

\* 클래스 문법

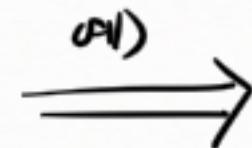
클래스

① 데이터 타입 정의 (User-defined Data Type)  
개별화

② 메서드 분류

\* ↗ 클래스 문법 - 데이터 타입 정의

class 데이터타입 {  
 변수선언  
 :  
}

예) 

class Contact {  
 String name;  
 String email;  
 String tel;  
 String company;  
}

← 메모리  
설정

||  
~~new~~ 명령을 실행하면  
클래스에 정의된 대로  
변수가 준비된다.

\* 클래스를 이용하여 새 데이터 타입의 변수 만들기

Contact c = new Contact();

Contact 클래스의 주소를 저장하는  
리퍼런스(reference)

리터리터링 = 클래스명

클래스 선언에 따라 메모리 주소 => Heap 영역

인스턴스(instance) = 개체(object)

c | 200

200 | name email tel company

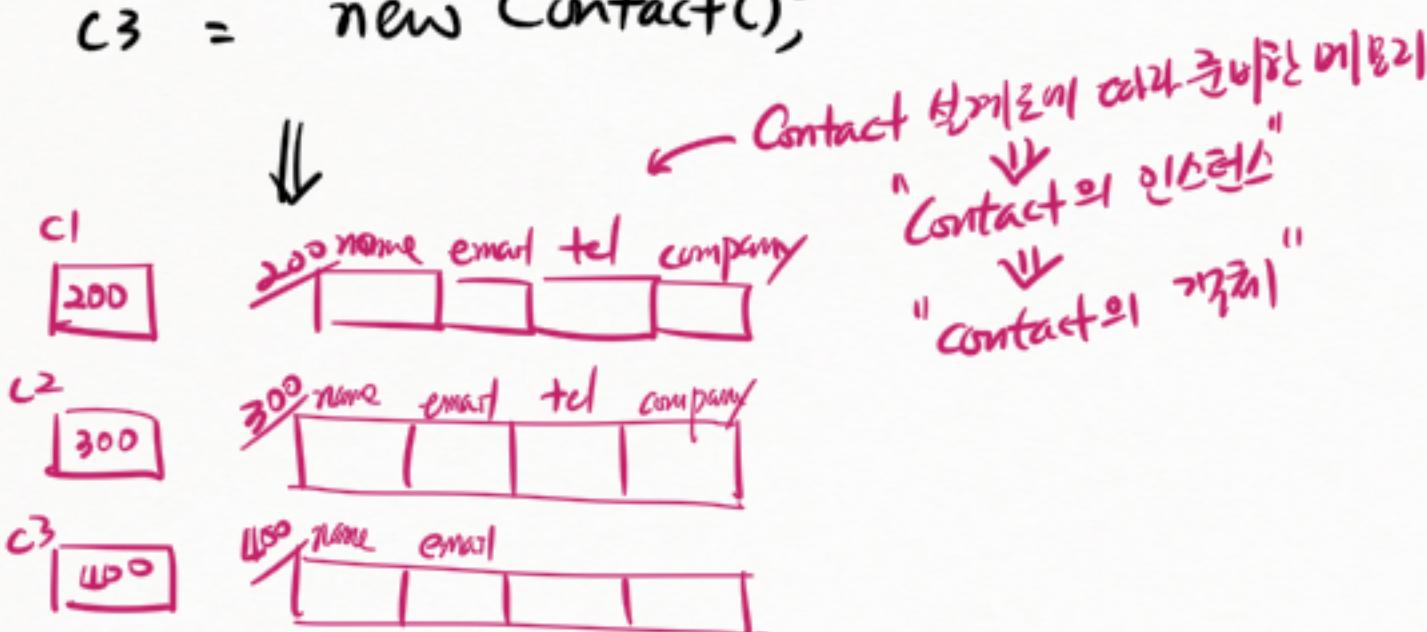
## \* 레퍼런스 배열

## ① 배포 사용 전

```
c1 = new Contact();
```

```
c2 = new Contact();
```

```
c3 = new Contact();
```

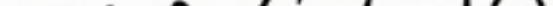


## ② 배포 사용 흐름

Contact[] arr = new Contact[3];  
레퍼런스를 3개 만드는 명령



```
arr[0] = new Contact();
```



A hand-drawn diagram of a Contact object. It consists of a horizontal line with five boxes. The first box is crossed out with a large red X. To its right are four boxes labeled 'name', 'email', 'tel', and 'company' respectively.



```
arr[1] = new Contact();
```



```
arr[2] = new Contact();
```

## \* 레퍼런스와 인스턴스 변수

Contact c = new Contact()



① 인스턴스 변수가 없는 저장

c.name = "홍길동";

인스턴스 주소를  
알고 있는  
레퍼런스

↑  
인스턴스  
변수

c.email = "hong@";

c.tel = "1111";

c.company = "비트";

② 인스턴스 변경

c = new Contact()



c.name = "이꺽정";

c.email = "leem@";

c.tel = "2222";

c.company = "캐논";

기존 인스턴스의  
주소를 알고 있어  
레퍼런스가 한 개로 고정된  
''garbage'' 가 된다.

### Method Area

```
class Score {  
    String name;  
    int kor;  
    int eng;  
    int math;  
    int sum;  
    float aver;  
}
```

### JVM Stack

```
Score s;
```

s 200

↑  
Score의 레퍼런스

### Heap

```
new Score()
```



200 name kor eng math sum aver

↑  
Score의 인스턴스  
(기록체)

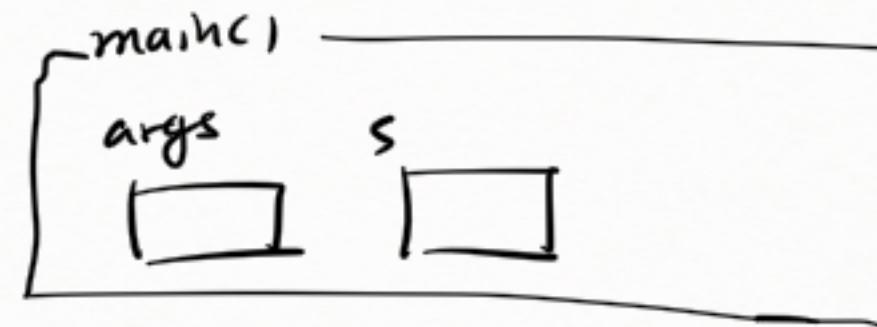
\* com.eomcs.oop.exam. Exam0114

Method Area

```
class Exam0114 {  
    public static void main(String[] args) {  
        System.out.println("Hello Java");  
    }  
}
```

```
class Score {  
    public static void main(String[] args) {  
        System.out.println("Hello Java");  
    }  
}
```

JVM Stack



Heap

Diagram illustrating the Heap:

200	name	kar	eng	math	sum	aver
323	100	100	100			

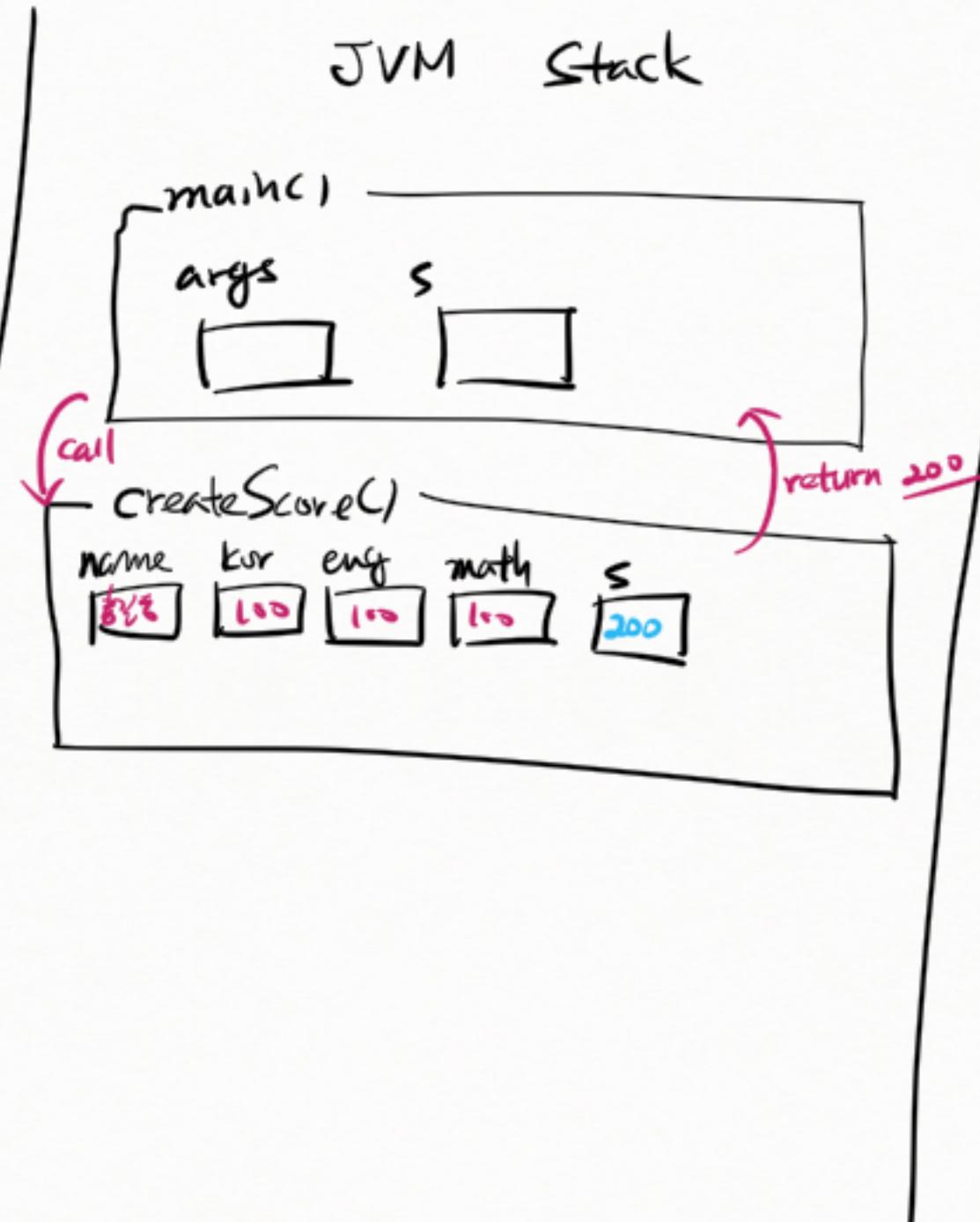
\* com.eomcs.oop.ex01.Exam0114

Method Area

```
class Exam0114 {  
    public static void main(String[] args) {  
        Score s = new Score();  
        s.createScore("B/C", 100, 100, 100);  
        System.out.println(s.getAver());  
    }  
}
```

```
class Score {  
    private String name;  
    private int kur;  
    private int eng;  
    private int math;  
    private int sum;  
    private double aver;  
  
    public void createScore(String name, int kur, int eng, int math) {  
        this.name = name;  
        this.kur = kur;  
        this.eng = eng;  
        this.math = math;  
        this.sum = kur + eng + math;  
        this.aver = sum / 3.0;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public int getKur() {  
        return kur;  
    }  
  
    public int getEng() {  
        return eng;  
    }  
  
    public int getMath() {  
        return math;  
    }  
  
    public int getSum() {  
        return sum;  
    }  
  
    public double getAver() {  
        return aver;  
    }  
}
```

JVM Stack



Heap

name	kur	eng	math	sum	aver
200	100	100	100		

name	kur	eng	math	sum	aver
B/C	100	100	100	300	100

\* com.eomcs.oop.exam. Exam0114

Method Area

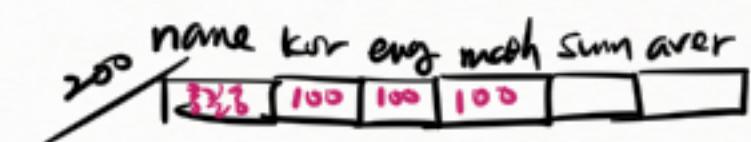
```
class Exam0114 {  
    public static void main(String[] args) {  
        System.out.println("Hello Java");  
    }  
}
```

```
class Score {  
    public static void main(String[] args) {  
        System.out.println("Hello Java");  
    }  
}
```

JVM Stack



Heap



\* com.eomcs.oop.exam. Exam0114

Method Area

```
class Exam0114 {  
    public static void main(String[] args) {  
        Score s = new Score();  
        s.printScore();  
    }  
}
```

```
class Score {  
    public void printScore() {  
        System.out.println("Score");  
    }  
}
```

JVM Stack



Heap

name	kor	eng	math	sum	aver
200	100	100	100	300	100

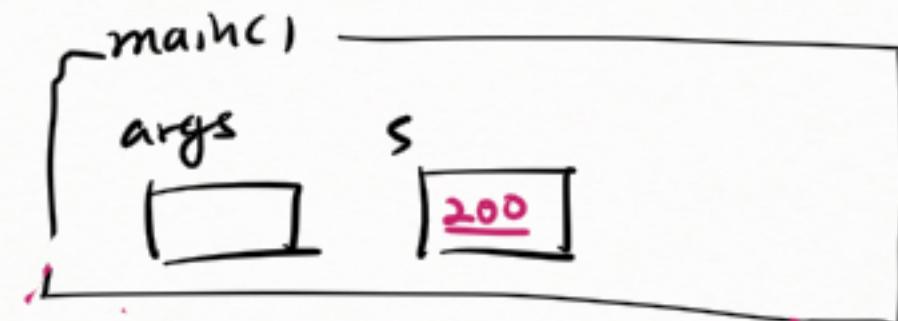
\* com.eomcs.oop.exo1.Exam0114

Method Area

```
class Exam0114 {  
    public static void main(String[] args) {  
        System.out.println("Hello Java");  
    }  
}
```

```
class Score {  
    public static void main(String[] args) {  
        System.out.println("Hello Java");  
    }  
}
```

JVM Stack



Heap

name	kor	eng	math	sum	aver
200	72	100	100	300	100