

● 선언과 생성: String[] 참조변수 = new String[크기]; 배열의 이름인 참조변수를 위한 공간이 마련되고(선언), 선언된 크기만큼 String타입의 참조변수를 저장하기 위한 공간(요소)이 마련된다(생성). String은 참조형이고, 참조형 변수의 기본값은 null이므로 각 요소의 값은 null로 초기화 된다. null은 어떠한 객체도 가리키고 있지 않다는 뜻이다.

- String은 클래스이므로 new 연산자를 통해 객체를 생성해야 하지만, 특별히 String 클래스만 큰따옴표만으로도 간략히 표현하는 것이 허용된다.

- 참조형 배열의 경우, 배열의 요소에 저장되는 것은 실제 객체가 아닌 객체의 주소이다. 참조형 배열을 객체 배열이라고도 한다. (String 배열의 참조변수, 즉 배열의 이름에는 배열의 주소가 저장되어 있고, 배열의 각 요소(참조변수)에는 객체의 주소가 저장되어 있다).

- 참조형 변수를 간단히 참조변수라고도 하며, 모든 참조형 변수에는 객체가 메모리에 저장된 주소인 4 byte의 정수 값(0x0~0xffffffff) 또는 null이 저장된다.

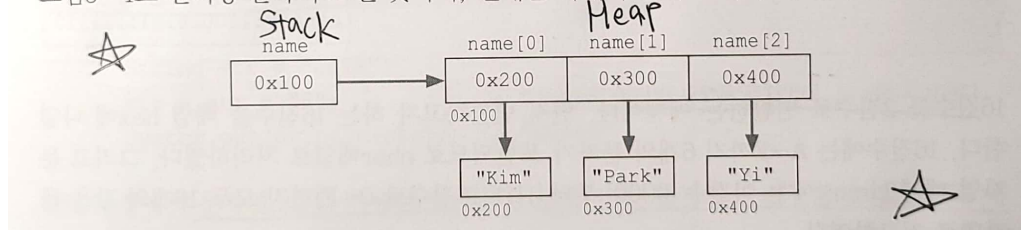
특히 String클래스만 "Kim"과 같이 큰따옴표만으로도 간략히 표현하는 것이 허용되지만, 원래 String은 클래스이므로 아래의 왼쪽처럼 new 연산자를 통해 객체를 생성해야 한다.

```
String[] name = new String[3];
name[0] = new String("Kim");
name[1] = new String("Park");
name[2] = new String("Yi");
```

→

```
String[] name = new String[3];
name[0] = "Kim";
name[1] = "Park";
name[2] = "Yi";
```

그림5-4도 편의상 간략히 그린 것이며, 원래는 아래와 같이 그려야 더 정확한 그림이다.



● String클래스: char배열에 기능(메서드)를 추가한 것이다 (객체지향언어에서는 데이터와 그에 관련된 기능을 하나의 클래스에 묶어서 다룰 수 있기 한다). String객체(문자열)은 읽을 수 있지만 변경할 수는 없다. 그래서 문자열에 새로운 문자열을 더하는 경우 기존의 문자열이 바뀌는 것이 아닌 결과로 나온 새 문자열이 변수에 저장되고, 기존의 문자열은 참조되지 않고 버려져 JVM의 Garbage Collector에 의해 제거된다. (변경 가능한 문자열을 다루려면, StringBuffer클래스를 사용하면 된다).

#### ▶ String 클래스의 주요 메서드

메서드	설명
char charAt(int index)	문자열에서 해당 위치(index)에 있는 문자를 반환한다.
int length()	문자열의 길이를 반환한다.
String substring(int from, int to)	문자열에서 해당 범위(from~to)에 있는 문자열을 반환한다. (to는 범위에 포함되지 않음)
boolean equals(Object obj)	문자열의 내용이 obj와 같은지 확인한다. 같으면 결과는 true, 다르면 false가 된다. (대소문자 구분 x → equalsIgnoreCase())
char[] toCharArray()	문자열을 문자배열(char[])로 변환해서 반환한다.

▲ 표 5-3 String클래스의 주요 메서드

## ▶ String -> int 변환

```
int num = Integer.parseInt(문자열);
```

## ▶ char배열 <-> String클래스 변환

```
char[] chArray = {};  
String str = new String(chArray);           // char배열 -> String  
char[] chFinal = str.toCharArray();         // String -> Char배열
```

## ▶ char배열과 String배열의 길이는 .length 메서드를 통해 얻을 수 있다.

● 커맨드 라인을 통해 입력 받기: 프로그램을 실행할 때 클래스 이름 뒤에 공백문자로 구분하여 여러 개의 문자열을 호면을 통해 사용자로 부터 입력 받아 프로그램에 전달할 수 있다. 커맨드라인을 통해 입력된 두 문자열은 String배열에 담겨서 해당 클래스의 main method의 매개변수(args)에 전달된다. main메서드 내에서 args[0], args[1]...와 같은 방식으로 커맨드라인으로부터 전달받은 문자열에 접근하여 원하는 기능을 수행할 수 있다. 커맨드라인에 입력된 매개변수는 공백문자로 구분하기 때문에 입력될 값에 공백이 있는 경우 큰따옴표(")로 감싸주어야 한다. 커맨드라인에서 숫자를 입력해도 자동으로 문자열로 처리되어 전달된다. 커맨드라인에 매개변수를 입력하지 않으면 크기가 0인 배열이 생성되어 args.length의 값은 0이 된다. 크기가 0인 배열을 생성하는 것도 가능하다. 만일 입력된 매개변수가 없다고 해서 배열을 생성하지 않으면 참조변수 args의 값은 null이 될 것이고, 배열 args를 사용하는 모든 코드에서 에러가 발생할 것이다. 그러나 JVM이 입력된 매개 변수가 없을 때 null 대신 크기가 0인 배열을 생성해서 args에 전달되도록 구현한다. 만약 커맨드라인으로부터 입력받은 숫자로 사칙연산을 수행하려면, 먼저 문자열을 숫자와 문자로 변환이 필요하며, Integer.parseInt()를 사용해서 가능하다.

1) Alt+shift+F10 -> Program arguments에 문자열 입력 -> Run

2) window command창: 윈도우키+R - cmd 입력 - 엔터 - cd 띄우고 클래스 파일 경로(open in - file path, 복사) - dir 입력 - 실행할 클래스 이름 입력 java 띄우고 파일명.class 입력 띄우고 입력할 문자열 입력.

(type 클래스이름.class로 하면 파일의 내용을 볼 수 있다. 그런데 여기서 매개 변수에 원하는 값을 주면 파일의 내용을 바꿀 필요없이 다르게 실행 가능하다).

Resource: 자바의 정석