CSEN202 – Introduction to Computer Programming

Topics:

Array of Objects
Command Line Arguments
Two-Dimensional Arrays

Prof. Dr. Slim Abdennadher

Arrays of Objects

- **An array can store object references**
- **This code declares and creates an array of ten `String` references:**

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

- **This code does not create the individual objects in the array: The elements of `words` are all `null` references**
- **The individual elements in an array of objects must be created separately**

Arrays of Objects – Example

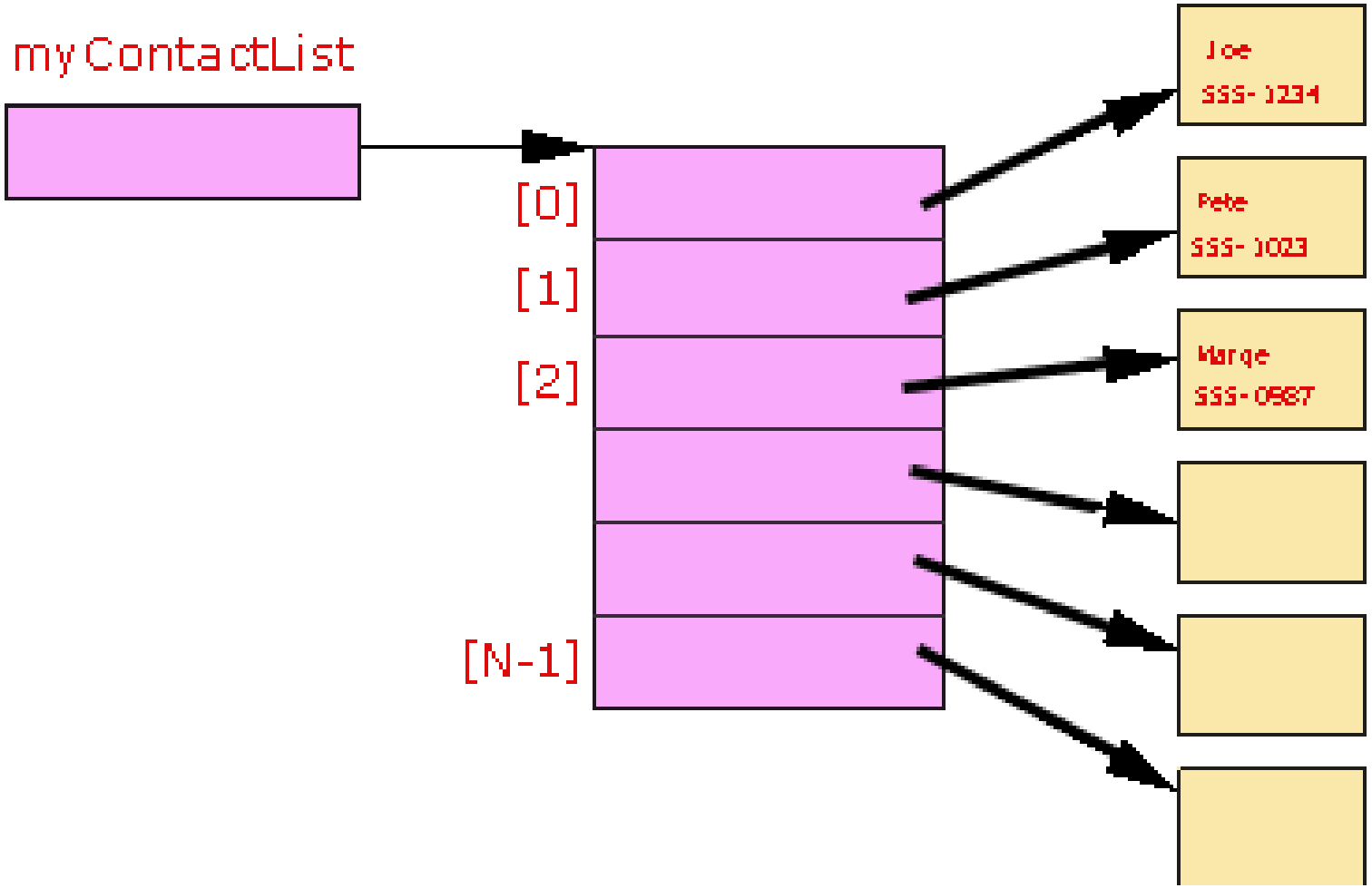
- **An Array of Objects in Java:**

```
int N = 100;
Contact[] myContactList = new Contact[N];
System.out.println(myContactList.length);
```

- **Create the elements in the array with `new`:**

```
myContactList[0] = new Contact("Joe","555-1234");
myContactList[1] = new Contact("Peter","555-1023");
myContactList[2] = new Contact("Marge","555-0987");
```

Arrays of Objects – Example



Command Line Arguments: String[] args

- **The familiar signature of the `main()` method:**

```
public static void main( String[] args )
```

- **`String[] args` says that `main()` has a parameter which is an array of String references.**
- **The elements of the array refer to Strings that contain text from the command line that starts the program.**
- **Example:**

```
class StringDemo {
  public static void main ( String[] args ) {
    for (int j=0; j  < args.length; j++ )
      System.out.println( "Parameter " + j + ": " + args[j] );
  } }
```

- **Output:**

```
C:\>java StringDemo Hello World
Parameter 0: Hello
Parameter 1: World
```

Command Line Arguments: Numeric Input

- **The command line data are always Strings**
- **Use `Integer.parseInt(String)` or `Double.parseDouble(String)` to convert the character data to numeric data.**
- **Example:**

```
class InputDemo
{
  public static void main ( String[] args )
  {
    int sum = 0;
    for (int j=0; j  < args.length; j++ )
      sum += Integer.parseInt(  args[j]  );

    System.out.println( "Sum: " + sum );
  }
}
```

Two-dimensional Arrays

- **Since an array is itself an object reference, arrays of arrays are possible.**
- **Arrays of arrays are more commonly called multidimensional arrays.**
- **Here `a` is declared to be an array of integer arrays: `int[][] a;`**
- **It is convenient to visualize this array of arrays as a two-dimensional structure like a table.**

Two-dimensional Arrays

```
int[][] a;
a = new int[2][3];
a[0][0] = 5;
a[0][1] = 10;
a[0][2] = 3;
a[1][0] = 25;
a[1][1] = 11;
a[1][2] = 13;
```

	0	1	2
0	5	10	3
1	25	11	13

Accessing an Element and Creating 2D Arrays

- **A double index is required to access an element of 2D-array:**
  - **The first index is the row**
  - **The second index is the column**
- **Alternatives to create 2D arrays:**
  - ```
    int [][] a;
    a = new int[][]{new int[] {5, 10, 3},
                    new int[] { 25, 11, 13}};
    ```
  - ```
    int [][] a = { {5, 10, 3}, { 25, 11, 13} } ;
    ```

2D Arrays Details

Since

- **a 2D array is an array of 1D arrays and**
- **each of these 1D arrays are treated as rows,**

the length attribute of a 2D array specifies the number of rows.

Example:

```
int[][] a;
```

- `a.length` **represents the number of rows in array** `a`**.**
- `a[i]` **represents row** `i`**, itself an array.**
- `a[i].length` **is the number of elements in row** `i`
- `a[i][j]` **represents an integer element at row** `i` **and column** `j`**.**

Ragged Arrays

- **Two-dimensional arrays can have different numbers of columns in each row.**
- **How to declare a ragged array?**

```
int[][] b;
b = new int[3][];
b[0] = new int[5];
b[1] = new int[7];
b[2] = new int[4];
```

Ragged Arrays – Example

```
1
0 1
0 0 1
```

- **The following code declares a (lower) triangular array in which row `a[i]` has `i+1` elements.**

```
int[][] a;
a = new int[3][];
for (int i = 0; i < a.length; i++) {
        a[i] = new int[i+1];
}
```

- **The following code initializes an array where only the diagonal cells consist of 1's otherwise all cells have the value 0.**

```
for (int i = 0; i < a.length; i++) {
    for (int j = 0; j < i; j++) {
            a[i][j] = 0;
    }
    a[i][i] = 1;}
```