COMP1409: Introduction to Software Development I

Mike Mulder (mmulder10@bcit.ca)

Week 9

Agenda

- Week 8 Review
- Quiz
  - Quiz 8
  - Review Answers
- Logistics
- Lesson 9
  - Two-Dimensional Arrays
  - For Loops
- Lab 9
  - Due Wednesday
  - Should be to finish in-class

Review

1. How do you declare an array variable?

```
private int[]      gradePercentages;
private String[]   studentNames;
```

2. How do you initialize an array?

```
gradePercentages = new int[24];
studentNames = new String[CLASS_SIZE];
```

3. How do you use an array?

```
gradePercentages[FINAL_EXAM] = 90;
return gradePercentages[FINAL_EXAM];
System.out.println(studentNames[index]);
```

Review

1. What is the index for the first element in an array?

```
0
```

2. What is the index of the last element in an array?

```
length-1 (i.e., one less than the maximum number of elements in the array)
```

3. What happens if you try to access an element with a negative index or an index equal to or greater than the maximum number of elements (i.e., >= length)?

```
Array index out of bounds exception
```

Question

What is a limitation of arrays?

The size has to be known upfront (i.e., when we initialize and before we use it).

We will address that Week 12 with an ArrayList.

```
public String[] getProvincesWithPopulationBetween(int min, int max) {

    int i = 0;
    int j = 0;
    int numOfProvWithPop = 0;
    String[] matchingProvinces;

    while (i < provinces.length) {
        if ((provinces[i].getPopulation() >= min) &&
            (provinces[i].getPopulation() <= max)) {
            numOfProvWithPop++;
        }
        i++;
    }

    if (numOfProvWithPop > 0) {
        matchingProvinces = new String[numOfProvWithPop];
    } else {
        return null;
    }

    i = 0;
    while (i < provinces.length) {
        if ((provinces[i].getPopulation() >= min) &&
            (provinces[i].getPopulation() <= max)) {
            matchingProvinces[j] = provinces[i].getName();
            j++;
        }
        i++;
    }

    return matchingProvinces;
}
```

Declare

Initialize

Use (in this case, populate the array)

Quiz 8

Closed book, laptop, phone, etc.

You have a maximum of 20 minutes to complete

Raise your hand when you are done, and I will retrieve your paper

We will review the answers afterwards

Logistics – Remaining Classes

Week	Date	Topics	Comments
11	Nov. 17	More arrays for loops	Quiz 8, Assignment 2 Due
Course Evaluation: To be conducted online during Session 11 prior to the class break.			
12	Nov. 24	ArrayList class Enhanced for (foreach) loop	Quiz 9
13	Dec. 1	Iterators	Quiz 10
14	Dec. 8	Final Exam	Assignment 3 Due

Assignments

Assignment 2

- Marked by Tuesday

Assignment 3

- On D2L now – includes unit test code
- Two classes - Warehouse and Package
- You'll probably want to use an ArrayList (covered next week)

Final Exam Format

Total 3 hours

- First 90 minutes – Written/Closed Book (i.e., big quiz)
- Second 90 minutes – Coding/Open Book (i.e., small assignment)

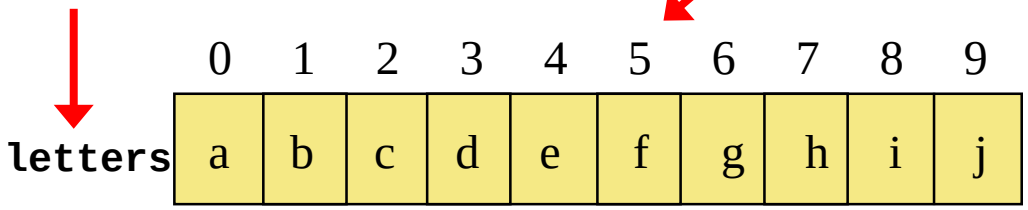
Arrays of Arrays
Two-Dimensional Arrays
for loops

From Last Class: Arrays

An Array is an ordered list of values:

The entire Array has a single name

Each element has an integer *index*, *(always, automatically)*

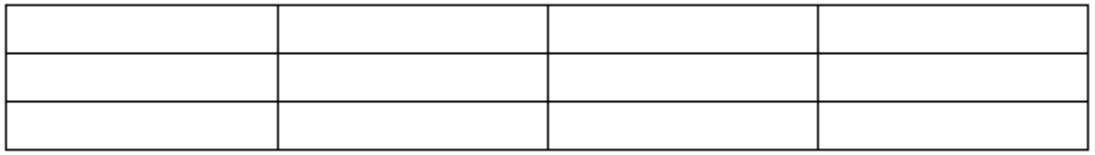


An Array of size n is always indexed from zero to n-1

The Array above contains 10 chars that are indexed from 0 to 9

Two-dimensional arrays

- Used to represent the data in table format (rows and columns)
- Uses two subscripts: rows and columns
- Can be visualized as a matrix or a table:



int[][] a = new int [3][4]; // [num rows][num columns]

- The first subscript indicates the number of rows, the second one is for the number of columns
- a is an array of size three; a is an array of arrays
- a[0] is the top row
- a[0][3] is the top right int
- Each of the three elements of a is an array of four ints

Two-dimensional arrays

- The created array **a**, will have three rows and four columns.
- A total of 12 elements; 12 ints.

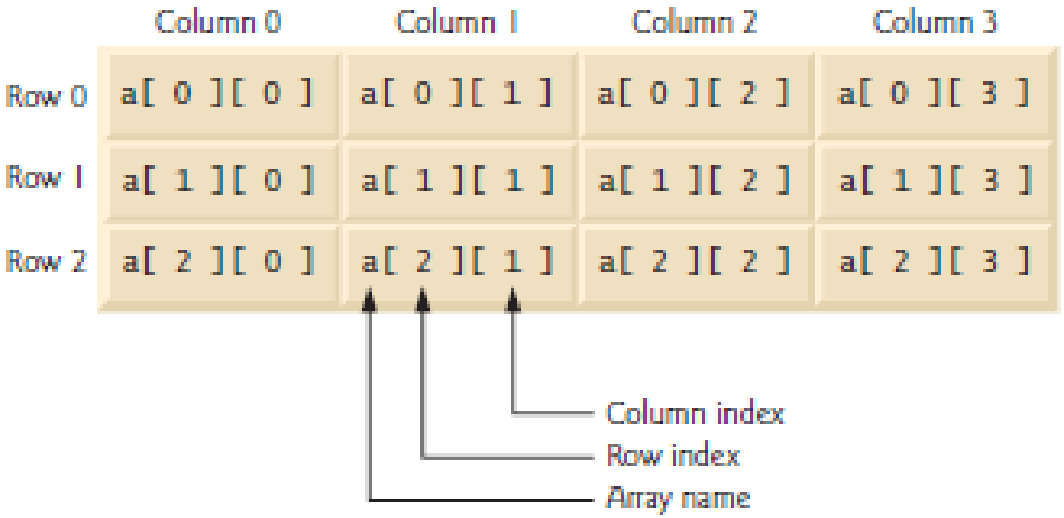


Fig. 7.16 | Two-dimensional array with three rows and four columns.

Steps to Using a Two Dimensional Array

Declare

```
int[][] a;  // local variable
private String[][] b; // instance variable
```

Initialize

```
a = new int[3][4];
private String[][] b = new String[3][4]; // Declare and Initialize
```

Use

```
b[2][3] = “Hello World”;
System.out.println(b[2][3]);
a[0][0] = 5; a[0][1] = 3; a[0][2] = 2; a[0][3] = 4;
int sum = a[0][0] + a[0][1] + a[0][2] + a[0][3];
```

Two-dimensional arrays

- Like the one-dimensional array…
- String[] s = {“hello”, “world”, “the”, “end”};
- …array initializers can be used in declarations to initialize the array when it is being declared:

  int [][] numbers ={{1,2},{3,4}};
- In the above statement numbers[0][0] and numbers[0][1] will be initialized to 1 and 2 respectively. 3 and 4 will initialize numbers[1][0] and numbers [1][1] respectively

Two-dimensional arrays with varying length

- Array rows can be different lengths.
- Array rows all must be the same type; e.g. array of ints

```
int [][] numbers;
numbers = new int[3][];        // defines three int arrays

numbers[0] = new int[4];       // first row holds 4 ints
numbers[1] = new int[500];     // second row holds 500 ints
numbers[2] = new int [7];      // third row holds 7 ints
```

Two-dimensional arrays length

- Can get the length of the rows
- Can get the length of the columns in a row

```
int [][] numbers;
numbers = new int[3][4];      // defines a 3 x 4 int array

numbers.length;         // returns 3
numbers[0].length;      // returns 4
numbers[1].length;      // returns 4
numbers[2].length;      // returns 4
```

Two-dimensional array length (variable length rows)

```
int [][] numbers;
numbers = new int[3][];        // defines three int arrays

numbers[0] = new int[4];       // first row holds 4 ints
numbers[1] = new int[500];     // second row holds 500 ints
numbers[2] = new int [7];      // third row holds 7 ints

numbers.length;          // returns 3
numbers[0].length;       // returns 4
numbers[1].length;       // returns 500
numbers[2].length;       // returns 7
```

Question

When would we use a two dimensional array?

- Large data sets of fixed size, i.e. csv comma separated values) files
- Mathematics – such as matrices

What are some limitations of the two dimensional array?

- Fixed size
- Same type for all elements

for loop

while loop

```
int i = 0;

while(i < names.length){
        System.out.println(names[i]);
        i++;
}
System.out.println(i); // ok
```

for loop

```
for(int i = 0; i < names.length; i++){
        System.out.println(names[i]);
}
System.out.println(i); // error; undefined
```

```
int i = 0;  // Index Variable

i < names.length // Condition on Array Length

System.out.println(names[i]);  // Array Access
```

Example for loop

```
class NameRegistry{
   private String[] names;

   public NameRegistry(){
      names = new String[5];

      names[0] = "Bill";
      names[1] = "Tom";
      names[2] = "Mary";
      names[4] = "Sue";
   }

   public void displayNamesInUpperCase(){
      for(int i = 0; i < names.length; i++){
         if(names[i] != null){
            System.out.println(names[i].toUpperCase());
         }
      }
   }
}
```

Two-dimensional arrays

- Nested for loops are used to process each array element in a Two-Dimensional array.

```
public class Demo{
    private int[][] numbers = {{1,2,3},{4,5,6}};
    public Demo(){
        for(int row = 0;row<numbers.length;row++){
            // loop through each row
            for(int column = 0;column < numbers[row].length;column++){
                // loop through columns of current row
                System.out.print(numbers[row][column]+" ");
            }
            System.out.println();
        }
    }
}
```

while loop vs. for loop

- Any while loop of the following form

```
<initialization>;
while (<condition>) {
        <statement(s)> ;
        <update> ;
}
```

can be replaced by a for loop of the following form:

```
for (<initialization>; <condition>; <update>) {
        <statement(s)> ;
}
```

while loop vs. for loop

- The following while loop:

```
int i = 1;
while (i <= 10) {
    System.out.println( i );
    i++;
}
```

Can be replaced with:

```
for (int i = 1; i <= 10; i++) {
    System.out.println( i );
}
```

Questions

When would we use a while loop?

Typically when the condition is not related to an index.

- Perhaps you have a more complex condition (i.e., when this is true and that is true)
- When you want an infinite loop (i.e., user input handling)

When would we use a for loop?

Typically when you need to loop a fixed number of times (i.e., through all the elements in an array).

References

- Arrays in Java
  - https://www.tutorialspoint.com/java/java_arrays.htm (includes for loops with arrays)
  - https://www.javatpoint.com/array-in-java (more on arrays, including 2D arrays)

- For loop in Java
  - https://docs.oracle.com/javase/tutorial/java/nutsandbolts/for.html
  - https://www.geeksforgeeks.org/loops-in-java/ (loops in Java – while, for, foreach, do while)

Today's Lab (Lab 9)

- We are going to store information about Canada's provinces in a 2D array.

	Name Column 0	Capital Column 1	Largest City Column 2
Row 0	Alberta	Edmonton	Calgary
Row 1	British Columbia	Victoria	Vancouver
Row 2	Saskatchewan	Regina	Saskatoon
…	…	…	…

Why would we choose a 2D array over a Class?

Lab 9 and Next Week

Lab 9

- Due Wednesday, Nov. 21 at midnight
- Should be able to finish in class today

Next Week

- ArrayList
- For each loop
- This will be needed for Assignment 3