Computer Science 101 S1

Lecture 19

Contents

Arrays of objects: declare, create, assign to an array of object type.

Processing each element in an array of object type.

Coursebook: pages §15.3 - §15.7

Ex01 - Complete `getAverage()`

This method is passed two parameters, an `int[] array`, and an `int` indicating the number of elements which are to be used to calculate the average. The method returns a `double`, the average of the given number of elements in the array.

```
public class ProgramClassL19 {
 public void start() {
   int[] array = { 3, 5, 7, 3, 2, 7, 9, 12, 14, 7 };
   double average = getAverage(array, 4);
   System.out.println("Average of first 4 items: " + average);
   average = getAverage(array, 7);
   System.out.println("Average of first 7 items: " + average);
 }
 private              getAverage(             ) {
```

Ex01 - Complete `getAverage()`

```
 }
}
```

The output with the completed method is shown below:

```
> java L19
 Average of first 4 items: 4.5
 Average of first 7 items: 5.142857142857143
```


Declaring arrays of objects

An example declaring two arrays of primitives:

```
int[] assign2Marks;
boolean[] isSingle;
```

An example declaring two arrays of objects:

```
String[] surnames;
ForSale[] items;
```

Note: I have assumed here that the `ForSale` class has been defined.

Arrays - create and initialise

Just as we saw with arrays of primitives

```
int[] assign2Marks = { 97, 86, 74, 99, 100, 76, 55 };
boolean[] isSingle = { true, true, true, false, true, false };
```

we can use the `{}` syntax to declare and initialise an array of objects in one statement, for example:

```
String[] carMakes = {"Toyota", "Nissan", "BMW","Honda",
                                "Saab", "Mazda", "Ferrari"} ;
String[] akeWords = {"bake", "brake", "cake", "flake",
                          "lake", "make", "rake", "stake",
                                          "take", "wake" } ;
ForSale[] items = {new ForSale(114, "021-3245678", 155,
          "Dior dress, 1972, Rarely worn, cost $35000"),
                    new ForSale(115, "021-3288678", 265,
                      "Avanti Bicycle, 2000ks, near new")
                  };
```

7

Creating the array space

Just as we saw with arrays of primitives,

```
int[] assign2Marks = new int[15];
boolean[] isSingle = new boolean[25];
```

before we can store values in an array, we need to create the array space:

```
String[] carMakes = new String[3482];
String[] akeWords = new String[10];
ForSale[] items = new ForSale[500];
```

8

Storing objects in an array

Once the array has been created, values of the array type can be assigned to each element of the array.

```
int[] assign2Marks;
assign2Marks = new int[15];
boolean[] isSingle;
isSingle = new boolean[25];
```

```
assign2Marks[0] = 94;
isSingle[0] = true;
assign2Marks[1] = 87;
isSingle[1] = false;
```

```
String[] carMakes = new String[3482];
String[] akeWords = new String[10];
ForSale[] items = new ForSale[500];
carMakes[0] = "Toyota";
akeWords[4] = "make";
akeWords[1] = "overtake";
items[0] = new ForSale(114, "021-3245678", 155,
          "Dior dress, 1972, Rarely worn, cost $35000");
items[1] = new ForSale(115, "021-3288678", 265,
                    "Avanti Bicycle, 2000ks, near new");
```

9

Array elements are variables

Each element in an array is a variable of the array type e.g. if the array is an array of type `int` then each element in the array is an `int` variable, if the array is an array of type String then each element in the array is a String variable.

```
int[] assign2Marks = { 97, 86, 74, 99, 100, 76, 55 };
assign2Marks[2] = assign2Marks[2] + 10;
int pos = assign2Marks.length - 1;
assign2Marks[pos] = (int) (assign2Marks[2] * 1.5);
int value = 100 % assign2Marks[0];
```

```
String[] akeWords = { "bake", "brake", "cake", "flake",
      "lake", "make", "rake", "stake", "take", "wake"  };
String word1 = akeWords[4].toUpperCase();
akeWords[7] = akeWords[7].substring(1, 3);
char c = akeWords[5].charAt(0);
System.out.println(akeWords[3] + " & " + akeWords[2]);
```

10

Array size

The public instance variable, `length`, is used to obtain the size of the array:

```
int[] assign2Marks = { 97, 86, 74, 99, 100, 76, 55 };
int size = assign2Marks.length;
```

```
String[] months = {"Jan", "Feb", "Mar","Apr","May","Jun",
                   "July", "Aug", "Sep", "Oct", "Nov", "Dec"};
int size = months.length;
```

What value is assigned to the `int` variable, `length`, below?

```
String[] months = {"Jan", "Feb", "Mar","Apr","May","Jun",
                   "July", "Aug", "Sep", "Oct", "Nov", "Dec"};
int length = months[3].length();
```

11

Processingelementsof an array

Each element in an array can be processed using a loop.

```
int[] assign2Marks = { 97, 86, 74, 99, 78, 76, 55, 86,
                       74, 99, 78, 86, 74, 99, 91 };
for(int i=0; i<assign2Marks.length; i++) {
     assign2Marks[i] = assign2Marks[i]/2;
}
```

```
String[] akeWords = { "bake", "brake", "cake", "lake",
         "flake","make","rake","stake", "take", "wake"};
for(int i=0; i<akeWords.length; i++) {
      akeWords[i] = akeWords[i] + "s";
}
```

12

Processingelementsof an array

Sometimes we create a large array and only fill some of the elements in the array. In this case the size of the array is different to the number of elements in the array:

```
String[] phobias = new String[450];
int wordsSoFar = 0;
phobias[wordsSoFar] = "cyberphobia";
wordsSoFar++;
phobias[wordsSoFar] = "cynophobia";
wordsSoFar++;
phobias[wordsSoFar] = "astraphobia";
wordsSoFar++;
phobias[wordsSoFar] = "ergophobia";
wordsSoFar++;
for(int i=0; i<wordsSoFar; i++) {
     phobias[i] = phobias[i].toUpperCase();
}
```

13

Arrayelements- default value

For arrays of an object type, each element has the value, `null`, unless an object has been assigned to it e.g.

```
String[] phobias = new String[5];
int wordsSoFar = 0;
phobias[wordsSoFar] = "cyberphobia";
wordsSoFar++;
phobias[wordsSoFar] = "cynophobia";
wordsSoFar++;
for(int i=0; i< phobias.length; i++) {
       System.out.print(phobias[i] + " ");
}
```

prints the following:

```
cyberphobia  cynophobia null null null
```

14

Processingelementsof an array

Each element in an array can be processed using a loop.

```
int[] assign2Marks = { 97, 86, 74, 99, 78, 76, 55, 86,
                       74, 99, 78, 86, 74, 99, 91 };
 int mark;
 int highestSoFar = assign2Marks[0];
 for(int i=1; i<assign2Marks.length; i++) {
       mark = assign2Marks[i];
       if (mark > highestSoFar) {
                 highestSoFar = mark;
       }
 }
 System.out.println("Highest mark: " + highestSoFar);
```

15

Ex02 - Complete the code

Complete the code so that the shortest String in the array is printed. Assume the array contains at least one element.

```
String[] phobias = { "myophobia", "mysophobia",
             "nebulaphobia","necrophobia", "negrophobia",
              "neophobia", "nosophobia", "novercaphobia",
              "nyctophobia", "ochlophobia", "oenophobia",
        "ombrophobia", "onomatophobia", "ophidiophobia",
      "ophthalmophobia", "optophobia", "ornithophobia"};
 String shortest = phobias[0];
 for (int i=     ; i<                        ; i   ){


 }
 System.out.println("Shortest phobia: " + shortest);
```

16

The `ForSale` class

```
public class ForSale {
      private int iDNumber;
      private String description, contact;
      private int askingPrice;
      private boolean isSold;
      public ForSale(int iD, String phone, int price,
                                          String info) {
            this.iDNumber = iD;
            this.description = info;
            this.askingPrice = price;
            this.contact = phone;
            this.isSold = false;
      }
      public boolean getIsSold() {
            return isSold;
      }
```

17

The `ForSale` class

```
 public int getAskingPrice() {
       return askingPrice;
 }
 public void setAskingPrice(int price) {
       askingPrice = price;
 }
 public int getIDNumber() {
       return iDNumber;
 }
 public String toString() {
    String info = "Article: " + iDNumber;
    info = info + " " + description + ",";
    info = info  + "\n  Contact: " + contact;
    info = info + ", $" + askingPrice;
    if (isSold) {
       info = info + " SOLD";
    }
    return info;
}}
```

18

Storing `ForSale` items

```
public class Program {
 public void start() {
  int idNumber = 4456;
  int itemsSoFar = 0;
  ForSale[] items;
  items = new ForSale[500];
  items[itemsSoFar] = new ForSale(idNumber, "021-4512349",
                                  196 ,"Number plate: 2Fast:4U");
  itemsSoFar++;
  idNumber++;

  items[itemsSoFar] = new ForSale(idNumber, "021-4566749",
                                  457, "Salvador Dali teaspoon");
  itemsSoFar++;
  idNumber++;
  items[itemsSoFar] = new ForSale(idNumber, "027-4512555",
                    54, "T-Shirt - Those who cannot remember
                      the past are condemned to repeat it");
  itemsSoFar++;
  idNumber++;

}}
```

19

Ex03, Ex04, Ex05

In the following three exercises assume that the array **items** has been created and partially filled. Assume that the number of **ForSale** items stored in the **items** array is given by the parameter, **howMany**.

20

Ex03 - displayItems()

Complete the **displayItems()** method which prints all the items in the **ForSale** array:

```
private void displayItems(ForSale[] items, int howMany){
   ForSale item;




}
```

21

Ex04 - getHighestPrice()

Complete the **getHighestPrice()** method which returns the highest **askingPrice** of all the items in the **ForSale** array:

```
private int getHighestPrice(ForSale[] items, int howMany) {
   ForSale item;
   int highestSoFar;




}
```

22

Ex05 - getHighestPriceItem()

Complete the **getHighestPriceItem()** method which returns the **ForSale** item with the highest **askingPrice**:

```
private ForSale getHighestPriceItem(ForSale[] items,int howMany){
   ForSale item;
   int highestSoFar;




}
```

```
months = new String[12];
```

23

Ex06 - Why do I get an error?

```
public class Program {
 public void start() {
     String[] months;
     months[1] = "January";
     months[2] = "February";
 }
}
```

```
System.out.println(months[0].length());
```

24

Ex07 - Why do I get an error?

```
public class Program {
 public void start() {
     String[] months = new String[12];
     months[0] = "January";
     months[1] = "February";
     System.out.println(months[0].length);
 }
}
```

25

```
for (int i=0; i<num ; i++) {
```

Ex08 - Why do I get an error?

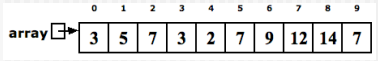
```
public class Program {
 public void start() {
       ForSale[] items = new ForSale[500];
       int num = 0;
       items[num] = new ForSale(numberSoFar , "021 4567189",
          140, "Armani 1960 Wedding dress - never been worn");
       num++;

       items[num] = new ForSale(numberSoFar , "021 7654345",
                    155, "Avanti Bicycle, 2000ks, near new");
       num++;

       for (int i=0; i<=num ; i++) {
             System.out.println(items[i].toString());
       }
 }
}
```

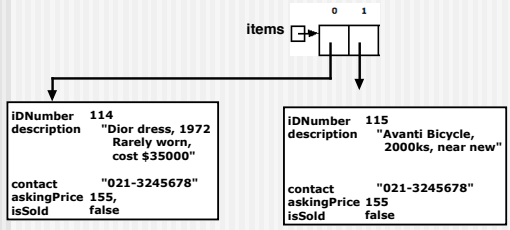
26

Visualising the array - slide 3



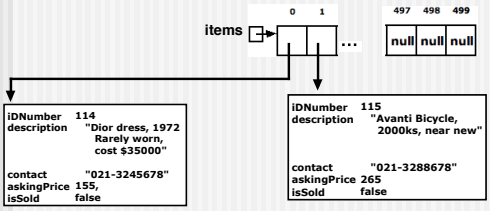
27

Visualising the array - slide 6



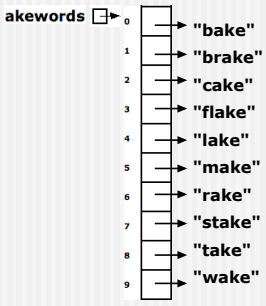
28

The `items` array - slide 8



29

the `akewords` array - slide 9



30

What you need to know

- How to declare an array of an object type.
- How to create the array space.
- How to fill the array with objects.
- How to visualise the array.
- How to process the objects in the array.