**05/20 금**

// 변수를 사용할 때에는 항상 초기화를 시켜주어야 한다.

// 그렇지 않으면 컴파일 에러가 발생한다.

**!** → Not 연산자

형변환 → 자동으로 다른 타입 형으로 변환되는 것.

자료형의 확장. → 작은 타입이 큰 타입에 들어가게 되는 것.(int 값이 → long 값 박스로)

long l = 10000; //int 형이 long 타입 박스에 들어가는것 => 자료형의 확장

//------------ 항상 '축소형'은 손실의 가능성을 확인 하고 사용---------------

// int i = 1000L; => 컴파일러에러 ; 자료형의 축소 ; 값의 손실 가능성이 생김

// int i = (int) 1000L; => 에러 없음. 가능한 값이기 떄문

// int i = (int) 3214545461L; => 출력값이 바뀜 문제발생 손실을 감수하고 결과를 도출해낸 것이기 때문

참조형 타입 : 문자열(String) → 파일 TestString 참조

String 은 앞이 항상 대문자 → 클래스의 이름을 불러오는 것이기 때문.

String 의 메소드 중 하나charAt( ); → 해당하는 index값을 정수 형태로 넣어주면 그 위치의 문자 하나를 알려준다.

String hello = "Hello";

char h = hello.charAt(0);

System.out.println(h);

숫자를 문자열로 바꿔주는 방법

String strNum = String.valueOf(12345); → Static(스태틱) 메소드

String strNum = “12345”; → 쌍따옴표를 사용해서 처음부터 문자열 표현.

String strNum2 = "" + 12345; → 숫자값을 문자열 형태로 바꿈.

// length 메소드를 호출해서, 문자열이 몇 글자로 이루어져 있는지 int 형태로 결과값을 출력함;

int length = hello.length();

( ) → 입력 파라미터

**StringMethods 참조;**

String hello = "Hello";

// index 값을 찾아서 출력해줘;

int index = hello.indexOf("o");

System.out.println(index); // 4 출력

* Char는 한 문자만 입력 받는데 입력 받을 때는 홑따옴표를 쓴다. 잊지말기

// 문자열의 동등함 비교 ----StringEquals.java----

★주의★ → == 과 equals()차이점

== → 참조 하는 대상이 같으면 참

Equals() → 값이 완전히 같으면 참 (문자열 동등함 비교)

public class StringEquals {

public static void main(String[] args) {

String a = "Hello";

String b = "Hello";

String llo = "llo";

// String 은 참조형이기 때문에 값의 확인이 아닌

// 참조의 대상이 같은지 확인한다;

System.out.println(a == b);

// 참조의 대상이 다르기 때문에 false 출력

System.out.println(a == ("He" + llo));

}

}

**// 문자열 입력 받기 → InputString 참조**

//중괄호 → BlockScope.java

{ } 중괄호의 안 → Block이라고 표현.

범위를 열고 닫는다고 표현; 그 범위는 Scope;

Block scope;

// 제어문, 흐름, if() → TestIf.java

if문의 첫 사용

// 조건식은 boolean형 타입 값 true / false ;

if ( 조건식(논리값) ) { }

//

if ( 참 ) {

// 문장1

// 문장2

}

// 문장 3

참이면 → 문장 1, 2, 3 전부 출력

if ( 거짓 ) {

// 문장1

// 문장2

}

// 문장3

거짓이면 → 문장 3번만 출력

//If조건문 사용 → Umbrella, EvenNum, PassFail, DivLeft, BigOne, Size, Grade, Calc .java