

레포트#1

자료구조(SW) 2분반

소프트웨어학과

32170578 김산

<코드>

```
#include <iostream >
using namespace std;
class Polynomial; // 전방선언
class Term{
friend Polynomial; // Polynomial 클래스에서 접근가능
private:
    float coef; // 계수
    int exp; //지수
};
class Polynomial{
private:
    Term *termArray; //0이아닌 항의 배열
    int capacity; //termArray의 크기
    int terms; //0이 아닌 항의 수
public:
    void Print();
    void NewTerm(const float theCoeff,const int theExp);
    Polynomial Add(Polynomial b); //다항식의 덧셈 연산
    Polynomial Multiply(Polynomial b); //다항식의 곱셈 연산
    Polynomial();
    ~Polynomial(){};
};
void Polynomial::Print(){ //다항식을 출력한다.
    cout << termArray[0].coef <<"x^"<<termArray[0].exp;
    for (int i =1; i < terms; i ++){
        cout <<" + " << termArray[i].coef <<"x^"<< termArray[i].exp;
    }
}
void Polynomial::NewTerm(const float theCoeff,const int theExp){ //새로운 항을 termArray끝에 첨가한다.
    if(terms ==capacity){
        capacity *=2;
        Term *temp =new Term[capacity];
        for (int i =0; i < terms; i ++){
            temp[i] = termArray[i];
        }
        delete [] termArray;
        termArray = temp;
    }
    termArray[terms].coef = theCoeff;
    termArray[terms++].exp = theExp;
}
Polynomial Polynomial::Add(Polynomial b){ // 다항식의 덧셈 연산
    Polynomial c;
    int aPos =0, bPos =0;
    while((aPos < terms) && (bPos < b.terms)){
        if(termArray[aPos].exp == b.termArray[bPos].exp){
            float t = termArray[aPos].coef + b.termArray[bPos].coef;
            if(t){
                c.NewTerm(t, termArray[aPos].exp);
            }
            aPos++;
        }
    }
}
```

```

        bPos++;
    }
    else if(termArray[aPos].exp < b.termArray[bPos].exp){
        c.NewTerm(b.termArray[bPos].coef, b.termArray[bPos].exp);
        bPos++;
    }
    else{
        c.NewTerm(termArray[aPos].coef,termArray[aPos].exp);
        aPos++;
    }
}
for (; aPos < terms; aPos ++){
    c.NewTerm(termArray[aPos].coef, termArray[aPos].exp);
}
for (; bPos < b.terms; bPos ++){
    c.NewTerm(b.termArray[bPos].coef, b.termArray[bPos].exp);
}
return c;
}
Polynomial Polynomial::Multiply(Polynomial b){ //다항식의 곱셈 연산
    Polynomial c;
    for (int aPos =0; aPos < terms; aPos ++){
        Polynomial temp;
        for (int bPos =0; bPos < b.terms; bPos ++){
            float t = termArray[aPos].coef * b.termArray[bPos].coef; //다항식 계수 계산
            int s = termArray[aPos].exp + b.termArray[bPos].exp; //다항식 지수 계산
            temp.NewTerm(t,s); //B의 단항과 A 다항식의 곱셈 결과를 temp에 저장
        }
        c = c.Add(temp); //B의 각 단항과 A 다항식의 곱셈 결과를 모두 더하여 결과 C반환
    }
    return c;
}
Polynomial::Polynomial(){ //다항식 생성자
    capacity =4;
    terms =0;
    termArray =new Term[capacity];
}
int main(){ //main 함수
    Polynomial A,B,C;
    int i,n,e;
    float c;
    //다항식 A입력
    cout <<"다항식 A의 항의 수:";
    cin >> n;
    for (int i =1; i <= n; i ++){
        cout <<"다항식 A의 "<< i <<" 번째 항의 계수와 지수 : ";
        cin >> c >> e;
        A.NewTerm(c,e);
    }
    //다항식 B입력
    cout <<"다항식 B의 항의 수:";
    cin >> n;
    for (int i =1; i <= n; i ++){
        cout <<"다항식 B의 "<< i <<" 번째 항의 계수와 지수 : ";
        cin >> c >> e;
        B.NewTerm(c,e);
    }
}

```


case 3

```

TERMINAL  PROBLEMS  OUTPUT  DEBUG CONSOLE

다항식 A의 항의 수:1
다항식 A의 1 번째 항의 계수와 지수 : 4 1
다항식 B의 항의 수:3
다항식 B의 1 번째 항의 계수와 지수 : 3 2
다항식 B의 2 번째 항의 계수와 지수 : 4 1
다항식 B의 3 번째 항의 계수와 지수 : 5 0
입력된 다항식 A :  $4x^1$ 
입력된 다항식 B :  $3x^2 + 4x^1 + 5x^0$ 
A x B 연산결과 :  $12x^3 + 16x^2 + 20x^1$ 
[1] + Done                                     "/usr/bin/gdb"
```

case 4

```

TERMINAL  PROBLEMS  OUTPUT  DEBUG CONSOLE

다항식 A의 항의 수:1
다항식 A의 1 번째 항의 계수와 지수 : 5 0
다항식 B의 항의 수:3
다항식 B의 1 번째 항의 계수와 지수 : 3 6
다항식 B의 2 번째 항의 계수와 지수 : 2 1
다항식 B의 3 번째 항의 계수와 지수 : 7 0
입력된 다항식 A :  $5x^0$ 
입력된 다항식 B :  $3x^6 + 2x^1 + 7x^0$ 
A x B 연산결과 :  $15x^6 + 10x^1 + 35x^0$ 
[1] + Done                                     "/usr/bin/gdb"
```

case 5

```

TERMINAL  PROBLEMS  OUTPUT  DEBUG CONSOLE

다항식 A의 항의 수:5
다항식 A의 1 번째 항의 계수와 지수 : 1 5
다항식 A의 2 번째 항의 계수와 지수 : 1 4
다항식 A의 3 번째 항의 계수와 지수 : 1 3
다항식 A의 4 번째 항의 계수와 지수 : 1 2
다항식 A의 5 번째 항의 계수와 지수 : 1 1
다항식 B의 항의 수:2
다항식 B의 1 번째 항의 계수와 지수 : 1 1
다항식 B의 2 번째 항의 계수와 지수 : 1 0
입력된 다항식 A :  $1x^5 + 1x^4 + 1x^3 + 1x^2 + 1x^1$ 
입력된 다항식 B :  $1x^1 + 1x^0$ 
A x B 연산결과 :  $1x^6 + 2x^5 + 2x^4 + 2x^3 + 2x^2 + 1x^1$ 
[1] + Done                                     "/usr/bin/gdb"
```