Arrays of Objects

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
Declaring and Using Arrays

Arrays of Objects

Variable Length Parameter Lists

Two-Dimensional Arrays

Polygons and Polylines

Mouse Events and Key Events
Arrays - review

• Declaration:

```
int[] scores = new int[10];
```

The entire array has a single name.

scores[2] = 94;

This array holds 10 values that are indexed from 0 to 9:

```
79 87 94 82 67 98 87 81 74 91
```

scores.length = 10
Initializer Lists

- An *initializer list* can be used to instantiate and fill an array in one step

- The values are delimited by braces and separated by commas

- Examples:

  ```
  int[] units = {147, 323, 89, 933, 540, 269, 97, 114, 298, 476};
  char[] grades = {'A', 'B', 'C', 'D', 'F'};
  ```
Initializer Lists

• Note that when an initializer list is used:
  – the new operator is not used
  – no size value is specified

• The size of the array is determined by the number of items in the list

• An initializer list can be used only in the array declaration

• See Primes.java
public class Primes {
    public static void main (String[] args) {
        int[] primeNums = {2, 3, 5, 7, 11, 13, 17, 19};

        System.out.println ("Array length: " + primeNums.length);
        System.out.println ("The first few prime numbers are:");

        for (int prime : primeNums)
            System.out.print (prime + " ");
    }
}
public class Primes {
    public static void main (String[] args) {
        int[] primeNums = {2, 3, 5, 7, 11, 13, 17, 19};

        System.out.println ("Array length: " + primeNums.length);
        System.out.println ("The first few prime numbers are:");

        for (int prime : primeNums) {
            System.out.print (prime + " ");
        }
    }
}
Arrays as Parameters

An entire array can be passed as a parameter to a method (just like any other object).

- **Exercise:** Write a method that increments the value of each element in an array. (Lab 12 (e))

- **Exercise:** Write a method that finds and returns the largest element in an array of double

- **Exercise:** Write a method that has two arrays of equal size as parameters and returns an array of twice that size, with the elements of the two arrays interleaved. For example, if we call our method `interleave(a, b)`, and it is invoked with arrays `a, b` initialized as follows:

  ```java
  int[] a = {2, 3, 5};
  int[] b = {0, 1, 10};
  ```

  it will return the array: `{2, 0, 3, 1, 5, 10}`
Command-Line Arguments

• It turns out we have been using arrays as parameters all along!

```java
public static void main (String[] args)
```
Command-Line Arguments

• It turns out we have been using arrays as parameters all along!

```java
public static void main (String[] args)
{
    System.out.println ();
    System.out.println ("  " + args[0]);
    System.out.println ("  " + args[1]);
}
```

• These values come from command-line arguments that are provided when the interpreter is invoked.

• jGrasp calls them “Run Arguments”
Arrays of Objects

- Example: An array of Strings

  ```java
  String[] words = new String[5];
  ```

- It does NOT create the `String` objects themselves
Arrays of Objects

• The `words` array when initially declared:

![Diagram of an empty array]

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

```java
System.out.println(words[0]);
```
Arrays of Objects

- After some `String` objects are created and stored in the array:

```java
words[0] = "friendship";
words[1] = "loyalty";
words[1] = "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 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 {
    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++;
}

continue
// Returns a report describing the DVD collection.
public String toString()
{
    NumberFormat fmt = NumberFormat.getCurrencyInstance();

    String report = "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    " + "My DVD Collection\n\n";

    report += "Number of DVDs: " + count + "\n";
    report += "Total cost: " + fmt.format(totalCost) + "\n";
    report += "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;
    }

    continue
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;
}
Homework

• Review Section 8.1-8.3
• Read Section 8.4 to prepare for next class

Exercises Handout