第十四周12/15

設計一個 Polynomial 類別，可以表示任意多項式，並且提供多項式的加法與減法運算。撰寫主程式展現 Polynomial 類別的這些功能。注意: 相加或相減的二個多項式，至少要有一個非零項目的次方數一樣，例如: a(x)=12x^5+5x+1, b(x)=6x^7-12x-9

```
/* 12/15 楊育哲
 * 實作第一題: polynomals加減
 */
public class h1_1215_w {
    public static void main(String[] args){
        Polynomals.Iterator Ai = new Polynomals(3).new Iterator();
        Polynomals.Iterator Bi = new Polynomals(3).new Iterator();
        for(int i=3; i>0; i--) Ai.add(i, i-1); //3x^2+2x^1+1x^0
        for(int i=5; i>2; i--) Bi.add(i, i-1); //5x^4+4x^3+3x^2
        Polynomals C = Ai.calc(Bi, true); //true for add
        Polynomals D = Ai.calc(Bi, false); //false for sub
        Polynomals.Iterator Ci=C.new Iterator();
        Polynomals.Iterator Di=D.new Iterator();
        System.out.print("A(x) = "); Ai.show();
        System.out.print("B(x) = "); Bi.show();
        System.out.print("C(x) = A(x) + B(x) = "); Ci.show();
        System.out.print("D(x) = A(x) - B(x) = "); Di.show();
    }
```

```
}
class Polynomals{
    private Iterator.term[] termArray;
    private int terms;
    public Polynomals(int size){
        terms = 0;
        if(size<1) size=1;
        termArray = new Iterator.term[size];
    }
    class Iterator{
        class term {
            public float coef;// private 會取不到值, 所以改public
            public int exp;
            public term(float c, int e){coef=c; exp=e;}
        }
        private int index;
        public Iterator(){index=0;}
        public void add(float c, int e){
            if(terms>=termArray.length){
                term[] tmp=new term[terms*2];
                for(int i=0; i<terms; i++) tmp[i]=termArray[i];
                termArray = tmp;
            }
            termArray[terms++]=new term(c, e);
        }
```

```
        public boolean end(){return index==terms;}
        public term current(){return termArray[index];}
        public void next(){index++;}
        public void reset(){index=0;}
        public Polynomals calc(Iterator bi, Boolean oper){
            reset(); bi.reset();
            Polynomals C = new Polynomals(terms);
            Polynomals.Iterator Ci = C.new Iterator();
            while(!end()&&!bi.end()){
                if(current().exp==bi.current().exp){
                    Ci.add((oper)?current().coef+bi.current().coef:current().coef-bi.current().c
                    next(); bi.next();
                }else if(current().exp>bi.current().exp){
                    Ci.add(current().coef, current().exp);
                    next();
                }else{
                    Ci.add((oper)?bi.current().coef:(-1)*bi.current().coef, bi.current().exp);
                    bi.next();
                }
            }
            while(!end()){Ci.add(current().coef, current().exp);next();}
            while(!bi.end()){Ci.add((oper)?bi.current().coef:(-1)*bi.current().coef, bi.current(
            return C;
        }
        public void show(){
```

```
            reset();
            System.out.printf("%.2fx^%d ", current().coef, current().exp);
            next();
            while(!end()){
                if(current().coef>=0) System.out.printf("+ %.2fx^%d ", current().coef, current()
                else System.out.printf("- %.2fx^%d ", (-1)*current().coef, current().exp);
                next();
            }
            System.out.println("");
        }
    }
}
```

輸出:

A(x) = 3.00x^2 + 2.00x^1 + 1.00x^0
B(x) = 5.00x^4 + 4.00x^3 + 3.00x^2
C(x) = A(x) + B(x) = 5.00x^4 + 4.00x^3 + 6.00x^2 + 2.00x^1 + 1.00x^0
D(x) = A(x) - B(x) = -5.00x^4 - 4.00x^3 + 0.00x^2 + 2.00x^1 + 1.00x^0