배열 → 변수의 연장 | 배열은 무조건 참조형이다.

동일한 유형의 동일한 성격의 데이터가 여러 개 선언되어 주어야 될 때 사용한다.

① 배열의 선언

int[ ] myarr;

타입[ ] 변수이름.

int 변수이름 → 기본형 | int [] 변수이름 → 참조형

② 배열의 생성

배열이 heap에 저장되기 때문에 new를 사용하여 배열을 생성한다.

변수이름 = new 타입[길이];

myarr = new int[5];

③ 배열의 초기화

myarr[4] = 100;

String은 참조형이다

인덱스 개수: myarr.length -1

초기값을 주지 않았을 때 모든 reference(참조형) 타입은 다 null로 출력된다.

그러나 그 배열에 동일한 타입의 객체를 할당하면 그 할당된 객체의 주소값이 각 배열에 들어간다.

자바는 인터프리터 언어라서 쭉 한 줄 한 줄씩 해석하다가 오류나면 오류 출력하고 멈춘다.

배열은 한 번 생성하면 배열의 길이를 바꿀 수 없다.

java.lang.system

배열이 어렵다고 느끼는 것은 연습부족이다.

public static void main(String[] args)

(String[] args ) 괄호 안에 있는 것이 명령행 매개변수이다.

콘솔로 실행할 때 명령어를 입력할 수 있는데 이 명령어들을 저장하기 위한 변수이다.

명령어들을 객체로? 변환할 수 있게 기능을 제공해준다.

자바 OOP(object-oriented programming)

자바에서는 (기본형데이터, 오류) → 객체로 관리

기본형 변수 데이터를 String 타입으로 변환해 주거나

string[ ] 안 매개변수를 기본형 타입으로 변환해 준다.

java.lang.Integer class

static int parseInt(String s)

static float parseFloat(String s)

**public** **class** ArgsTest {

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

/\*

\* args = new String[2]

\* ------

\* 명령행 매개변수의 갯수만큼 설정

\* args[0] = "100"

\* args[1] = "200"

\*/

//명령행 매개변수 2개를 입력 받겠다는 의미이다.

System.***out***.println("명령행매개변수->"+args[0]);

System.***out***.println("명령행매개변수->"+args[1]);

**for**(**int** i=0;i<args.length;i++) {

System.***out***.println(args[i]);

}

//String으로 입력받은 명령행 매개변수를 int로 변환

**int** num1 = Integer.*parseInt*(args[0]);// int =String x

**int** num2 = Integer.*parseInt*(args[1]);

System.***out***.println("합=>"+(num1+num2));

}

}

javac ArgsTest.java

명령어 프롬프트에서 정식으로 커맨드 라인 사용하는 방법

java ArgsTest 100 200

Eclipse에서 명령어 프롬프트를 실행하는 방법

run->run configurations->java application 더블 클릭->ArgsTest 클릭 ->Arguments 탭->program arguments의 Variables클릭->string\_prompt클릭->ok->run클릭

입력받은 모양이 string이라서 합쳐지지 않고 문자열처럼 그대로 출력된다.

다차원 배열은 배열을 참조하는 배열이다.

④