Arrays, Part 2

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

Arrays - Review

• Declaration:
  double[] scores = new double[10];

The entire array
has a single name

• Initialization:
  scores[2] = 7.9;
  scores[1] = 8.7;
  scores[2] = 9.4;
  scores[3] = 8.2;
  scores[4] = 6.7;
  scores[5] = 9.8;
  scores[6] = 8.7;
  scores[7] = 8.1;
  scores[8] = 7.4;
  scores[9] = 9.1;

• Instantiation:
  double[] scores = new double[10];

Sizes of array

Initializer Lists

• Alternative way to declare, instantiate, and initialize an array. For example:
  int[] ratings = {4, 3, 3, 1, 4, 2, 1, 0, 3, 4};
  char[] grades = {'A', 'B', 'C', 'D', 'F'};

• NOTE:
  – the new operator is not used
  – size of array is determined by the number of items listed
  – can only be used in the array declaration

try this with the vowel array

The “for-each” Loop

• A simple way of processing every array element:
  for (double score : scores)
    System.out.println(score);

NOTE:
• Only appropriate when processing all array elements starting at index 0
• It can't be used to set the array values

try this with the vowel array
Another example

String[] animals = {"dog", "cat", "mouse", "fox"};
for (String word: animals)
    System.out.println("The " + word + " ate the cake");

for (String word: animals)
    for (String otherWord: animals)
        System.out.println("The " + word + " ate the " + otherWord);

Try this: Use the “for each” loop to scan through an array of int containing ratings (range: 0 - 4) and count up how many 4’s.

int[] ratings = {4, 3, 3, 1, 4, 3, 1, 0, 3, 4};

More array examples (see textbook):
• BasicArray.java
• Primes.java
• ReverseOrder.java
• LetterCount.java
import java.util.Scanner;
public class ReverseOrder {
    // Reads a list of numbers from the user, storing them in an array, then prints them in the opposite order.
    public static void main (String[] args) {
        Scanner scan = new Scanner (System.in);
        double[] numbers = new double[10];
        System.out.println ("The size of the array: " + numbers.length);
        continue
        for (int index = 0; index < numbers.length; index++) {
            System.out.print ("Enter number " + (index+1) + ": ");
            numbers[index] = scan.nextDouble();
        }
        System.out.println ("The numbers in reverse order:");
        for (int index = numbers.length-1; index >= 0; index--)
            System.out.print (numbers[index] + "  ");
    }
}

Sample Run
The size of the array: 10
Enter number 1: 18.36
Enter number 2: 48.9
Enter number 3: 53.5
Enter number 4: 29.06
Enter number 5: 72.404
Enter number 6: 34.8
Enter number 7: 63.41
Enter number 8: 45.55
Enter number 9: 69.0
Enter number 10: 99.18
The numbers in reverse order: 99.18 69.0 45.55 63.41 34.8 72.404 29.06 53.5 48.9 18.36

Another example: Computing letter frequency counts
Sample run:
Enter a sentence: In Casablanca, Humphrey Bogart never says "Play it again, Sam."
A: 0    a: 10
B: 1    b: 1
C: 1    c: 1
d: 0
e: 3

Let's write a program to do this...
Arrays, Part 2

CSC 1051 Villanova University

Dr. Papalaskari

CSC 1051 M.A. Papalaskari, Villanova University

LetterCount.java
Author: Lewis/Loftus

Demonstrates the relationship between arrays and strings.

import java.util.Scanner;

public class LetterCount{
    public static void main(String[] args){
        final int NUMCHARS = 26;
        Scanner scan = new Scanner(System.in);
        int[] upper = new int[NUMCHARS];
        int[] lower = new int[NUMCHARS];
        char current; // the current character being processed
        int other = 0; // counter for non-alphabetics
        System.out.println("Enter a sentence:");
        String line = scan.nextLine();
        // Count the number of each letter occurrence for
        // int ch = 0; ch < line.length(); ch++
        for (int ch = 0; ch < line.length(); ch++)
            current = line.charAt(ch);
            if (current >= 'A' && current <= 'Z')
                upper[current-'A']++;
            else
                if (current >= 'a' && current <= 'z')
                    lower[current-'a']++;
                else
                    other++;
        // Print the results
        System.out.println();
        for (int letter=0; letter < upper.length; letter++)
        System.out.println();
        System.out.println("Non-alphabetic characters: " + other);
    };
}

Sample Run
Enter a sentence:
In Casablanca, Humphrey Bogart never says "Play it again, Sam."
A: 0  a: 10
B: 1  b: 1
C: 1  c: 1
D: 0  d: 0
E: 0  e: 3
F: 0  f: 0
G: 0  g: 2
H: 1  h: 1
I: 1  i: 2
J: 0  j: 0
K: 0  k: 0
L: 0  l: 2
M: 0  m: 2
N: 0  n: 4
O: 0  o: 1
P: 1  p: 1
Q: 0  q: 0
Non-alphabetic characters: 14

Arrays as Parameters

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

// Draws a triangle and a V-shape using polygons and polylines.
public void paintComponent(Graphics page){
    super.paintComponent(page);
    int[] xPoints = {100, 120, 150};
    int[] yPoints = {150, 40, 110};
    page.setColor(Color.cyan);
    page.fillPolygon(xPoints, yPoints, xPoints.length);
    page.setColor(Color.red);
    page.drawPolyline(xPoints, yPoints, xPoints.length);
}

see TrianglePanel.java

CSC 1051 M.A. Papalaskari, Villanova University
Arrays, Part 2

Example: A method that adds 3 to the value of each element in an array of type int[].

```
public void addThree(int[] a)
{
    for (int i = 0; i < a.length; i++)
        a[i] += 3;
}
```

Try this: Write a method that adds n (an int) to the value of each element in an array of type int[].

Try this method with the TrianglePanel:
- invoke it with the array xPoints
- add code to draw another triangle in a different color, using the updated array xPoints

Command-Line Arguments
- It turns out we have been using arrays as parameters all along!

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

Command-Line Arguments
- It turns out we have been using arrays as parameters all along!

```
public class Test {
    public static void main (String[] args)
    {
        System.out.println();
        System.out.println("  "+args[0]);
        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”
What does it mean to “copy an array”?  

- Suppose we have two arrays: 
  ```java
  int[] a = {147, 323, 89, 933};
  int[] b = {100, 200, 300, 400};
  ```

### Copying elements vs. copying array variables:

```java
for (int i=0; i<a.length; i++)
  a[i] = b[i];
```

Afterwards, what is the effect of the following?

### 1) Copying elements:

- After copying elements:
  ```java
  a = b;
  for (int i=0; i<a.length; i++)
    a[i] = b[i];
  ```

### What changes?

- After the first line: 
  ```java
  a = b;
  ```

- After the second line: 
  ```java
  a[1] = 1000;
  ```

### 2) Copying array variables:

```java
a = b;
```  

### What changes?

- After the first line:
  ```java
  a = b;
  ```

- After the second line:
  ```java
  a[1] = 1000;
  ```
Managing a collection of objects

- Example: a Movie database (collection of DVD objects)

```java
public class DVD {
    private String title, director;
    private int year;
    private double cost;
    private boolean bluRay;
    
    // Constructor: 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;
    }
    
    // Returns a string description of this DVD.
    public String toString() {
        NumberFormat fmt = NumberFormat.getCurrencyInstance();
        String description;
        description = fmt.format(cost) + "	" + year + "	" + title + "	" + director;
        if (bluRay) description += "	" + "Blu-Ray";
        return description;
    }
}
```

Test client – create a few DVDs, print their info:

```java
public class TestDVD {
    public static void main(String[] args) {
        DVD one = new DVD("Casablanca", "Michael Curtiz", 1942, 19.95, false);
        DVD two = new DVD("District 9", "Neill Blomkamp", 2009, 19.95, false);
        DVD three = new DVD("Iron Man", "Jon Favreau", 2008, 15.95, false);
        
        System.out.println(one);
        System.out.println(two);
        System.out.println(three);
    }
}
```
What if we want to store more DVDs?

- Use an array of DVD objects:

```java
public class MyTenMovies {
    public static void main(String[] args) {
        DVD one = new DVD("Casablanca", "Michael Curtiz", 1942, 19.95, false);
        DVD two = new DVD("District 9", "Neill Blomkamp", 2009, 19.95, false);
        DVD three = new DVD("Iron Man", "Jon Favreau", 2008, 15.95, false);
        System.out.println(one);
        System.out.println(two);
        System.out.println(three);
    }
}
```

Next: A collection of DVD's that can grow to accommodate as many items as needed!

- No limit like this:

```java
DVD[] list = new DVD[10];
```

Let's store the data in a tab-delimited file:

```
The Godfather  Francis Ford Coppola 1972 24.95 true
District 9     Neill Blomkamp 2009 19.95 false
Iron Man       Jon Favreau 2008 15.95 false
All About Eve  Joseph Mankiewicz 1950 17.50 false
The Matrix     Andy & Lana Wachowski 1999 19.95 true
Iron Man 2     Jon Favreau 2010 22.99 false
Casablanca     Michael Curtiz 1942 19.95 false
```

Client code fragment to input lines from file:

```java
Scanner fileScan = new Scanner(new File(args[0]));
int i = 0;
while (fileScan.hasNextLine() && i < list.length) {
    list[i] = new DVD(fileScan.nextLine());
    i++;
}
```

Remember, we also need:

- import java.util.Scanner;
- import java.io.*;
- throws IOException

where do these go???

We need constructor to handle this.

The current DVD constructor CANNOT be used in this manner:

```java
list[i] = new DVD(fileScan.nextLine());
```

old constructor

```java
this.title = title;
this.director = director;
this.year = year;
this.cost = cost;
this.bluRay = bluRay;
}
```

new constructor

```java
Scanner scan = new Scanner(data);
scan.useDelimiter("\t");
title = scan.next();
director = scan.next();
year = scan.nextInt();
cost = scan.nextDouble();
bluRay = scan.nextBoolean();
```
Managing a collection of objects

• Example: a Movie database (collection of DVD objects)

```
import java.util.Scanner;
import java.io.*;
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) throws IOException {
        Scanner fileScan = new Scanner(new File(args[0]));
        DVDCollection movies = new DVDCollection();
        while (fileScan.hasNextLine())
            movies.addDVD(fileScan.nextLine());
        System.out.println(movies);
    }
}
```

Output

```
My DVD Collection
Number of DVDs: 5
Total cost: $98.30
DVD List:
$24.95  1972  The Godfather  Francis Ford Coppola  Blu-Ray
$19.95  2009  District 9  Neill Blomkamp
$11.95  2008  Iron Man  Jon Favreau
$15.95  2008  All About Eve  Joseph Mankiewicz
$19.95  1999  The Matrix  Andy & Lana Wachowsk Blu-Ray
```
Arrays, Part 2

CSC 1051 Villanova University

Dr. Papalaskari

10

CSC 1051 M.A. Papalaskari, Villanova University

public void addDVD(String info)
{
  if (count == collection.length)
    increaseSize();
  collection[count] = new DVD(info);
  count++;
}

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);
  count++;
}

private void increaseSize()
{
  DVD[] temp = new DVD[collection.length * 2];
  for (int i = 0; i < collection.length; i++)
    temp[i] = collection[i];
  collection = temp;
}