Revised 1.2.2A Today's Top 40

Otakar Andrysek

Subtarget 1.8: Create and use arrays
Subtarget 1.9: use a for-each loop to iterate through an array
I will look at 8, 10, 14, 17.

Introduction

In the last activity, 1.2.1 Parsing Text, you displayed a number of song titles, one at a time, using either a **for loop** or a **while loop**. Imagine that you wanted to *store* all of this information at one time in your program and that you had not 10 songs, but hundreds, perhaps thousands of songs. Maybe you do have hundreds or thousands of songs stored in a program somewhere! In this activity, you will learn a new construct to store many data items at once, and that construct is an **array**.

Materials

- Computer with BlueJ and Android™ Studio
- Android™ tablet and USB cable, or a device emulator

Activity

Part I: Make an Array

To start off this activity you will populate an array with a list of values.

1. [Learn about Arrays in Java](). Be sure to answer the **Check for understanding** questions at the end.

   Done.

2. When you write a statement such as

```
1 String[] names = {"Jamal", "Emily", "Destiny", "Mateo", "Sofia"};
```

   you are using an **initialization list**. In one step, you declare and initialize the String array names. This statement can be rewritten in other ways, too:

```
1 String[] names = {"Jamal",
2                   "Emily",
3                   "Destiny",
4                   "Mateo",
5                   "Sofia"  // notice no comma
6 };
```

   or

```
1 String[] names = {new String("Jamal"),
2                   new String("Emily"),
3                   new String("Destiny"),
4                   new String("Mateo"),
5                   new String("Sofia")
6 };
```

   Okay.

   Recall that the Java compiler ignores **whitespace** (the non-printable characters such as blanks, returns, or tabs) when it interprets code.

   Got it.

3. Open your BlueJ MediaLib project and create a new class ArrayMediaLib, with a main method similar to your other MediaLib classes.

   Done.

4. Using an initialization list, create a String array of at least five friends or family members with whom you would like to share your media items. Call the array sharingFriends.

   Lorem Ipsum.

Part II: Looping and Arrays

You can loop over the items in an array in much the same way you did with the characters in a string. In this part of the activity, you will explore looping over the items in an array and also what happens if you try to access an index in an array that does not exist.

5. Use a for loop to iterate over your friends' array using the following:

```
1  for (int i = 0; i < 5; i++)
2  {
3      System.out.println(sharingFriends[i]);
4  }
```

Done.

6. *Test your program.*

It works!

What do you think would happen if you had less than five people in your array? Find out.

7. Comment out one person from the initialization list using //. (If you comment the last element, be sure to delete the comma , on the line above it to indicate it is the last item in the initialization list.)

Done.

8. Run your program. What happens?

java.lang.ArrayIndexOutOfBoundsException: 4
at ArrayMediaLib.main(ArrayMediaLib.java:24)

The problem is called a **runtime error** because it happened while your program was running. It is also an **exception** because something out of the ordinary happened in your code. In this case, you tried to iterate beyond the length of your array causing an "array index out of bounds" exception.

9. At what line of code did that error occur?

Line 24.

10. To make your loop work with any number of friends, read **Using a For Loop to Loop Through an Array**. Then, modify your for loop to fix the problem. Test your program and record how you fixed it. (Paste a snip of your fixed loop.)

```
// Iterate over my friends' array
for (int i = 0; i < sharingFriends.length; i++)
{
    System.out.println(sharingFriends[i]);  // Print elements in order
}
```

Part III: Use a For-Each Loop

The for-each loop is a java construct that is very useful for iterating over the items in an array. Explore this construct in this part of the activity.

11. Because looping over arrays is such a common algorithm, Java provides a convenient construct called a **for-each** loop. **Learn about the For-Each Loop**, but do not read past the second paragraph of text, as these are advanced topics.

    Done.

12. In your media library, assume you have been tracking the number of days between song purchases. In ArrayMediaLib, create the following array:

```
1  int[] daysBtwnPurchase = {2, 5, 1, 2, 4, 2, 1, 3};
```

    Done.

13. Use an integer total and a for-each loop to calculate the average purchase time in days. Like you did at the end of Part II, make the algorithm versatile; do not hard code the number of data items.

    Done.

14. Display the result. (…and also paste a snip of your for-each loop)

```
// Calculate mean purchase time
for (int j = 0; j < daysBetweenPurchase.length; j++)     // i < length of array
{
    sumDays += daysBetweenPurchase[j];
}
float meanDays = sumDays / daysBetweenPurchase.length;
System.out.println(meanDays);  // Print elements in order
```

```
Tamir
Dane
Andrew
Stan
2.0
```

Part IV: Arrays of Objects

In addition to integers, doubles, and strings, you can create arrays of other types of data, including objects.

15. In ArrayMediaLib, declare and initialize an array of Song objects:

```
Song[] topTenSongs = {new Song(),
                      new Song(),
                      new Song()
                      };
```

Compare this syntax with syntax you used to create your array of friends. Notice, in particular, the Song[] syntax in place of String[]. This indicates the data type of the array. For topTenSongs, the data type is Song.
You *could* use 10 calls to setTitle(…) to set all of your song titles, but an initialization list requires less code.

Okay.

16. Modify your array of songs to declare and initialize 10 Song objects. Use the top 10 songs in **Billboard's top 100 Rock-n-Roll songs of all time**. You will not have a price or rating for these songs, so create and use a new Song constructor with only a title as its parameter.

Okay.

17. Display the song titles using a for-each loop.

```
1  for (Song s: topTenSongs)
2  {
3      System.out.println(s.getTitle());
4  }
```

In the body of the for loop, notice s.getTitle(). Since s is a Song, you can access all of the public methods with s. syntax.

Paste a snip of your output.

```
Tamir
Dane
Andrew
Stan
2.0
THE TWIST
SMOOTH
MACK THE KNIFE
HOW DO I LIVE
PARTY ROCK ANTHEM
I GOTTA FEELING
MACARENA (BAYSIDE BOYS MIX)
PHYSICAL
YOU LIGHT UP MY LIFE
HEY JUDE
```