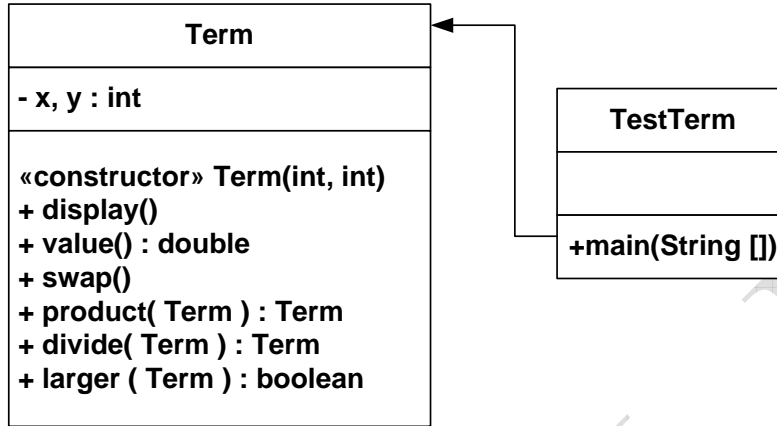


السؤال الثاني:

الحد الرياضي x^y (الأساس x مرفوعه للأس y) يمثل بالصنف Term كما هو موضح في UML:



- constructor : يقوم بإستقبال قيمتين وإنشاء object مع ملاحظة أن 0^0 مرفوضه وتستبدل بالقيمة 1^1
- display : تطبع الحد بالشكل $[X^Y]$ – مثال: $[3^4]$
- value : ترجع قيمة الحد – مثال: إذا كان الحد 3^4 فترجع الدالة القيمة 81.0
- swap : تقوم بإستبدال x مع y – مثال: إذا كان الحد 7^3 فإنه يصبح 3^7
- product : تقوم بضرب حدين بشرط تساويهما في الأساس وترجع الحد الناتج، أما إذا كان الأساس مختلف فترجع 0^0 – مثال: حاصل ضرب الحدين 4^9 و 4^6 هو 4^{15}
- divide : تقوم بقسمة حدين بشرط تساويهما في الأساس وترجع الحد الناتج، أما إذا كان الأساس مختلف فترجع 0^0 – مثال: حاصل قسمة الحدين 4^9 على 4^6 هو 4^3
- larger : تقوم بمقارنة حدين بشرط تساويهما في الأساس وترجع true إذا كان الأول أكبر من الثاني، أما إذا كان الأساس مختلف فترجع false

المطلوب:

- أكتب الصنف Term ثم اكتب برنامج يقوم بالتالي:

1. يقرأ عددين صحيحين a و b ثم يقوم بإنشاء Term يمثل الحد a^b ثم يطبعه ويطبع قيمته value
2. ينفذ العملية swap ليصبح الحد المعطى b^a ثم يطبع ناتج مقارنة بالعملية larger مع الحد 5^9
3. يطبع حاصل ضرب الحد المعطى مع 5^9 ويطبع حاصل قسمة الحد 7^8 على الحد المعطى.

```
package myPackage;

public class Term {
    private int x, y;

    public Term() {
        x = 0; y = 0;
    }

    public Term(int a, int b) {
        if ( a == 0 && b == 0)
            a = b = 1;
        x = a;
        y = b;
    }

    public void display() {
        System.out.println("[ "+x+"^"+y+" ]");
    }

    public double value() {
        double p = 1;
        for (int j = 1; j <= y; j++)
            p *= x;
        return p;
    }

    public void swap() {
        int temp = x; x = y; y = temp;
    }

    public Term product( Term obj2) {
        Term obj3;
        if ( x == obj2.x)
            obj3 = new Term(x, y + obj2.y);
        else
            obj3 = new Term();
        return obj3;
    }
}

/* Another solution:
public Term product( Term obj2) {
    Term obj3 = new Term(x,y);
    if ( x == obj2.x)
        obj3.y += obj2.y);
    else
        obj3.x = obj3.y = 0;
    return obj3;
}
*/

public Term divide ( Term obj2) {
    Term obj3;
    if ( x == obj2.x)
        obj3 = new Term(x, y - obj2.y);
    else
        obj3 = new Term();
    return obj3;
}

public boolean larger(Term obj2) {
```

```

        return x == obj2.x && y > obj2.y;
    }
}
//=====

package myPackage;

import java.util.Scanner;

public abstract class TermTest {
    public static void main(String[] args) {

        Scanner s = new Scanner(System.in);
        System.out.println("Enter the base: ");
        int a = s.nextInt();
        System.out.println("Enter the exponent: ");
        int b = s.nextInt();

        Term t1 = new Term(a, b);
        System.out.print("The term is : ");
        t1.display();
        System.out.println("The value is : " + t1.value());
        t1.swap();
        Term t2 = new Term(5, 9);
        if (t1.larger(t2))
            System.out.println("The term is larger than 5^9.");
        else
            System.out.println("The term is NOT larger than 5^9.");

        Term t3 = t1.product(t2);
        System.out.print("The result of product is : ");
        t3.display();

        Term t4 = new Term(7,8);

        t3 = t4.product(t1);
        System.out.print("The result of divide is : ");
        t3.display();
    }
}

```