Lesson 69: Arrays Intro (W23D1)

Balboa High School

Michael Ferraro

February 1, 2016

Do Now

While your teacher pulls updates from all students' GitHub repositories and assembles all *FirstnameLastname*`.java` files into a single JAR file, read through `InterfacePeople/src/PeopleDriver.java`'s `main()` method, answering the following.

- The `ArrayList` holding all students' objects holds what kind of object?
- What keyword is being used to determine what class a given object belongs to?
- Can you find where casting is being used on objects? (You've only seen casting on primitives thus far.)

next slide →

Do Now

Your teacher is using shell commands to combine all students' class sources. Note the use of a `for()` loop – which is syntactically different from Java:

```
# pull all repos...

cd ~mferraro/apcs/git/[5|6]
mkdir tmp

for USER in ‘cat ~/apcs/apcs-users-[5|6].txt‘; do
  cp -f $USER/IntefacePeople/*.java  tmp
done

cd tmp

rm -f Comedian.java Dancer.java Gamer.java Idealist.java Person.java \
      MichaelFerraro.java PeopleDriver.java Pessimist.java Singer.java

jar cMf People-[5|6]th.jar *.java

# copy to web root on apcs02, students visit http://apcs02/people/
```

Aim

Students will begin learning about one-dimensional Java arrays.

Java Arrays

Arrays are data structures

Java Arrays

Arrays are data structures

- blocks of consecutive memory locations

Java Arrays

Arrays are data structures

- blocks of consecutive memory locations
- hold primitive values or references to objects

Java Arrays

Arrays are data structures

- blocks of consecutive memory locations
- hold primitive values or references to objects
- are indexed just like `ArrayLists`: an array with n elements has indices $0 \ldots n-1$

Java Arrays

Arrays are data structures

- blocks of consecutive memory locations
- hold primitive values or references to objects
- are indexed just like `ArrayLists`: an array with n elements has indices $0 \ldots n - 1$
- size isn't automatically managed for us like with `ArrayLists`; programmer manages size!

First Array Example

Here is an array of `int`s.

```
//will hold up to 5 ints:
int[] intArr1 = new int[5];

intArr1[0] = 0;
intArr1[1] = 2;
intArr1[2] = 4;
intArr1[3] = 6;
intArr1[4] = 8;
```

Note the use of Java keyword `new`. Arrays aren't pure Java objects, but this is one similarity to objects. Also note that `new` is being called without a constructor afterward!

An Array Walk Using for()

for() can be used to traverse the elements of an array.

```
int[] intArr1 = new int[5];

intArr1[0] = 0;
intArr1[1] = 2;
intArr1[2] = 4;
intArr1[3] = 6;
intArr1[4] = 8;

for ( int i = 0 ; i < 5 ; i++ ) {
    System.out.println( intArr1[i] );
}
```

Populating an Array

- Create a new project called `L69`.
- Write a program that uses a `for()` loop to populate an array of length 5 with the values $\{0, 2, 4, 6, 8\}$, just as in `intArr1`.
- Use another `for()` loop to print out the stored elements.

Array Length

- **Q:** What if you don't know the length of an array when writing a `for()` loop?

Array Length

- **Q:** What if you don't know the length of an array when writing a `for()` loop?
- **A:** Arrays, while not Java objects per se, provide a field called `length`, available via dot notation.

Array Length

- **Q:** What if you don't know the length of an array when writing a `for()` loop?
- **A:** Arrays, while not Java objects per se, provide a field called `length`, available via dot notation.
- `for(int i = 0; i < ???; i++)`

Array Length

- **Q:** What if you don't know the length of an array when writing a `for()` loop?
- **A:** Arrays, while not Java objects per se, provide a field called `length`, available via dot notation.
- `for(int i = 0; i < ???; i++)`
- `for(int i = 0; i < intArr1.length; i++)`

Array Length

- **Q:** What if you don't know the length of an array when writing a `for()` loop?
- **A:** Arrays, while not Java objects per se, provide a field called `length`, available via dot notation.
- `for(int i = 0; i < ???; i++)`
- `for(int i = 0; i < intArr1.length; i++)`
- `length` is not a method, but it's comparable to `ArrayList`'s `size()` method.

Array Length

- **Q:** What if you don't know the length of an array when writing a `for()` loop?
- **A:** Arrays, while not Java objects per se, provide a field called `length`, available via dot notation.
- `for(int i = 0; i < ???; i++)`
- `for(int i = 0; i < intArr1.length; i++)`
- `length` is not a method, but it's comparable to `ArrayList`'s `size()` method.
- **modify your `for()` loops to use the `length` field variable.**

Arrays Can Hold Objects, Too

Example #1:

```
String[] words = new String[3];

words[0] = "abc";
words[1] = "def";
words[2] = "ghi";
```

Arrays Can Hold Objects, Too

Example #2:

```
    String[] words = new String[3];

    words[0] = "abc";
    words[1] = "def";
    words[2] = words[1];
```

What is being held at `words[2]`? A **reference** to the same object pointed to by `words[1]`.

What happens if that object is *modified*?

Declare and Initialize in One Step

```
int[] intArr1 = new int[5];

intArr1[0] = 0;
intArr1[1] = 2;
intArr1[2] = 4;
intArr1[3] = 6;
intArr1[4] = 8;
```

Declare and Initialize in One Step

```
//int[] intArr1 = new int[5];

int[] intArr1 = { 0, 2, 4, 6, 8 };
```

HW

- Start working on PS #12.
- Finish §§1–3, inclusive.
- Find the required reading here.