① Java의 기본 개념

특징 : 인터프린트언어interpreter 간단Simple 강력Robust *객체지향성Object Oriented(object oriented program)

- 자바는 클래스 파일이 필요하다_자바의 기본 원리는 하나의 Class를 만들어서 다르게 사용는데에 있음

-Java는 무조건 JVM을 거쳐야 한다_간단한 중앙제어장치에서 시작한 프로그램언어로, 4byte로 모든 것을 분화해서 생각하기 때문에 오차가 큼
초기 자바는 "카다, 끄다"와 같은 단순한 가능에서 시작했으나,"윈도우, 위닉스, 라눅스" 등 어떤 OS에서도 동일하게 적용될 수 있어서 상용화됨

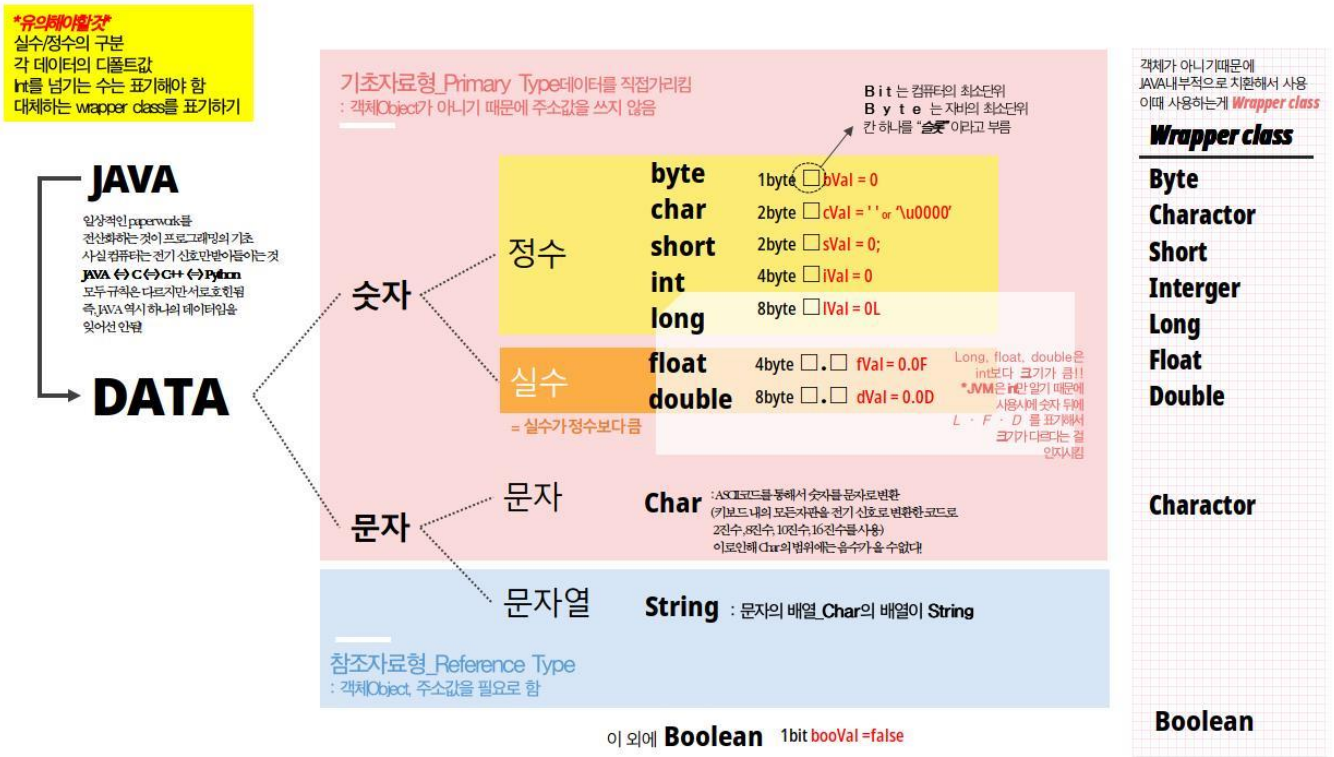
= JDK에 속해있는 *컴파일러Compiler인 javac를 통해서 확장자 java 파일을 바이트 코드byte code인 class코드로 변환하여 CPU에서 구현되도록함
*컴파일Compile 은 기존자원을 이용하여 java소스가 기존 키워드에 맞게 작성되었는지 검사 후 바이트코드로 전환하는 과정을 의미함

구성 : *영역*Field *생성자*Constractor *함수*Method
기본 구성외 + *식별자* 사용자가 식별하기 위해 부여한 이름(class, 변수, 함수의 이름)

클래스Class : Java의 최소단위, 클래스 하위에 함수와 변수가 존재 Class를 활용하는 것이 중요,
Class를 활용하기 위해서 참조변수(메모리에 올리는 과정)가 필요하며
new 연산자를 이용해서 객체object로 만드는 과정을 거침
= 클래스는 단독으로 사용할 수 없으며 상위에 Package가 필요함
- **빌트인 클래스Built-In** : 기존 java프로그램에 존재하고 있는 클래스 (rt.jar 에 있는 자원)
- **사용자 정의 클래스**RefereceType : 사용자가 직접 드는 것
모든 RefereceType은 주소값을 갖고있는데 겉으로는 보이지 않는다(실제론 빌트인에 포함)

② Data Type_자바언어의 데이터타입Data Type은 크게 숫자와 문자로 나뉘며 이는 다시 정수, 실수, 문자, 문자열로 구별할 수 있다.

또한 이를 주소값 사용 여부에 따라서 기초자료형PrimaryType과 참조자료형RefenceType으로 나누어 볼 수 있음



이때 자바클래스 자체가 하나의 데이터이자 resource임을 간과해선 안됨!

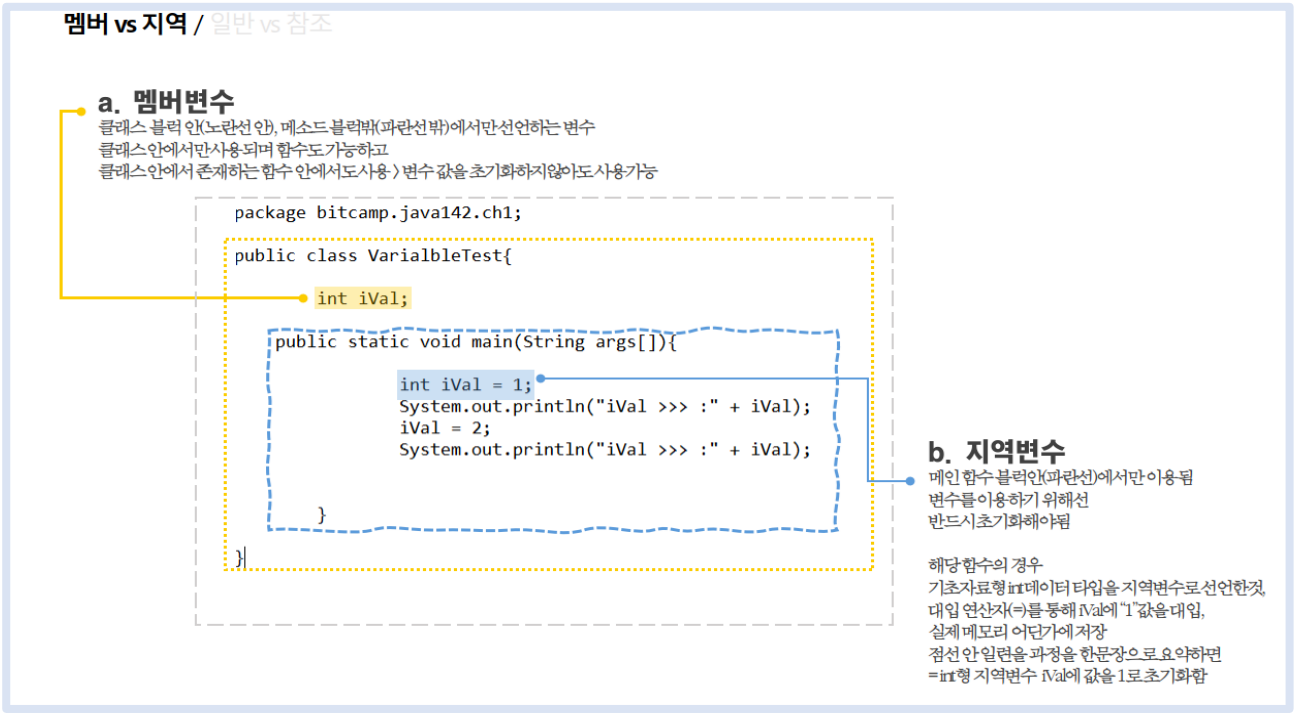
③ 변수Varialble

정의 : 광의의 의미로는 변하는 수, 변화하는 데이터를 저장하는 공간을 의미하며 원론적인 의미로는 데이터를 저장하는 공간을 가리키는 것을 통칭
종류 : 변수의 종류는 많으나 크게 **지역변수**와 **멤버변수**로 나뉘며 이는 다시 **일반변수**와 **참조변수**로 구분지을 수 있다.

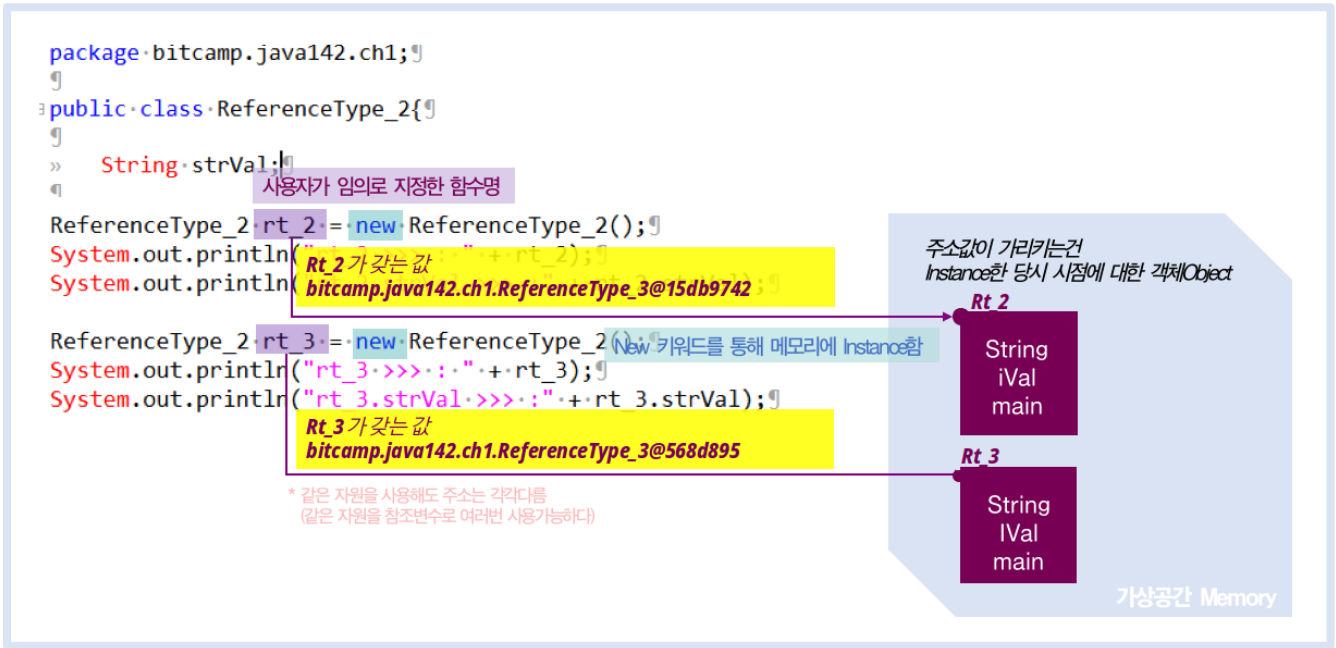
속한 범위	멤버변수	class에는 포함되나 함수 블록안에는 포함되지 않는 변수
	지역변수	함수블록안의 변수, 무조건 값을 초기화default시켜줘야함
주소값	일반변수	주소값 없이 값을 직접 넣는 변수
	참조변수	주소값을 갖고있는 변수, 이때 주소값을 *객체object라고 함

객체object : 유,무형의 행위, 눈에 보이지 않는 것들을 전산화하는 것을 일컬음 Object languge라고 하며, 객체 하위에는 또다른 객체를 가진다.

- 멤버변수와 지역변수는 속한 범위가 어디에 있는가에 따라 나뉨



일반변수 : 기초자료형은 직접 값을 대입하기 때문에 모두 일반변수에 속한다
참조변수 : 실제값이 없는 주소값만을 가지는 변수, "new"를 통해서 주소값을 올려줌(멤버변수, 함수 등)



= 변수는 순차적으로 (멤버변수 / 지역변수)를 먼저 구별한 후 (일반변수 / 참조변수)를 구별해야함

④스트링 클래스String class

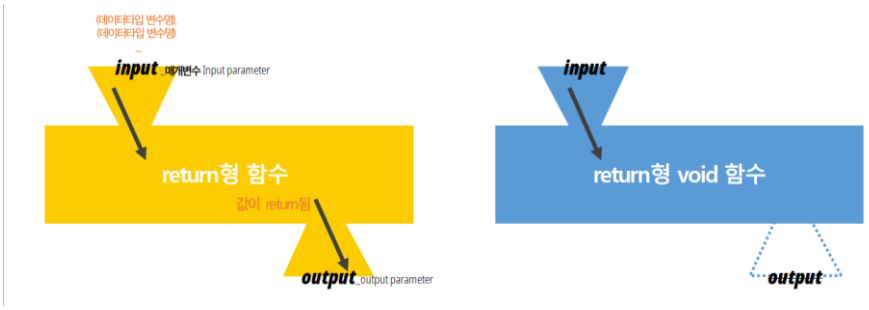
특징 1. 기존 "rt.jar"안에 들어있는 package class를 사용하기 전에 무조건 import를 사용해야함
2. "java lang"안에 들어있는 package들은 import없이 class명만 사용해도 바로 가능!(ex_ srting, system)
3. String은 클래스, 원래 클래스는 무조건 참조변수를 가져야함 ex) a = new A (필드 초기화)
하지만 string의 경우 new선언(new instans) 없이 바로 데이터 선언 가능함!
모든 변수는 초기화*되어있어야함
*초기화initialization : 값을 설정하는 것 , 함수의 상단에는 늘 기초자료 초기화를 설정해야함

Instance 종류

1. static : 생성자가 설정돼있지 않아도 기본 생성자를 설정해줌(기초값),적혀진 통째로 넣어줌
= 메모리를 한번 할당한 후 프로그램이 종료될 때 해제되는 것을 의미함
static키워드를 통해 static영역에 할당된 메모리는 모든 객체가 공유되는 메모리라는 장점이있음
단, *Garbage Collector(가지비 컬렉터_gc)*의 관리 영역 밖에 존재, 프로그램 종료시까지 메모리가 할당된 채로 유지되기 때문에
잦은 사용은 시스템 퍼포먼스에 악영향을 줄 수 있다.
2. new : 필요한 자원을 필요할때마다 Heap영역에 메모리를 올림instance (순차적인게 아니라 필요할때마다 올라가기 때문에 JVM에겐 불편함)
; 기초자료형은 직접 값을 설정하지만 , 참조자료형은 공간을 가짐(주소값, 포인터 라고 부름)
= 이때 필요시마다 new를 통해 주소값을 설정함
즉, 참조변수는 "new" 키워드(리소스)를 통해 자원resource을 메모리 다 올리며,
자원들은 모두 참조변수 안에 있음
=> 참조변수를 사용할 때 new키워드를 사용하면 전체를 올려주고 String에서는 new가 생략
Ex) *String ReferenceType_2 rt_2 = new ReferenceType_2() =>*: 생략가능하다

⑤ 함수Method

함수 규칙 1. 함수는 이름이 있어야하며, 소문자로 시작
2. 함수 만들때는 Return형을 사용해야함
3. 함수이름 뒤에는 파라미터가 들어오는 괄호가 꼭 필요함(단, 데이터는 꼭 선언할 필요 없음
4. 함수 블록(영역Scope)을 만들어야 하며, 함수 블록이 없는 함수도 존재는 해
5. - *Return형이 있는 경우 : Return 키워드 뒤에 오는 데이터타입과 같은 데이터타입을 사용해야함
Return값의 데이터타입은 함수의 Return형 데이터타입과 항상 동일해야함
*Return _ 그 함수에서 어떤일을 하던간에 결과값만 반환해주는것
- Return형이 없는 경우 : Void를 통해서 막아야함



6. 접근제한자(Access Modifier)을 사용할 수 있다.
7. 수정자를 사용할 수 있다.

=자바시그니처(자바가 이 모습을 그대로 기억함)
[접근제한자] [수정자] 리턴형 aName*(데이터타입 변수명(,를 이용해 다수 작성가능)) {
*파라미터(input 파라미터) : 매개변수

}

함수 기본형태

⑦ 생성자Constrator_

자바 클래스는 생성자 없이는 일 할수 없다, 필드값(데이터) 초기화에 필수적이기 때문에
자바의 최소 단위는 클래스, 클래스(자원)를 이용하기 위해선 참조변수, 즉 필요한 데이터를 메모리에 올리는 과정이 필요함
= JVM이 객체Object생성(instance)시에 기본생성자를 만들어 호출해 선언된 자원 중 맴버필드 변수를 **디폴트값**Default Value으로 초기화함
이때 규칙으론 new연산자를 이용해 객체를 만드는 과정을 거치나, 매개변수없는 생성자를 무조건 만들어야함
*생성과정에서 this 키워드는 꼭 사용하지만, this() 함수는 꼭 필요한 것은 아님을 명심할 것

⑥ 접근제한자Access Modifier

종류 1. Public : Package를 넘나들며 사용가능, 메인함수는 무조건 Public을 사용해야함
2. Defualt : 기본 접근 제한자, Package 내에서 사용가능 (생략된 경우 전부 Defualt이며 Package라고도 칭함)
3. Private : 자기 Class내부에서만 사용 가능 〉 Set/Get을 사용하는 경우 Private를 써야함
- Setter/Getter 함수 만드는 법 : ⓐ set/get 접두어 사용 ⓑ 멤버필드 iVal의 첫 번째 이니셜을 대문자로 ⓒ set(get) + iVal = setIval(getIval)
4. Protecte :상속 관계에서만 사용가능
=같은 package를 사용하면 import가 필요없지만 public이라도 다른 package에 있다면 import로 상단에 꼭 불러올 것

은닉화Encapsulation : 캡슐화, 데이터를 보호하는 것으로 아무나 데이터를 사용하지 못하게 하는 것

⑧ 그 외

- 식별자 규칙 : 카멜규칙 지킬 것/ 클래스, 인터페이스는 대문자로 시작 / 변수, 함수는 소문자로 시작 / 모두 언더바(_) 사용가능하다
- 자바 소스 구성 규칙 : 패키지 이름 →import → 클래스 이름 →*필드 → 생성자 → 함수 →main함수(시작점)

*1.상수 2.클래스변수 3.전역변수(public 어디서든 부를수있음) 4.멤버변수 5.은닉화, 변수는 대문자로 쓰며

```
package a.b.c ; //①패키지 이름

import a.b.c.V; //②import
import a.b.c.vo.ValueOfObject;

public class 클래스 이름{ //③클래스 이름
    //④ 필드
    //④ - a. 상수
    public static final String xxxx_xx = "문자
»   public static final int xxxx_xx = 1; //"숫
»   //④ - b 클래스변수
»   public static String XXX_XX = "클래스 변수";
»   //④ - c 전역변수
»   public String sVal;
»   public int iVal;
»   //④ - d 멤버변수
»   int i;
»   String s;
»   //④ - e 은닉화
»   private String sVal;
»   private int iVal;

»   //⑤ 생성자
»   public 클래스 이름(){
»   }
»   public 클래스 이름(String ssVal, int iVal){
»   »   this.ssVal = ssVal;
»   »   this.iival = iiVal;
»
»   }
»
»   //⑥함수
»   public void aMethod(){
»   }
»   public String bMethod(){
»   »   return "";
»   }
»   //⑦main 함수 -> 시작점
»   public static void main(String args[]){
»    }
```