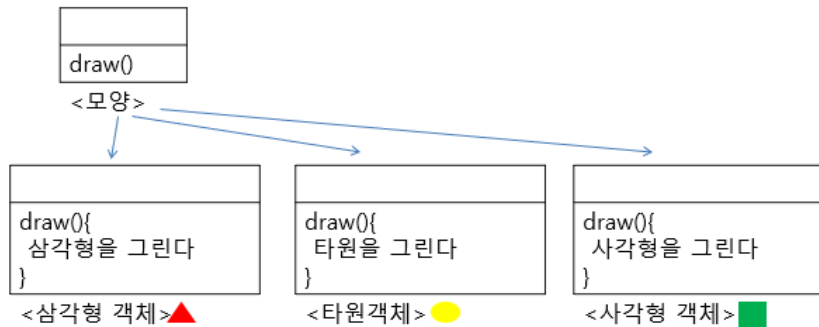
*사람: 생물, 동물, 포유류의 속성을 가지고 있음(=상속)

*동물은 사람의 슈퍼클래스, 사람은 동물의 서브클래스

❖ 객체지향 언어의 특징

다형성(Polymorphism)

- 객체 메소드 형태가 다양한 것.
- 메소드 오버로딩(overloading)
- 메소드 오버라이딩(overriding)



객체지향의 개요

❖ 객체지향 언어의 특징

메시지(message)

- 객체 간에 서로 통신하는 방법
- 객체들은 메시지를 주고 받으면서 일을 수행함

예: 자동차를 타고 가던 사람 객체가 자동차 객체를 멈추기 위해



김선린 객체(송신)



초록자동차 객체(수신)

public

public

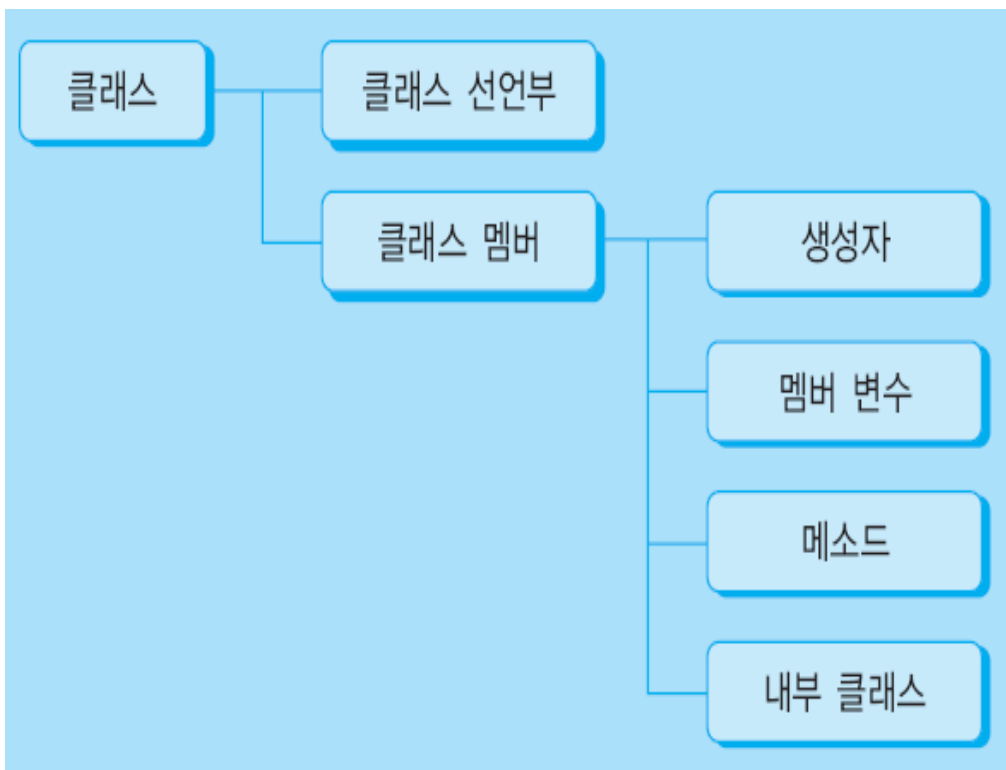
public

전달

{ INDEX }

1. 객체지향의 개요
- 2. 클래스**
3. 상속
4. 추상클래스와 인터페이스
5. 패키지의 활용

❖ 클래스의 구조 = 구성 멤버



클래스의 구조

형식

```
class 클래스 이름    // 클래스 선언부
{
    // 클래스 멤버
}
```

❖ 1 클래스의 정의

- 필드(Field) ————— 객체의 데이터가 저장되는 곳
- 생성자(Constructor) ————— 객체 생성시 초기화 역할 담당
- 메소드(Method) ————— 객체의 동작에 해당하는 실행 블록

```
public class ClassName {  
  
    //필드 = 멤버 변수  
    int fieldName;  
  
    //생성자  
    ClassName() { ... }  
  
    //메소드  
    void methodName() { ... }  
  
}
```

❖ 1 클래스의 정의

- **필드**(=멤버 변수) 선언: 타입 필드 [= 초기값];

- 예

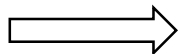

```
String model = "그랜저";
int maxSpeed = 300;
int productionYear;
```

- 필드의 **초기값**: 초기값을 지정하지 않으면, 자동으로 **기본값**으로 초기화됨

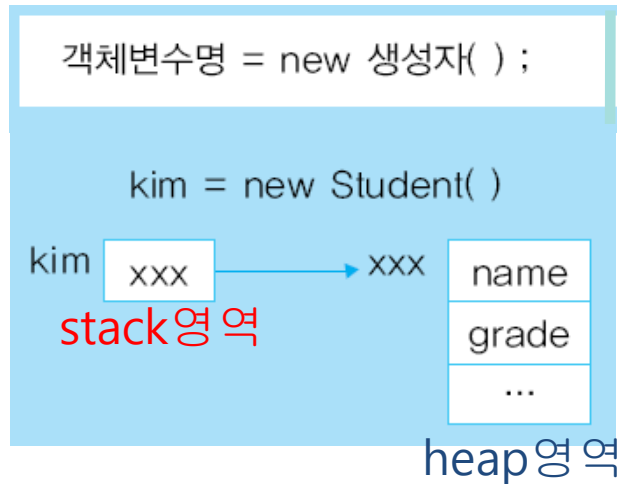
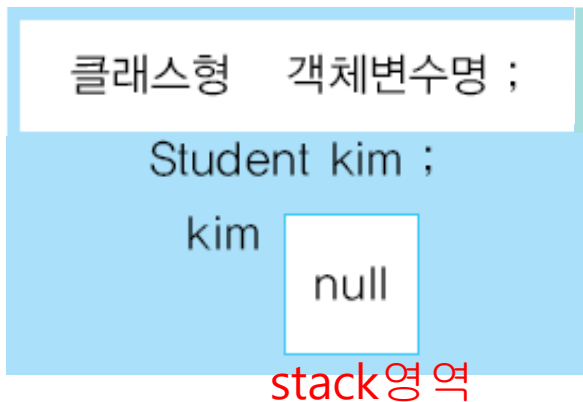
분류		데이터 타입	초기값
기본 타입	정수 타입	byte	0
		char	Wu0000 (빈 공백)
		short	0
		int	0
		long	0L
	실수 타입	float	0.0F
		double	0.0
	논리 타입	boolean	false
참조 타입		배열	null
		클래스(String 포함)	null
		인터페이스	null

❖ 2 객체 생성

* 1단계 : 객체참조형 변수 선언



* 2단계 : 객체 생성



* (참고) 1,2단계를 하나의 문장으로 표현할 수 있다.

클래스형 객체변수명 = new 생성자();

예 Student kim = new Student();

❖ 2 객체 생성

- **new** 연산자
 - 객체를 생성하는 역할

- `new 클래스();`
- 클래스()는 생성자를 호출하는 코드
- 생성된 객체는 힙 메모리 영역에 생성



- 객체를 생성한 후, **객체의 주소를 리턴**

❖ 3 객체 사용 : 객체의 멤버 접근

- 생성된 객체의 속성이나 메소드에 접근하려면 .(dot) 연산자를 이용한다.
 - 예) 객체 참조 변수 **kim**의 멤버 변수인 name은 **kim.name**을 통해 접근 가능하다
- 사용위치에 따른 접근
 - **객체 외부** : 객체 생성 후 객체를 사용할 때... (보통)
 - "**참조변수.필드(또는 메소드)이름**" 으로 접근
 - **객체 내부** : 클래스 내에서 사용할 때...
 - "**필드(또는 메소드)이름**" 으로 바로 접근
 - "**this.필드(또는 메소드)이름**" 으로 접근

this

- 객체(인스턴스) 자신의 참조(번지)를 가지고 있는 키워드
- 객체 내부에서 인스턴스 멤버임을 명확히 하기 위해 this. 사용

```
void setModel(String model) {  
    this.model = model;  
}
```

- super는 부모 객체 참조(참고: this는 자신 객체 참조)

❖ 3 객체 사용 : 객체의 멤버 접근

예제 II-1 ClassExam.java

```
1 public class ClassExam{
2     public static void main( String args[ ] ) {
3         Student kim = new Student( ) ;           //㉠
4         kim.name = "김현우" ;                     //㉡
5         kim.grade = 2 ;
6         System.out.println("학생의 이름은 " + kim.name + "이고, " + kim.grade + "
7         학년입니다.") ;
8     }
9 }
10 class Student{                                   //㉢
11     String name ;                                // 이름
12     int grade ;                                  // 학년
13     int class ;                                  // 반
14     int number ;                                 // 번호
15     String telephone ;                          // 전화번호
16 }
```

:: 실행 결과

학생의 이름은 김현우이고, 2학년입니다.

:: 설명

㉠ : Student 클래스를 정의한다.

㉡ : Student 클래스형의 객체참조형 변수 kim을 선언 및 생성한다.

㉢ : kim의 멤버 변수 값을 초기화한다.

❖ **접근 제어자** : 클래스 필드, 메소드에 대한 접근권한 제어 \Longrightarrow 캡슐화, 정보은닉

접근 제어자	사용 범위				설명
	클래스	하위 클래스	동일 패키지	모든 클래스	
★ private	○	×	×	×	<ul style="list-style-type: none"> 자신 클래스 안에서만 사용할 수 있는 멤버 선언
(default)	○	×	○	×	<ul style="list-style-type: none"> 접근 권한 modifier를 지정하지 않은 경우 동일 패키지의 클래스에서만 접근 가능
protected	○	○	○	×	<ul style="list-style-type: none"> 동일 패키지의 클래스 또는 동일 패키지는 아니지만 하위 클래스 관계인 클래스에서만 접근 가능
public	○	○	○	○	<ul style="list-style-type: none"> 모든 클래스에서 접근 가능

❖ 접근 제어자 : 클래스 필드, 메소드에 대한 접근권한 제어 \Longrightarrow 캡슐화, 정보은닉

예제 11-2 MemberExam.java

```
1 public class MemberExam{
2     public static void main( String args[ ] ) {
3         Student kim = new Student( ) ;
4         kim.name = "김현우" ;
5         kim.grade = 2 ;
6         kim.telephone = "111-1111" ;
7         System.out.println("학생의 이름은 " + kim.name + "이고, " + "학년은 " +
8         kim.grade + "이고, 전화번호는 " + kim.telephone + "입니다.") ;
9     }
10 }
11 class Student{
12     public String name ;           // 모든 클래스에서 접근 가능한 멤버 변수 name
13     int grade ;                   // 동일 패키지에서만 접근 가능한 멤버 변수 grade
14     int class ;                   // 동일 패키지에서만 접근 가능한 멤버 변수 class
15     int number ;                 // 동일 패키지에서만 접근 가능한 멤버 변수 number
16     private String telephone ; // 자신의 클래스 안에서만 접근 가능한 멤버 변수 telephone
17 }
```

:: 실행 결과

```
MemberExam.java:6: telephone has private access in Student
    kim.telephone = "111-1111" ;
    ^
```

```
MemberExam.java:8: telephone has private access in Student
    kim.grade + "이고, 전화번호는" + kim.telephone + "입니다." ;
    ^
```

2 errors

:: 설명

멤버 변수 telephone은 private 접근 권한을 가지므로 자신의 클래스인 Student에서만 사용가능하다. 따라서, MemberExam 클래스에서 사용하면 접근 권한 에러가 발생한다.



접근 권한 에러

❖ **static** : 정적 멤버

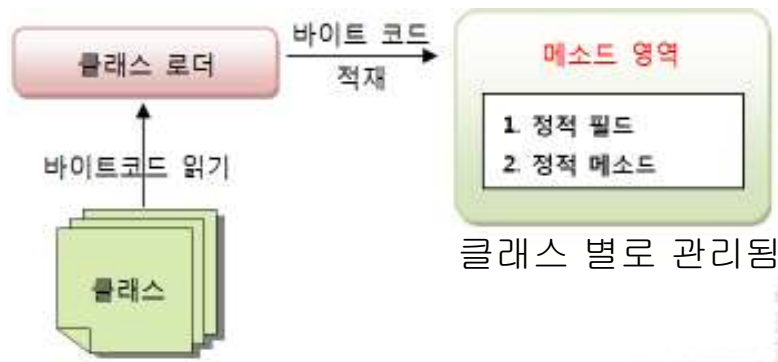
클래스에 고정된 멤버. 객체를 생성하지 않고 사용할 수 있는 필드, 메소드

■ 정적 멤버 선언

- 필드 또는 메소드 선언시, **static** 키워드를 붙임

```
public class 클래스 {
    //정적 필드
    static 타입 필드 [= 초기값];

    //정적 메소드
    static 리턴타입 메소드( 매개변수선언, ... ) { ... }
}
```



❖ static : 정적 멤버

클래스에 고정된 멤버. 객체를 생성하지 않고 사용할 수 있는 필드, 메소드

■ 정적 멤버 **사용**

- 클래스 이름과 함께 **도트 연산자**로 접근한다.

- 예)

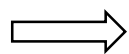
```
public class Calculator {  
    static double pi = 3.14159;  
    static int plus(int x, int y) { ... }  
    static int minus(int x, int y) { ... }  
}
```

[바람직한 사용]

```
double result1 = 10 * 10 * Calculator.pi;  
int result2 = Calculator.plus(10, 5);  
int result3 = Calculator.minus(10, 5);
```

클래스.필드;

클래스.메소드(매개값, ...);



예) **Calculator.pi**;

[바람직하지 못한 사용]

```
Calculator myCalcu = new Calculator();  
double result1 = 10 * 10 * myCalcu.pi;  
int result2 = myCalcu.plus(10, 5);  
int result3 = myCalcu.minus(10, 5);
```

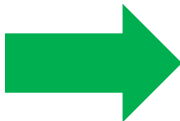
❖ 실습예제1



자동차 모델명

자동차 판매가격

정보를 출력하다()



Car

String modelName

Int price

void printInfo() {...}

CarTest

main 메소드 {

- Car 객체 생성
- 모델 이름 입력: 초기화
- 판매가격 입력: 초기화
- 객체정보 표시: 호출

}

[출력 예시]

Problems | @ Javadoc | Declaration | Console

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

모델명은 선린카이며, 가격은1000만원입니다.

❖ 실습예제1 도움코드

```
1 class Car{
2     // 필드
3     String modelName;
4     int price;
5     // 메소드
6     void printInfo() {
7         System.out.print("모델명은 "+modelName);
8         System.out.println("이며, 가격은"+price+"만원입니다.");
9     }
10 }
11 public class CarTest {
12     public static void main(String[] args) {
13         Car sunrin = new Car();
14         sunrin.modelName = "선린카";
15         sunrin.price = 1_000;
16         sunrin.printInfo();
17     }
18 }
```

Problems @ Javadoc Declaration Console

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

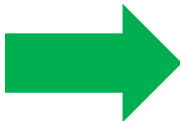
모델명은 선린카이며, 가격은1000만원입니다.

❖ 실습예제2



현재 종이 매수

출력하다(출력매수)



Printer

int numOfPapers

void print(int amount) {...}

[출력예시]

70장을 출력하였습니다.

남은 종이 매수: 30장

PrinterTest

main 메소드 {

- Printer 객체 생성
- 용지 100장 넣기: 초기화
- 70장 프린트하기: 호출
- 남은 매수 표시: 화면출력

}

❖ 실습예제2 도움코드

```
class Printer {  
    int numOfPapers;  
    void print(int amount) {  
        int usage;  
        if(amount<=numOfPapers) usage = amount;  
        else usage = numOfPapers;  
        System.out.println(usage+"장을 출력하였습니다.");  
    }  
}  
  
public class PrinterTest {  
    public static void main(String[] args) {  
        Printer p = new Printer( );  
        p.numOfPapers = 100;  
        p.print(70);  
        System.out.println("남은 종이 매수"  
                           +p.numOfPapers+"장");  
    }  
}
```

Printer

int numOfPapers**void** print(int amount) {...}

PrinterTest

main 메소드 {

- Printer 객체 생성
- 용지 100장 넣기: 초기화
- 70장 프린트하기: 호출
- 남은 매수 표시: 화면출력

}

❖ 실습예제3

Util

```
static final int thisYear = 2018;
```

```
static int getAge(int year) {...}
```

Animal

```
private String group = "동물"
```

```
void setGroup(String group) {...}
```

```
String getGroup() {...}
```

AnimalTest

```
main 메소드 {
```

- Animal 객체 생성
- 종류 입력: setGroup() 호출
- 종류 출력: getGroup("사람") 호출
- 나이 출력: Util.getAge() 이용

```
}
```

[출력 예시]

Console Problems @ Java

<terminated> AnimalTest (1) [Java App

sunrin은 사람

sunrin의 나이: 만**16**세

❖ 실습예제3 도움코드

```
1 public class Util {
2     public static final int thisYear = 2018;
3     public static int getAge(int year) {
4         if(year > thisYear)
5             return 0;
6         return thisYear - year;
7     }
8 }
```

```
1 public class Animal {
2     private String group = "동물";
3     public void setGroup(String group) {
4         this.group = group;
5     }
6     public String getGroup() {
7         return group; // return this.group;
8     }
9 }
```

```
1 public class AnimalTest {
2     public static void main(String[] args) {
3         Animal sunrin = new Animal();
4         // sunrin.group = "사람"; // 접근권한 에러 ★
5         sunrin.setGroup("사람");
6         System.out.println("sunrin은 "+sunrin.getGroup());
7         System.out.println("sunrin의 나이: 만"
8             + Util.getAge(2002)+"세");
9     }
10 }
```

Console Problems @ Jav.

<terminated> AnimalTest (1) [Java App]

sunrin은 사람

sunrin의 나이: 만16세

중간고사 화이팅!

