

EXAMPLES

(Arrays)

Arrays of objects

- It is also possible to create array of objects
- We will see how to create and manipulate an array of strings
 - Read names from the user
 - Display names on the screen
 - Search the given name

```
import java.util.*;

public class Names{
    public static void main(String[] args){
        String [] names;
        int result;

        names = readNames();
        displayNames(names);

        result = search(names, "Cigdem");
        if (result == -1)
            System.out.println("not found");
        else
            System.out.println("at index " + result);
    }
    // implement the helper methods here
    // ...
}
```

```
// This method reads the names from the user and
// fills them in a string array
private static String[] readNames(){
    Scanner sc = new Scanner(System.in);
    String[] N;
    int no;

    System.out.print("Enter the number of names: ");
    no = sc.nextInt();
    N = new String [no];
    for (int i = 0; i < no; i++){
        System.out.print("Enter a name: ");
        N[i] = sc.next();
    }
    return N;
}
```

```
// This methods displays the contents of the string
// array on the screen
private static void displayNames(String[] N) {
    for (String str: N)
        System.out.println (str);
}
```

```
// This method searches the given value in the
// string array. If the value is found, it returns
// its corresponding index number. Otherwise, it
// returns -1
private static int search(String[] N, String value) {
    for (int i = 0; i < N.length; i++)
        if (N[i].equals(value))
            return i;
    return -1;
}
```

Two-dimensional arrays

- It is also possible to use two-dimensional arrays
- We will see how to implement some of the matrix operations in Java
 - Read matrix elements
 - Display matrix elements on the screen
 - Perform element-by-element addition
 - Find the column-sum

```
import java.util.*;

public class MatrixOperations{
    public static void main(String[] args){
        double [][] m1, m2, result;
        double [] sum;

        m1 = read();
        m2 = read();
        result = add(m1,m2);
        display(result);

        sum = columnSum(m1);
        for (double v: sum)
            System.out.print(v + " ");
        System.out.println();
    }
    // implement the helper methods here
    // ...
}
```

```

// It reads the matrix elements from the user
private static double[][] read(){
    Scanner sc = new Scanner(System.in);
    double[][] M;
    int row, column;

    System.out.print("Enter the number of rows: ");
    row = sc.nextInt();
    System.out.print("Enter the number of columns: ");
    column = sc.nextInt();

    M = new double [row][column];
    for (int i = 0; i < M.length; i++)
        for (int j = 0; j < M[0].length; j++){
            System.out.print("Enter a value: ");
            M[i][j] = sc.nextDouble();
        }
    return M;
}

```



```
// It displays the contents of the matrix on the
// screen
private static void display(double[][] M) {

    for (int i = 0; i < M.length; i++) {
        for (int j = 0; j < M[0].length; j++)
            System.out.print(M[i][j] + " ");
        System.out.println();
    }

}
```

```
// It computes the sum of two matrices. In this
// method, we assume that the row and column numbers
// of the operands are the same.
private static double[][] add(double[][] A,
                               double[][] B) {
    double[][] result;

    result = new double[A.length][A[0].length];
    for (int i = 0; i < result.length; i++)
        for (int j = 0; j < result[0].length; j++)
            result[i][j] = A[i][j] + B[i][j];

    return result;
}
```

```
// It computes the column sum of the matrix;  
// i.e., for each column, it computes the values in  
// all rows  
private static double[] columnSum(double[][] M) {  
    double[] sum = new double [M[0].length];  
  
    for (int i = 0; i < M[0].length; i++) {  
        sum[i] = 0.0;  
  
        for (int j = 0; j < M.length; j++)  
            sum[i] += M[j][i];  
    }  
  
    return sum;  
}
```

Two-dimensional arrays

- We can declare a two-dimensional array with a variable number of columns for each row (**ragged arrays**)
- We will implement a program to keep and manipulate the exam grades of a student for different courses. In this question, each course has different number of exams.
 - Take exam grades
 - Calculate the average of the exam grades for each course
 - Display the exam grades and their average for each course

```
import java.util.*;

public class ExamGrades{

    public static void main(String[] args){
        int [][] grades;
        double [] avg;

        grades = readGrades();
        avg = calculateAverage(grades);
        displayGrades(grades, avg);

    }
    // implement the helper methods here
    // ...

}
```

```

private static int[][] readGrades() {
    Scanner sc = new Scanner(System.in);
    int[][] grades;
    int courseNo, examNo;

    System.out.print("Number of courses: ");
    courseNo = sc.nextInt();
    grades = new int [courseNo][ ];

    for (int i = 0; i < grades.length; i++) {
        System.out.print
            ("No of exams for course " + i + ": ");
        examNo = sc.nextInt();
        grades[i] = new int[examNo];
        for (int j = 0; j < grades[i].length; j++) {
            System.out.print("Next grade: ");
            grades[i][j] = sc.nextInt();
        }
    }

    return grades;
}

```

```
private static double[] calculateAverage
                                (int[][] grades){
    double[] avg = new double [grades.length];

    for (int i = 0; i < grades.length; i++){
        avg[i] = 0.0;

        for (int j = 0; j < grades[i].length; j++)
            avg[i] += grades[i][j];

        if (grades[i].length != 0)
            avg[i] /= grades[i].length;
    }
    return avg;
}
```

```
private static void displayGrades(int[][] grades,
                                   double[] avg) {

    for (int i = 0; i < grades.length; i++) {
        System.out.print("Course " + i + "\n\t");

        for (int j = 0; j < grades[i].length; j++)
            System.out.print(grades[i][j] + "\t");

        System.out.print("\n\tAverage: " + avg[i]);
        System.out.println();
    }
}
```