Java Arrays

CS 1400 - 2022
Cody Henrichsen

1

Java Arrays

- The fixed size data structure
- No Import needed
- Only data structure that can hold either primitives or objects

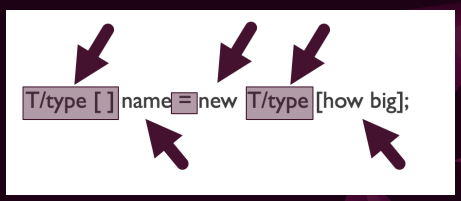
2

Array Vocabulary

- [ ]
  - The "keyword" in Java for an array
  - The only way to access the contents of an array, especially for replacement
  - Subscript notation is how you access array contents as well
- Index
  - Where you are in the array
  - Valid ranges from 0 … arrayName.length – 1
- .length
  - How many slots in the array
- ArrayIndexOutOfBoundsException
  - The error thrown if you access an index < 0 or >= arrayName.length

3

Array Syntax



4

Why so hipster?

- An array is an Object with NO constructors, or methods of its own
  - Not a single use of ()!!!
- An array has a public data member!
- Array are the only use of the [ ] symbols in Java
- No keyword
- Fixed size
- Holdover from C/C++

5

Arrays: Replace not remove

- Since it is fixed in size you can only replace values with another value of the same t/Type
- Place the arrayName[index] on the left hand side of the assignment operator to assign/replace the current contents at that index
- In the case of primitive value arrays the default value is the default for the data type so you may want to use a better value to check
- Object based arrays contain null as the default value

6

Arrays: Retrieval

- Place the arrayName[index] on the right hand side of the assignment operator to retrieve the value at the current index
- Retrieving a primitive value gives you a COPY of the value stored at that index
- Strings are immutable, so if you change a String it is a new object
- Retrieving any other object gives you a reference to the object.
  - So if you modify its internal state it will change

7

Arrays: Take a look inside

- Contents are most often accessed via loop
- The index parameter must meet BOTH of these conditions
  - Index >= 0
  - Index < arrayName.length
- Use a standard for loop for any replacement based operations or algorithms
- A for-each loop is best when accessing regardless of position or providing output
  - Reminder for-each has no access to index
  - For-each does not allow replacement

8

Reminders

- Java Strings are NOT an Array
  - No use of subscript [ ] notation works in Java Strings
- ArrayLists are also NOT an Array
  - No subscript [ ] notation
  - Methods rock 😸
  - Dynamically sized

9