

OOP Using JAVA

07 – Arrays

Dr. Mostafa A. El-Hosseini

melhosseini@eng.kfs.edu.eg

www.melhosseini.spaces.live.com

Recap

- » For loop
- » While loop
- » Do-while loop
- » Nested loops
- » Examples
- » Break
- » Continue

Agenda

- » Arrays
- » One-Dimensional Arrays
- » Multidimensional Arrays
- » String class
- » Example

Arrays

- » An Array is a **group of like-typed variables** that are referred to by a **common name**
- » Arrays of any type can be created and may **have one or more dimensions**
- » A specific element in an array is **accessed by its index**

One-Dimensional Arrays

- » Is a **list of like-typed** variables
- » To create any array, you first must create an array of the desired type, general form of array declaration is
`type var-name[ ];`
`int month_days[ ];`
- » Although this declaration establishes the fact that `month_days` is an array variable, **no array actually exist**
- » In fact, the value of **`month_days`** is set to **null**, which represents an array with no value
- » To link **`month_days`** with an actual, physical array of integers, you must allocate one using **new**

new

- `Array_var = new type[size]`
- `Month_days = new int [12]`
 - » After this statements executes, **month_days** will refer to an array of **12** integers. Further, all elements in the array will be initialized to zero
 - » All array indexes **start at zero**
 - » It is possible to combine the declaration of the array variable with the allocation of the array itself
- `Int age[] = new int [10]`

Hint:

- » Arrays can be initialized when they are declared
- `int x [] = {31,28,30,20,11};`

Example - Average

```
class myaver
{
    public static void main (String args[])
    {
        double nums[] = {10.1, 11.2, 12.3, 13.4, 14.5};
        double result = 0;
        for(int i = 0; i<5;i++)
            result +=nums[i];
        System.out.println("Average is " + result / 5);
    }
}
```

Example

```
D:\>javac myaver.java  
D:\>java myaver  
Average is 12.299999999999999  
D:\>
```

Multidimensional Arrays

- `int twoD [] [] = new int [4][5];`
- This allocates a 4 by 5 array and assigns it to `twoD`
- When you allocate memory for a multidimensional array, you need only specify the memory for the first(leftmost) dimension
- You can allocate the remaining dimensions separately .

Second dimension are unequal ...

- `int twoD [] [] = new int [4] [];`
- `twoD [0] = new int [5];`
- `twoD [1] = new int [5];`
- `twoD [2] = new int [5];`
- `twoD [3] = new int [5];`
- » You do not need to allocate the same number of elements for each dimension
- `int twoD [] [] = new int [4] [];`
- `twoD [0] = new int [1];`
- `twoD [1] = new int [2];`
- `twoD [2] = new int [3];`
- `twoD [3] = new int [4];`

Hint:

- » It is possible to initialize multidimensional arrays
- » `double m[][] = {
 {0, 1, 2, 3 },
 {20,40,30,50},
 {11,13,14,15}
};`

Alternative Array Declaration

- `type[]` var-name;
- `int al[] = new int [3];`
- `Int[] a2 = new int [3];`
- `char twod1 [] [] = new char [3][4];`
- `char [][] twod2 = new char [3][4];`

Strings

- » Java's string type, called **String** is not a simple type. Nor is it simply an array of characters .
- » Rather, **String** defines an object
- » **String** type is used to declare string variables
- **String str = "this is a test";**
- **System.out.println(str);**
- » **str** is object of type string

Homework

- Write down a program to read in 2-D matrix, and then calculate the sum of each row, sum of each column , and the sum of diagonal numbers