題目01名稱：**The 3D Points**

1. 題目描述(程式範例描述)：

Design a class named Point3D to represent a 3 dimensional point which contains the following attributes and services.

Fields

(1) An int-type class field named noOfPoints represents the total number of points having been created. It’s initial value is 0 and it is increased by 1 after a point is added.

(2) A double-type instance field named x represents the x coordinate of the point.

(3) A double-type instance field named y represents the y coordinate of the point.

(4) A double-type instance field named z represents the z coordinate of the point.

Constructor

(1) A default constructor which will initialize all the instance fields as 0 and increase noOfPoint by 1.

(2) A constructor consists of 3 parameters corresponding to the 3 fields listed above (numbered as 2, 3, and 4). The first parameter is for x, the 2nd for y, and so on. The constructor initializes all the instance fields and also increases noOfPoint by 1.

Methods

(1) Provide the getters for all the fields listed above, including the class field.

Note 1: The name of a getter must be of the form getXxx where Xxx is the field name. For example, the getter for x is getX.

(2) Provide the setters for all the fields listed above.

Note 2: The name of a setter must be of the form setXxx where Xxx is the field name. For example, the setter for x is setX.

(3) Provide a public instance method which returns a string containing the information of a point with the form as shown in the testing output.

(4) Provide a public instance method distanceFrom which takes a point q as the parameter and returns the Euclidian distance from the point to q.

The Point3D class will be tested by the following driven class TestPoint3D.

public class TestPoint3D {

public static void main(String[] args) {

Point3D pt1, pt2;

pt1 = new Point3D(3.0, 4.0, 5.0);

pt2 = new Point3D(0.0, 0.0, 10.0);

System.out.println(pt1);

System.out.println(pt2);

System.out.println(pt1.distanceFrom(pt2));

System.out.println(Point3D.getNoOfPoints());

}

}

1. 輸入說明：
2. 輸出說明：

(3.0,4.0,5.0)

(0.0,0.0,10.0)

7.0710678118654755

2

1. 裝備限詞

|  |  |
| --- | --- |
| 出現次數 | 語句 |
|  |  |

1. 輸入範例：(2組公開測資、1組暗資)

|  |  |
| --- | --- |
| 1 |  |

1. 輸出範例：

|  |  |
| --- | --- |
| 1 | (3.0,4.0,5.0)  (0.0,0.0,10.0)  7.0710678118654755  2 |

1. 程式碼解答(或是直接附.java檔也可以)：

TestPoint3D.java

題目02名稱：**Sum of Diagonal Elements of a Square Matrix**

1. 題目描述(程式範例描述)：

Your task is to write a program which returns the sum of diagonal elements of a square matrix.

1. 輸入說明：

The first line of input contains a single positive integer N. This is the size of the matrix. Each of the next N lines will contain N integers.

1. 輸出說明：

The sum of diagonal elements of the N\*N matrix

1. 裝備限詞

|  |  |
| --- | --- |
| 出現次數 | 語句 |
|  |  |

1. 輸入範例：(2組公開測資、1組暗資)

|  |  |
| --- | --- |
| 1 | 3  1 2 3  4 5 6  7 8 9 |
| 2 | 4  0 2 3 4  0 0 3 4  0 0 0 4  0 0 0 0 |
| 3 | 5  0 0 0 0 0  0 0 0 0 0  0 0 3 0 0  0 0 0 4 0  0 0 0 0 5 |

1. 輸出範例：

|  |  |
| --- | --- |
| 1 | NO |
| 2 | YES |
| 3 | YES |

1. 程式碼解答(或是直接附.java檔也可以)：

SumOfDiagonal.java

題目03名稱：**2x2 Matrix Inverse**

1. 題目描述(程式範例描述)：

The inverse of a square matrix A is denoted A-1, such that A×A-1 = I, where I is the identity matrix with all 1’s on the diagonal and 0 on all other cells. For example, the inverse of matrix  is , i.e., 

The inverse of a matrix A can be obtained using the following formula:



Implement the following method to obtain an inverse of the matrix:

public static double[][] inverse(double[][] A)

The method returns null if ad – bc is 0.

1. 輸入說明：

The line contains 4 numbers to represent the 2x2 matrix.

1. 輸出說明：

The inverse of the matrix

1. 裝備限詞

|  |  |
| --- | --- |
| 出現次數 | 語句 |
|  |  |

1. 輸入範例：(2組公開測資、1組暗資)

|  |  |
| --- | --- |
| 1 | 1 2 3 4 |
| 2 | 0.5 2 1.5 4.5 |
| 3 | -1 5 2.5 0 |

1. 輸出範例：

|  |  |
| --- | --- |
| 1 | -2.0 1.0  1.5 -0.5 |
| 2 | -6.0 2.6666666666666665  2.0 -0.6666666666666666 |
| 3 | 0 0.4  0.2 0.08 |

1. 程式碼解答(或是直接附.java檔也可以)：

.java

題目04名稱：**Common Prefix**

1. 題目描述(程式範例描述)：

Complete the prefix method that returns the common prefix of two strings. For example, the common prefix of "distance" and "disinfection" is "dis". The JAVA program is given as follows:

class Finder {

public static String prefix(String s1, String s2){

// Add Java statements

}

}

public class Main {

public static void main(String[] args) {

String s1, s2;

Scanner sc = new Scanner(System.in);

s1 = sc.next();

s2 = sc.next();

System.out.println(Finder.prefix(s1, s2));

}

}

If the two strings have no common prefix, the method returns an empty string.

1. 輸入說明：

Two strings.

1. 輸出說明：

The common prefix of two input strings.

1. 裝備限詞

|  |  |
| --- | --- |
| 出現次數 | 語句 |
| 1 | 題目給定的程式碼 |

1. 輸入範例：(2組公開測資、1組暗資)

|  |  |
| --- | --- |
| 1 | quickly  quite |
| 2 | distance  disinfection |
| 3 | computer  mobile |

1. 輸出範例：

|  |  |
| --- | --- |
| 1 | qui |
| 2 | dis |
| 3 |  |

1. 程式碼解答(或是直接附.java檔也可以)：

Main.java

Finder.java (解答)

題目05名稱：**Pyramid**

1. 題目描述(程式範例描述)：

Write an application that inputs an integer number N from the user and displays the following pyramid pattern (using N as the height of the pyramid). All asterisks (\*) should be printed by a single statement of the form System.out.print("\*"); which causes the asterisks to print side by side. A statement of the form System.out.println(); can be used to move to the next line. A statement of the form System.out.print(" "); can be used to display a space if needed. There should be no other output statements in the program.

1. 輸入說明：

An integer number ***N***.

1. 輸出說明：

A pyramid pattern with height ***N***.

1. 裝備限詞

|  |  |
| --- | --- |
| 出現次數 | 語句 |
|  |  |

1. 輸入範例：(2組公開測資、1組暗資)

|  |  |
| --- | --- |
| 1 | 1 |
| 2 | 2 |
| 3 | 5 |

1. 輸出範例：

|  |  |
| --- | --- |
| 1 | \* |
| 2 | \*  \*\*\* |
| 3 | \*  \*\*\*  \*\*\*\*\*  \*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\* |

1. 程式碼解答(或是直接附.java檔也可以)：

Pyramid.java