<자바 정리>

Data type

기본타입 – 정수타입, 실수 타입, 논리 타입

참조타입 – 배열타입, 열거 타입, 클래스, 인터페이스

참조 타입이란 객체의 번지를 참조하는 타입이다.

참조 타입으로 선언된 변수는 객체가 생성된 메모리 번지를 저장한다.

변수는 모두 stack이라는 영역에 생성된다.

기본 타입의 변수는 직접 값을 가지고 있지만 참조 타입의 변수는 heap 메모리의 객체 번지를 가지고 있고 이 번지를 통해 참조한다.

<JVM이 java 명령어로 구동될 때 운영체계에서 할당 받은 메모리 영역>

메모리 영역(Runtime Data Area) – 힙(heap)영역, 스레드-1(스택(stack)영역), 스레드-n

메소드 영역(Method Area) – 바이트코드 파일이 저장되는 영역(class, ,등)

Heap 영역 – 객체가 생성되는 영역

Stack 영역 – 메소드를 호출할 때마다 프레임(Frame)이 생성되는 영역이다.

(메소드 호출이 끝내면 프레임은 자동 제거되며 여기서 기본타입과 참조타입의 변수가 생성되고 제거된다.)

<참조 타입>

String – 문자열 (문자열 타입)

참조 타입의 ==, != 는 동일한 객체를 참조하였는지에 따라 달라진다.

참조 값이 null인 경우 NullPiontException이 발생할 수 있다.

자바에서는 객체를 제거하는 방법을 따로 제공하지 않는다.

객체를 참조하는 모든 참조를 제거하면 자동으로 제거한다.

자바는 리터럴이 동일하면 객체를 공유하도록 설계되어 있다.

하지만 새로운 객체를 만드는 연산자인 new 연산자(객체 생성 연산자)를 사용해서 서로 다른 객체의 번지를 가지게 할 수 있다.

객체 여부 없이 문자열 내부만을 비교하고자 한다면 equals() 메소드를 이용한다.

* 원본 문자열.equals(비교 문자열) –

문자열에서

* 특정 위치의 문자를 얻고 싶다면 charAt()메소드를 사용한다.
* 문자의 개수를 얻고 싶다면 leagth() 메소드를 사용한다.
* 특정 문자열을 다른 문자열로 대체하고 싶다면 replace() 메소드를 사용한다.
* 특정 위치의 문자열을 가져오고 싶다면 substring() 메소드를 사용한다. substring(int begin) or substring( int begin, int end)
* 특정 문자열의 위치를 찾고자 할 때에는 indexOf() 메소드를 사용한다.
* 주어진 문자열이 단순히 포함되어 있는지에 대해서만 조사한다면 contain() 메소드를 사용하여 true 또는 false를 반환한다.
* 문자열이 구분자로 분리되어 있는 경우, 이를 따로 분리해서 얻고 싶다면 spilt() 메소드를 사용한다.

<배열(Array) 타입>