

---

# 5장. 데이터타입

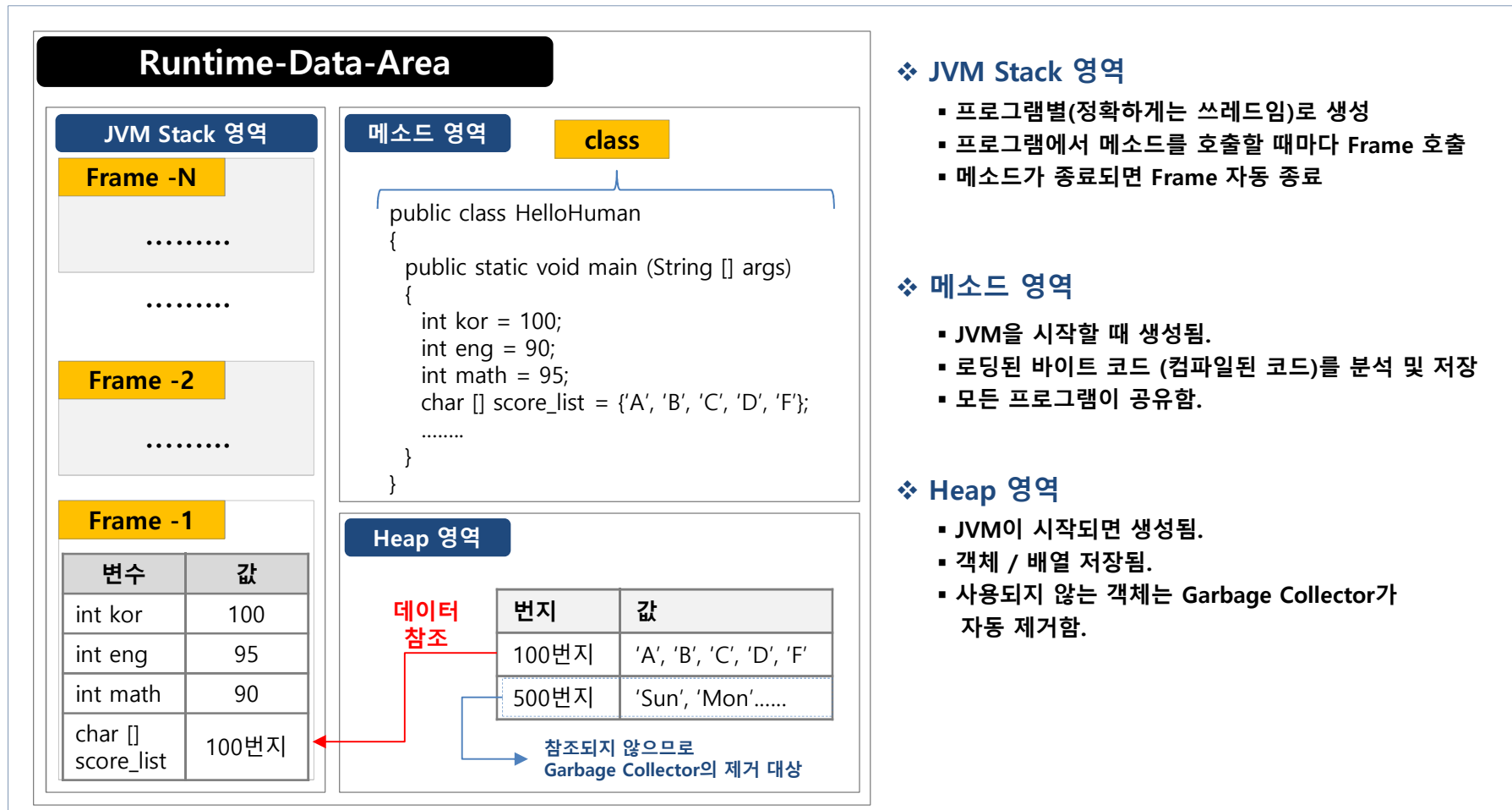
## (참조타입)



## 5.1.2. JVM 메모리 사용영역

- JVM 실행시 운영체제에서 Runtime-Data-Area를 할당 받아 운영됨.
- Runtime-Data-Area는 메소드영역, Heap 영역, JVM Stack 영역으로 구분됨.

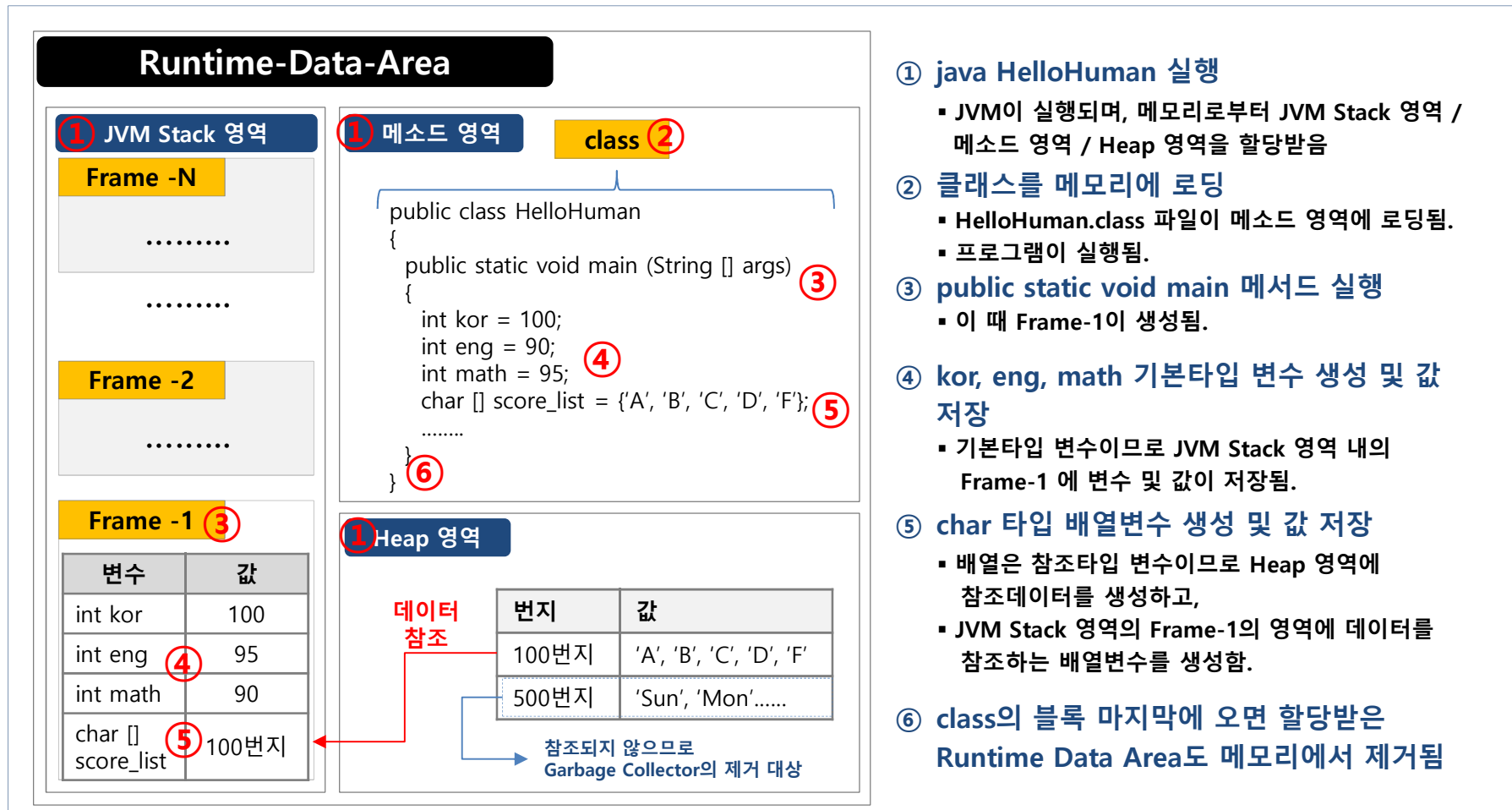
### 메모리 사용영역



### 5.1.3. JVM 메모리 운영방식

- JVM 실행시 운영체제에서 Runtime-Data-Area를 할당 받아 운영됨.
- Runtime-Data-Area는 메소드영역, Heap 영역, JVM Stack 영역으로 구분됨.

#### 메모리 사용영역



## 5.2. 참조타입 데이터 (String)

- String은 문자열을 다루는 클래스임
- 문자 단위를 Heap 영역에 두고 String 변수는 JVM Stack 영역에서 관리함.

### String 클래스 사용 문법

#### 사용 사례

```
String human1 ;
human1 = "휴먼교육센터";
String human2 = "휴먼교육센터";
String human3 = new String ("휴먼교육센터");
```

새로운 객체 생성

#### Stack 영역

변수	번지참조
String human1	110번지
String human2	110번지
String human3	200번지

#### Heap 영역

번지	값
110번지	"휴먼교육센터"

같은 객체참조

번지	값
200번지	"휴먼교육센터"

새로운 객체 참조

#### 주의 사항

```
boolean result1 = (human1 == human2);    // 동일 객체를 참조하고 있기 때문에 true 임.
boolean result2 = (human1 == human3);    // human3은 새로운 객체이므로 false임.

boolean result3 = human1.equals(human3);  // 참조하고 있는 값을 비교하는 것이므로 true 반환함.
```

## 5.3.1 참조타입 데이터 (일차원 배열)

- 배열타입은 여러 개의 문자를 한 개의 변수에 담을 수 있음.
- 한 개의 변수에 index로 구분하여 여러 개의 변수를 사용가능 함.

### 일차원 배열 문법

#### 사용 사례

한 반에 30명의 학생이 있고, 30명의 총점을 구하는 프로그램을 작성할 때 배열을 사용하지 않는다면??

##### 배열 미사용

```
int score1 = 100; .... int score = 80;
int total = 0;
total = score1 + score2 + score3 + ....+ score30;
System.out.println ("학생의 총점 = ", total);
```

코드가 간결해지고  
확장성 및 유지보수  
용이함.

##### 배열 사용

```
int [] score = {100, 90, 80, ..... , 80};
int total = 0;
for (int index = 0 ; index<30; index++) {
    total = total + score[index];
}
System.out.println ("학생의 총점 = ", total);
```

#### 문법

```
int [] human ;           // JVM Stack 영역에 배열변수 생성
human = {100, 200, 300} ;
// Heap 영역에 {100,200,300} 데이터 생성 및
// JVM Stack 영역에 존재하는 human의 배열변수에 번지수 저장
```

#### 주의사항

```
int [] human ;
human = {100, 200, 300} ;   정의한 int형의
                             데이터만 가능함.

human = {100, 200, "휴먼교육센터"} ;
                             String 타입이라
                             컴파일 에러 발생
```

## 5.3.2 참조타입 데이터 (다차원 배열)

- 다차원배열은 배열의 차원이 2개 이상임을 의미함.
- 다차원 배열은 Heap 영역에 1차원 배열의 형태로 메모리를 관리하듯이 처리함.

### 다차원 배열

#### 구조

예시) 2차원 배열구조

행	[0][0]	10	[0][1]	20	[0][2]	30	[0][3]	40
	[1][0]	50	[1][1]	60	[1][2]	70	[1][3]	80
	[2][0]	90	[2][1]	100	[2][2]	110	[2][3]	120

↑ 열

#### 메모리 운영

##### Stack 영역

변수	번지참조
Int [4][3] human	100번지

메모리위치  
참조

##### Heap 영역

변수	값
100번지	210번지, 220번지, 230번지

메모리위치  
참조

변수	값
210번지	10,20,30,40
220번지	50,60,70,80
230번지	90,100,110,120