

Chapter 4. Java methods, arrays and references

I. Exercises with solutions

1. Write an OOP program including a class that describes a point in the 3D space (each object of this class should have three variables: x, y and z). The class should have the following instance methods: a method that calculates the distance between the current point (on which the method is invoked) to another point; a method that calculates the distance between the current point (on which the method is invoked) to the center. a method that prints all the details of the point.

```
public class Point3D
{
    private double x,y,z;
    public Point3D(double x, double y, double z)
    {
        this.x = x;
        this.y = y;
        this.z = z;
    }
    public double distanceFromCenter()
    {
        return distanceFrom(0,0,0);
    }
    public double distanceFrom(Point3D other)
    {
        return Math.sqrt(Math.pow(x-other.x,2)+ Math.pow(y-
other.y,2)
                        + Math.pow(z-other.z,2));
    }
    public double distanceFrom(double xVal, double yVal,
double zVal)
    {
        return Math.sqrt(Math.pow(x-xVal,2)+
                        Math.pow(y-yVal,2)+
                        Math.pow(z-zVal,2));
    }
    public void details()
    {
        System.out.println("x="+x+ "y="+y+" z="+z);
    }
    public static void main(String args[])
    {
        Point3D point1, point2;
        point1 = new Point3D(2,3,4);
        point2 = new Point3D(3,7,8);
    }
}
```

```

System.out.print("point1:");
point1.details();
System.out.print("point2:");
point2.details();
System.out.println("The distance between point1 and
(0,0,0) is : " +
                    point1.distanceFromCenter());
System.out.println("The distance between point1 and
point2 is : " +
                    point1.distanceFrom(point2));
    }
}

```

2. Write an OOP program sorting an array of integers in ascending order.

Solution 1:

```

public class ArraySort {
    public static void main(String[] args) {
        /*Tạo và khởi tạo giá trị cho mảng*/
        int[] A = {5, 7, 2, 4, 8};

        /*Sắp xếp mảng theo chiều tăng dần*/
        for (int i=0;i<A.length-1;i++)
            for (int j=i+1;j<A.length;j++)
                if (A[i]>A[j])
                {
                    int t =A[i];
                    A[i]=A[j];
                    A[j]=t;
                }

        /*In mảng ra màn hình*/
        for (int i=0;i<A.length;i++)
            System.out.print(A[i]+" ");
    }
}

```

Solution 2.

```

/* Use class Arrays in java.util*/
import java.util.Arrays;

public class ArraySort {

    public static void main(String[] args) {

        /*Tạo và khởi tạo giá trị cho mảng*/

```

```
int[] A = {5, 7, 2, 4, 8};

/*Sử dụng hàm sort của Lớp Arrays để sắp xếp mảng A theo chiều tăng dần*/
Arrays.sort(A);

/*Hiển thị kết quả ra màn hình*/
for (int i=0;i<A.length;i++)
    System.out.print(A[i]+" ");
}
}
```



```
<terminated> ArraysSort [Java Application] C:\
2 4 5 7 8
```

3. Write an application that creates a two dimensions array 10 X 10 and stores the multiplication table in it. The application should also prints the multiplication table values.

```
public class TwoDemenArray
{
    public static void main(String args[])
    {
        int matrix[][];
        matrix = new int[10][10];
        for(int i=0;i<10;i++)
        {
            for(int j=0;j<10;j++)
            {
                matrix[i][j] = (i+1)*(j+1);
            }
        }
        for(int i=0;i<10;i++)
        {
            for(int j=0;j<10;j++)
            {
                System.out.print(matrix[i][j]+"\\t");
            }
            System.out.println();
        }
    }
}
```

4. Write an OOP program in which includes a factorial method.

Calculating $S = 7! + 10! + 12! + 14!$

```
public class Factorial {  
  
    /*Định nghĩa hàm tính giai thừa*/  
    public static long fact(int n)  
    {  
  
        /*Tính n giai thừa*/  
        int kq =1;  
        for (int i=2;i<=n;i++)  
            kq *=i;  
        return kq;  
    }  
  
    public static void main(String[] args) {  
  
        /*Gọi hàm fact, do hàm fact là static nên không cần tạo đối tượng để gọi hàm*/  
        long S = fact(7) + fact(10) + fact(12) + fact(14);  
  
        //In kết quả ra màn hình  
        System.out.println("Sum =" +S);  
  
    }  
  
}
```

```
<terminated> Factorial [Java Application] C:\Program Files\Java\jdk1.7.0_45\jre\bin\javaw.exe (Nov 18, 2013, 11:02:17 AM)  
Sum =1761580720
```

5. Write an OOP program to find the greatest common divisor of two integers entered from the keyboard.

```
import java.util.Scanner;  
public class CommonDivisor {  
  
    /*Định nghĩa hàm tính ước số chung Lớn nhất*/  
    public static int comDiv(int a,int b)  
    {  
        return (b==0)?a:comDiv(b, a%b);  
    }  
    public static void main(String[] args)
```

```

    {
        int a;
        int b;

        /*nhập dữ liệu từ bàn phím*/
        Scanner keyboard = new Scanner(System.in);
        System.out.print("a = ");
        a = keyboard.nextInt();

        System.out.print("b = ");
        b = keyboard.nextInt();

        /*Gọi hàm và in kết quả ra màn hình*/
        System.out.println("USCLN của "+a+" va "+b+" la
        "+comDiv(a,b));
    }
}

```

```

<terminated> CommonDivisor [Java Application] C:\Program Files\Java\jdk1
a = 16
b = 10
USCLN của 16 va 10 la 2

```

6. Write an OOP program for processing a give string:
- Count the frequency of character *a* in the string
 - Check if the string contains the word “Java”?
 - Check if the string starts with the word “Write”?
 - Check if the string ends with the word “easily”

```

public class StringProcessing {

    public static void main(String[] args) {
        String s ="Write a Java program very easily";

        /*Dùng để chứa số lượng ký tự 'a' có trong chuỗi*/
        int count=0;

        /* Đếm số ký tự 'a'*/
        for (int i=0;i<s.length();i++)
            if (s.charAt(i)=='a') count++;
        System.out.println("so luong ky tu a co trong chuoai =" +count);

        /*Kiểm tra chuỗi có chứa từ “Java” hay không*/
        int index =s.indexOf("Java");
    }
}

```

```

        if (index>=0) System.out.println("Tu Java dau tien cua chuoì o
vi tri "+index);
        else System.out.println("Chuoì khong chua tu Java");

/*Kiểm tra chuỗi có bắt đầu bằng cụm từ "Write" */
        if (s.startsWith("Write")==true) System.out.println("Tu Write
bat dau chuoì");
        else System.out.println("Tu Write khong bat dau chuoì");

/*Kiểm tra chuỗi có kết thúc bằng từ "easily" */
        if (s.endsWith("easily")==true) System.out.println("Tu easily
ket thuc chuoì");
        else System.out.println("Tu easily khong ket thuc chuoì");
    }
}

```

The screenshot shows a console window with the following output:

```

<terminated> StringProcessing [Java Application] C:\Program Files\Java\jdk1.7.0_45\jre\bin\javaw.exe (Nov 18, 2013, 11:18:57 AM)
so luong ky tu a co trong chuoì =5
Tu Java dau tien cua chuoì o vi tri 8
Tu Write bat dau chuoì
Tu easily ket thuc chuoì

```

7. Write an OOP program, using the Vector class in java.util package to store fruits. Display its size, capacity and elements.

```

import java.util.*;
public class Reference {
    public static void main(String args[]) {
        /* Vector of initial capacity(size) of 6 */
        Vector<String> vec = new Vector<String>(6);

        /* Adding elements to a vector*/
        vec.addElement("Banana");
        vec.addElement("Watermelon");
        vec.addElement("Mango");
        vec.addElement("Apple");

        /* check size and capacityIncrement*/
        System.out.println("Size is: "+vec.size());
        System.out.println("Default capacity increment is:
"+vec.capacity());

        vec.addElement("Grape");
        vec.addElement("Orange");
        vec.addElement("Clipart");
    }
}

```

```

        /*size and capacityIncrement after two insertions*/
        System.out.println("Size after addition: "+vec.size());
        System.out.println("Capacity after increment is:
"+vec.capacity());

        /*Display Vector elements*/
        Enumeration en = vec.elements();
        System.out.println("\nElements are:");
        while(en.hasMoreElements())
            System.out.print(en.nextElement() + " ");
    }
}

```

II. Do it yourself

1. Write an OOP program with a method to find the smallest number among three numbers.
2. Write an OOP program with a method to count all vowels in a string.
3. Write an OOP program with a method to count all words in a string.
4. Write an OOP program with a method to compute the sum of the digits in an integer.
5. Write an OOP application that stores the factorials of all the numbers between 1 (included) and 10 (included) in an array and prints them in a reverse order.
6. Write an OOP program sorting an array of integers in descending order.
7. Write an OOP program to check if the first and the last element of an array of integers are same.

Hint. The length of the array must be greater than or equal to 2.

8. Write an OOP program to swap the first and last elements of an array (length must be at least 1) and create a new array.
9. Write an OOP program to multiply corresponding elements of two arrays of integers

Hint. Sample Output:

Array1: [2, 3, 5, 3]

Array2: [2, 4, -5, -2]

Result: 4 12 -25 -6

10. Write an OOP program to check if a string starts with a specified word entered from the keyboard.
11. Write an OOP program to create a new array from a given array of integers (two array have the same size).

12. Write an OOP program to find the maximum and minimum value of an array.
13. Write an OOP program to reverse an array of integer values.
14. Write an OOP program to find the common elements between two arrays (string values)

Hint.

```
Array1 : [Python, JAVA, PHP, C#, C++, SQL]
Array2 : [MySQL, SQL, SQLite, Oracle, PostgreSQL, DB2, JAVA]
Common element is : [JAVA, SQL]
```

15. Write an OOP program with a method to displays an 8-by-8 matrix.
16. Write an OOP program to convert an array to ArrayList.
17. Write an OOP program to compare two strings lexicographically. Two strings are lexicographically equal if they are the same length and contain the same characters in the same positions.
18. Write an OOP program to concatenate a given string to the end of another string
19. Write an OOP program to get a substring of a given string between two specified positions.
20. Write an OOP program to check if a given string contains the specified substring.
21. Write an OOP program to create a new String object with the contents of a character array.
22. Write a OOP program to check whether two String objects contain the same data.
23. Write an OOP program to count the occurrences of a given string in another given string.

Hint.

Sample output: 'aa' has occured 3 times in 'abcd abc aabc baa abcaa'

24. Write an OOP method to check whether a string is a valid password.
 - Password rules:
 - A password must have at least ten characters.
 - A password consists of only letters and digits.
 - A password must contain at least two digits.
25. Write an OOP program, using the Hashtable class in java.util package to store a list of ids and studentnames. Display its elements information.

Hint.

```
Import java.util.*;
...
Hashtable list = new Hashtable();
list.put("VKU01", "Nguyen Van A");
list.put("VKU02", "Huynh Van B");
```

Enumeration elements = list.elements();

.....