

◎변수의 종류 와 유효범위

매서드안에서 만들어지는 변수 : 지역변수

클래스안에서 만들어지는 변수 : 멤버변수 (인스턴스변수)

클래스안에서 만들어지는 변수 : static 변수 (클래스변수 또는 static 변수)

◆지역변수

매서드 호출될 때 만들어진다

◆인스턴수 멤버변수는

new에 의해서 만들어진다 (각 객체마다 만들어짐)

◆ static 멤버변수 (클래스에 한 개 만들어짐)

프로그램 시작 할 때 먼저 자리를 잡는다. (main매서드 보다 먼저 실행됨)

◎메모리

static 영역 : 사용: 프로그램시작, 반납 :프로그램종료

stack 영역 : 사용(매서드 호출될 때), 반납 (매서드가 종료될 때)

heap영역 : 사용(new에 의해서), 반납 (프로그래머가 free(), delete, java 는 직접하지 않음)

자바는 직접하지 않는다. (참조를 잃은 객체는 가비지의 대상이 된다)

참조를 잃게 되면 가비지 대상(즉시 반납은 아니다, 가비지컬렉터가 일정 텀으로 실행됨)

■ 매서드 안에서 객체를 생성하는 경우 메모리 상태

```
void 객체생성해보기(){
```

```
    Score s = new Score("홍", 10,50);
```

```
    //여기서 생성된 객체는 (heap에 생성된 객체는) 함수반환시 s라는 참조변수가 지역변수 이므로
```

```
    // 반환시 사용하게 될 수 없으므로 참조를 잃어 가비지의 대상이 됨
```

```
}
```

함수안에서 만든 Score 객체는 힙영역에 자리잡음

s변수가 함수반환시 정리되면서 참조를 잃게 되어 가비지 대상이됨 !!

■ 매서드 안에서 객체를 생성하는 경우 메모리 상태 (객체를 반환하는 경우)

```
Score getScoreInstance(){
```

```
    Score s = new Score("홍", 10,50);
```

```
    return s;
```

```
}
```

```
Score result = 객체.getScoreInstance();
```

반환된 객체의 정보를 result가 다시 참조하게 되어 객체정보는 계속 유지 될 수 있음

heap메모리: 메모리를 가장 효율적으로 사용함 (동적메모리라고 함)

가비지컬렉터 : 참조를 잃은 객체의 자원의 반납한다.

변수의 사용과 반납 이해해야 함 !