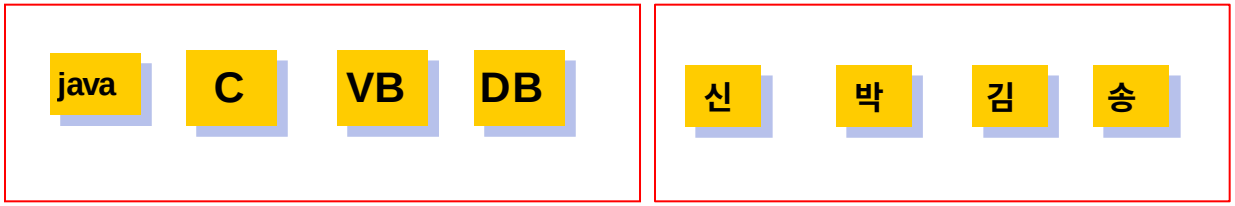
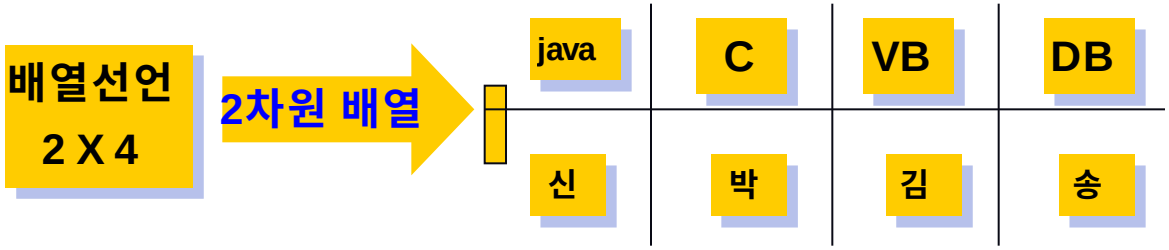
참조형(Reference type)

- **참조형 (Reference type)**
  - C의 포인터와 같은 기능을 수행
  - 기본 자료형은 변수를 선언하면 메모리에 공간이 할당된다.
  - 참조형은 객체를 생성하기 전까지는 메모리 공간이 할당 되지 않는다.

변수(Variable)

- **변수**
  - 프로그램이 실행되는 동안 정보(값)를 저장하는 기억 장소
  - 기억장소의 정보를 변경 가능
- **변수의 종류**
  - 클래스 변수
  - 객체(인스턴스) 변수
  - 지역 변수
- **배열**
  - 기억장소를 정의하는 이름
  - 동일한 자료형
  - 연속적인 자료형





배열 선언(Array)

- **배열**
  - 같은 자료형의 연속된 표현
  - 기억장소가 연속적으로 할당됨
  - Java에서는 선언된 배열의 크기를 넘는 배열의 참조를 허용하지 않음.
    - 에러(예외상황 : exception) 발생
- **일차원, 다차원 배열의 선언**
  - int number[ ] ;
  - char [ ] letters ;
  - int grid [ ] [ ] ;
- **배열의 크기(객체생성, 초기화)**
  - char alphabet [ ] = new char [26] //26개 기억장소 생성
  - int[ ] primes = {7,5,10} // 3개 기억장소 생성

배열 생성(Array)

Ü 배열 객체의 초기화

예제 Ch0304AryInit

- 숫자 : 0
- boolean : false
- 문자 : '\0'
- 객체 : NULL

첨자(index)는
0부터 시작

```
배열형 배열명 = 배열 성분 표현식 (예)

  int[ ] arrayOfInts = ; // compile error
  int[ ] simple = {0, 1, 2, 3, 4};
  int simple[ ] = {0, 1, 2, 3, 4};
```

배열 사용(Array)

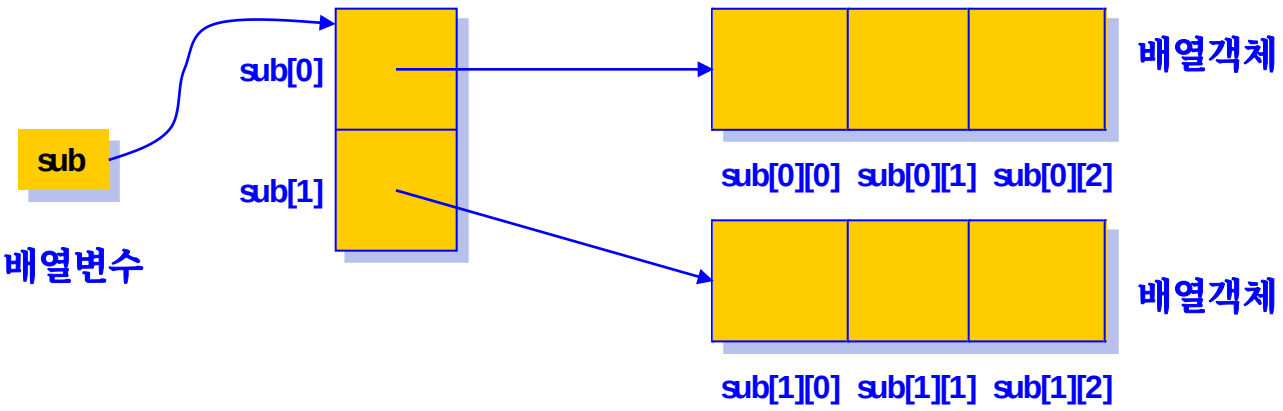
- **배열의 크기** 예제 Ch0318ExX3int(p.20)
  - 배열명.length를 이용
  - length
    - 모든 배열 객체는 length라는 객체 변수를 가지고 있다.
    - 배열의 원소 수 (배열이 가질 수 있는 단위 공간)를 담고 있다.
  - 배열 첨자의 크기 = length - 1
  - Len의 값은 10

```
String[] str = new String[10];

int len = str.length;
```

다차원 배열

```
int [][] sub;                          // 선언
sub = new int[2][3];                   // 생성 또는
int [][] sub = new int [2][3]; //선언 과 생성
```

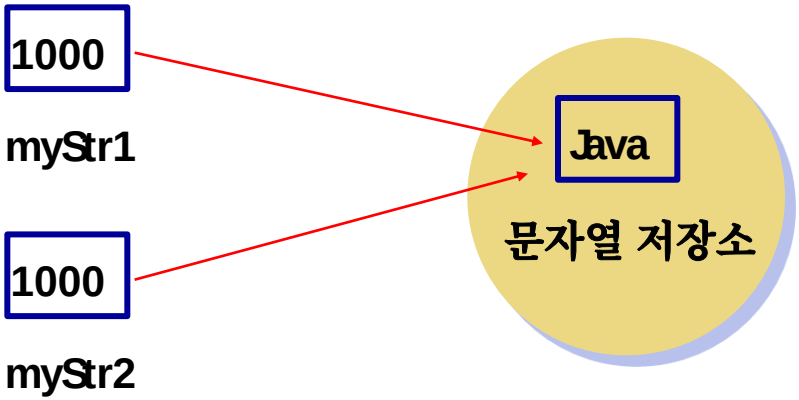


참조형 자료형 - String 클래스

- **String 클래스**
  - 여러개 문자 저장
  - 이중 인용부호 사용
  - String myname="ts lee";
- **char 자료형**
  - 문자 한 개 저장
  - 싱글 인용부호 사용
  - char Name= 'j';

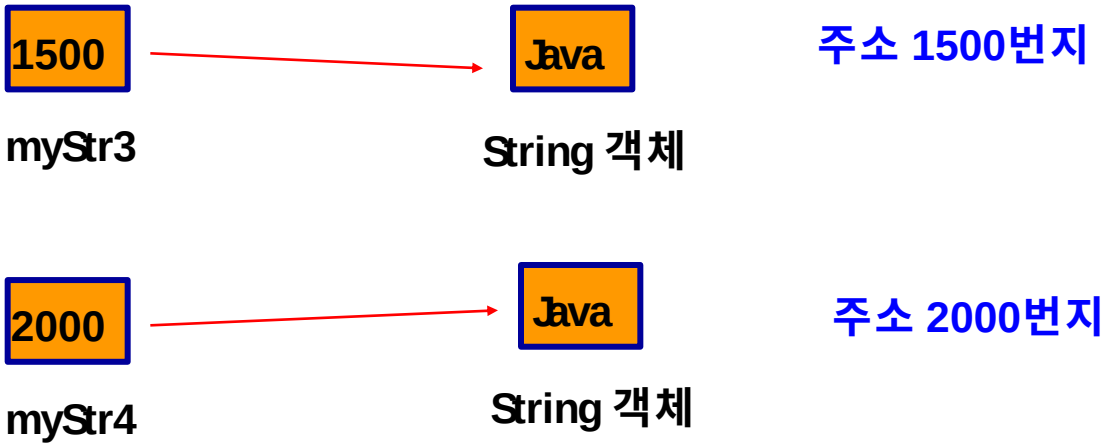
String 클래스 문자열(1)

- 문자열 상수를 지정하는 방법
  - String myStr1 = “java”
  - String myStr2 = “java”
- 생성자(new)를 이용하는 방법
  - String myStr3 = new String(“java”)
  - String myStr4 = new String(“java”)
- 문자열 상수를 선언한 문자열의 저장 형태



String 클래스 문자열(2)

- Ü 생성자를 선언한 문자열의 저장 형태
  - q myStr3 != myStr4



String 클래스 예제

예제 Ch0304EqualsString

```
public class EqualsString {
   public static void main(String[] args) {
      String mystr1 = "java";
      String mystr2 = "java";
      String mystr3 = new String("java");
      String mystr4 = new String("java");
      System.out.println("[ == 비교 결과   ");
      System.out.println("mystr1 = mystr2 : "+(mystr1 == mystr2));
      System.out.println("mystr3 = mystr4 : "+(mystr3 == mystr4));
      System.out.println("mystr1 = mystr3 : "+(mystr1 == mystr3));
      System.out.println("mystr1 = mystr4 : "+(mystr1 == mystr4));
      System.out.println("[ equals 메소드 비교 결과 ");
      System.out.println("mystr1 = mystr2 : " + (mystr1.equals(mystr2)));
      System.out.println("mystr3 = mystr4 : " + (mystr3.equals(mystr4)));
      System.out.println("mystr1 = mystr3 : " + (mystr1.equals(mystr3)));
      System.out.println("mystr1 = mystr4 : " + (mystr1.equals(mystr4)));
      }
}
```

참조형 자료형의 연산

- **String 클래스 인스턴스 간의 “+” 연산자**
  - 문자열의 결합
  - String str1 = “java”
  - str1 = str1 + “Powerful” -> “javaPowerful”
- **String 클래스와 기본 자료형과의 “+” 연산자**
  - String str1 = “java” + 1000 -> “java1000”

String 클래스 메소드

메 소 드	설 명
int length()	String 클래스의 문자열 길이를 반환
boolean equals(String str)	저장된 문자열과 str 문자열의 내용이 같은지를 비교
boolean equalsIgnoreCase(String str)	대소문자 구별없이 저장된 문자열과 str 문자열의 내용이 같은지를 비교
String substring(int beginindex)	문자열의 beginindex 위치부터 마지막까지의 문자열을
String concat(String str)	저장된 문자열과 str문자열을 결합
String replace(char old, char new)	문자열 내의 old 문자를 new 문자로 변경
String toLowerCase()	String 클래스 객체의 문자열을 소문자로 변경
String toUpperCase()	String 클래스 객체의 문자열을 대문자로 변경
char charAt(int index)	index 위치의 char 변수값(문자)을 반환
int indexOf(int ch)	저장된 문자열의 첫 번째 ch 문자의 위치를 반환
int lastIndexOf(int ch)	저징된 문자열의 마지막 ch 문자의 위치를 반환
String trim()	문자열 끝의 공백 문자를 제거

String 클래스 예제2

예제 Ch0305StringMethod

```
class StringMethod{
	public static void main(String[] args) {
		int alength, blength;
		char achar, bchar;
		String str1 = new String("java Powerful");
		String str2 = new String("java Programming");
		String str3 = str1 + str2;
		alength = str1.length();
		blength = str2.length();
		achar = str1.charAt(5);
		bchar = str2.charAt(10);
		System.out.println("str1에 저장 = "+str1 + "  길이= "+alength);
		System.out.println("str2에 저장 = "+str2 + "  길이= "+blength);
		System.out.println("str1의 5번째 문자 = "+achar);
		System.out.println("str2의 10번째 문자 = "+bchar);
		System.out.println("str1의 대문자로 변환 = "+str1.toUpperCase());
		System.out.println("str2의 a를 A로 변환 = "+str2.replace('a','A'));
	}
}
```

매개변수에 있어서 기본자료형과 참조자료형의 차이 예제
Ch0305TestRef(별도 제공)