Arrays of Objects & 2D Arrays

CSC 1051 – Data Structures and Algorithms I
Dr. Mary-Angela Papalaskari
Department of Computing Sciences
Villanova University

Course website:
www.csc.villanova.edu/~map/1051/

Some slides in this presentation are adapted from the slides accompanying Java Software Solutions by Lewis & Loftus
Outline

Declaring and Using Arrays

Arrays of Objects

Variable Length Parameter Lists

Two-Dimensional Arrays

Polygons and Polylines

Mouse Events and Key Events
Example: An array of Strings

• `String[] words = new String[5];`

At this point, the following line of code would throw a `NullPointerException`:

```java
System.out.println(words[0]);
```
Example: An array of Strings

- `String[] words = new String[5];`
- Now, store some `String` objects in the array:

```java
words[0] = "friendship";
words[1] = "loyalty";
words[2] = "honor";
```
Arrays of Objects

• The following declaration creates an array object called `verbs` and fills it with four `String` objects created using string literals

```java
String[] verbs = {"play", "work", "eat", "sleep", "run"};
```
Arrays of Objects

• Example: managing a collection of DVD objects
public class Movies
{
    public static void main (String[] args)
    {
        DVDCollection movies = new DVDCollection();

        movies.addDVD ("The Godfather", "Francis Ford Coppala", 1972, 24.95, true);
        movies.addDVD ("District 9", "Neill Blomkamp", 2009, 19.95, false);
        movies.addDVD ("Iron Man", "Jon Favreau", 2008, 15.95, false);
        movies.addDVD ("All About Eve", "Joseph Mankiewicz", 1950, 17.50, false);
        movies.addDVD ("The Matrix", "Andy & Lana Wachowski", 1999, 19.95, true);

        System.out.println (movies);

        movies.addDVD ("Iron Man 2", "Jon Favreau", 2010, 22.99, false);
        movies.addDVD ("Casablanca", "Michael Curtiz", 1942, 19.95, false);

        System.out.println (movies);
    }
}
public class Movies {
  
  // Creates a DVDCollection object and adds some DVDs to it. Prints reports on the status of the collection.
  
  public static void main (String[] args) {
    DVDCollection movies = new DVDCollection();
    movies.addDVD ("The Godfather", "Francis Ford Coppola", 1972, 24.95, true);
    movies.addDVD ("District 9", "Neill Blomkamp", 2009, 19.95, false);
    movies.addDVD ("Iron Man", "Jon Favreau", 2008, 15.95, false);
    movies.addDVD ("All About Eve", "Joseph Mankiewicz", 1950, 17.50, false);
    movies.addDVD ("The Matrix", "Andy & Lana Wachowski", 1999, 19.95, true);

    System.out.println (movies);

    movies.addDVD ("Iron Man 2", "Jon Favreau", 2010, 22.99, false);
    movies.addDVD ("Casablanca", "Michael Curtiz", 1942, 19.95, false);

    System.out.println (movies);
  }
}

Output

My DVD Collection

Number of DVDs: 5
Total cost: $98.30
Average cost: $19.66

DVD List:

$24.95 1972 The Godfather Francis Ford Coppala Blu-Ray
$19.95 2009 District 9 Neill Blomkamp
$15.95 2008 Iron Man Jon Favreau
$17.50 1950 All About Eve Joseph Mankiewicz
$19.95 1999 The Matrix Andy & Lana Wachowski Blu-Ray

continue
public class Movies {
    public static void main (String[] args) {
        DVDCollection movies = new DVDCollection();
        movies.addDVD ("The Godfather", "Francis Ford Coppala", 1972, 24.95, true);
        movies.addDVD ("District 9", "Neill Blomkamp", 2009, 19.95, false);
        movies.addDVD ("Iron Man", "Jon Favreau", 2008, 15.95, false);
        movies.addDVD ("All About Eve", "Joseph Mankiewicz", 1950, 17.50, false);
        movies.addDVD ("The Matrix", "Andy & Lana Wachowski", 1999, 19.95, true);
        System.out.println (movies);
        movies.addDVD ("Iron Man 2", "Jon Favreau", 2010, 22.99, false);
        movies.addDVD ("Casablanca", "Michael Curtiz", 1942, 19.95, false);
        System.out.println (movies);
    }
}
import java.text.NumberFormat;

public class DVDCollection {
    private DVD[] collection;
    private int count;
    private double totalCost;

    // Constructor: Creates an initially empty collection.
    public DVDCollection () {
        collection = new DVD[100];
        count = 0;
        totalCost = 0.0;
    }
}

continue
public void addDVD (String title, String director, int year, double cost, boolean bluRay)
{
    if (count == collection.length)
        increaseSize();
    collection[count] = new DVD (title, director, year, cost, bluRay);
    totalCost += cost;
    count++;
}
/-----------------------------------------------------------------
//  Returns a report describing the DVD collection.
-----------------------------------------------------------------
public String toString()
{
    NumberFormat fmt = NumberFormat.getCurrencyInstance();

    String report = "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    My DVD Collection
    Number of DVDs: " + count + "
    Total cost: " + fmt.format(totalCost) + "
    Average cost: " + fmt.format(totalCost/count);

    report += "\n\nDVD List:\n\n";

    for (int dvd = 0; dvd < count; dvd++)
        report += collection[dvd].toString() + "\n";

    return report;
}
private void increaseSize ()
{
    DVD[] temp = new DVD[collection.length * 2];

    for (int dvd = 0; dvd < collection.length; dvd++)
        temp[dvd] = collection[dvd];

    collection = temp;
}
import java.text.NumberFormat;

public class DVD {
    private String title, director;
    private int year;
    private double cost;
    private boolean bluRay;

    // Creates a new DVD with the specified information.
    public DVD (String title, String director, int year, double cost, boolean bluRay) {
        this.title = title;
        this.director = director;
        this.year = year;
        this.cost = cost;
        this.bluRay = bluRay;
    }
}
public String toString()
{
    NumberFormat fmt = NumberFormat.getCurrencyInstance();

    String description;

    description = fmt.format(cost) + "\t" + year + "\t";
    description += title + "\t" + director;

    if (bluRay)
        description += "\t" + "Blu-Ray";

    return description;
}
Outline

Declaring and Using Arrays
Arrays of Objects
Variable Length Parameter Lists
Two-Dimensional Arrays
Polygons and Polylines
Mouse Events and Key Events
Two-Dimensional Arrays

- A *one-dimensional array* stores a list of elements
- A *two-dimensional array* can be thought of as a table of elements, with rows and columns
Two-Dimensional Arrays

• To be precise, in Java a two-dimensional array is an array of arrays

• A two-dimensional array is declared by specifying the size of each dimension separately:

```java
int[][] table = new int[12][50];
```

• A array element is referenced using two index values:

```java
value = table[3][6]
```

• The array stored in one row can be specified using one index
double[][][] courseGrade = new double[3][10];

courseGrade

courseGrade[1][4]

2D array element

courseGrade[1][4]
Two-Dimensional Arrays

<table>
<thead>
<tr>
<th>Expression</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>table</td>
<td>int[][]</td>
<td>2D array of integers, or array of integer arrays</td>
</tr>
<tr>
<td>table[5]</td>
<td>int[]</td>
<td>array of integers</td>
</tr>
<tr>
<td>table[5][12]</td>
<td>int</td>
<td>integer</td>
</tr>
</tbody>
</table>

- See [TwoDArray.java](#)
- See [SodaSurvey.java](#)
public class TwoDArray
{
        // Creates a 2D array of integers, fills it with increasing integer values, then prints them out.
    
    public static void main (String[] args)
    {
        int[][] table = new int[5][10];

        // Load the table with values
        for (int row=0; row < table.length; row++)
            for (int col=0; col < table[row].length; col++)
                table[row][col] = row * 10 + col;

        // Print the table
        for (int row=0; row < table.length; row++)
        {
            for (int col=0; col < table[row].length; col++)
                System.out.print (table[row][col] + "\t");
            System.out.println();
        }
    }
}
public class TwoDArray {
    public static void main (String[] args) {
        int[][] table = new int[5][10];

        // Load the table with values
        for (int row=0; row < table.length; row++)
            for (int col=0; col < table[row].length; col++)
                table[row][col] = row * 10 + col;

        // Print the table
        for (int row=0; row < table.length; row++)
            { for (int col=0; col < table[row].length; col++)
                  System.out.print (table[row][col] + "\t");
                  System.out.println();
            }
    }
}
Multidimensional Arrays

• An array can have many dimensions – if it has more than one dimension, it is called a multidimensional array

• Each dimension subdivides the previous one into the specified number of elements

• Each dimension has its own length constant

• Because each dimension is an array of array references, the arrays within one dimension can be of different lengths
  – these are sometimes called ragged arrays