

## 資料結構 Homework 2

班級:資工二甲 姓名:胡楨 學號:41243126

這次的作業是要寫出一個可以實現多項式相加和相乘的程式

程式說明:

```
class tern {  
|   friend class polynomial;  
private:  
|   float coef; // 係數  
|   int exp;    // 指數  
public:  
|   tern(float c = 0, int e = 0) : coef(c), exp(e) {}  
};
```

在 **tern** 的類別裡，代表多項式中的係數和指數。每項都由兩個元素構成 **coef** 用來存入係數，**exp** 用來存入指數。**Public** 用來初始化 **tern** 類別，預設係數為 0 指數為 0。

在 **polynomial** 類別裡，**polynomial** 類別代表多項式，包含多個 **tern** 類別的元素。這個類別有處理多項式運算所需的各種功能。

```
class polynomial {  
private:  
    tern* termArray; // 儲存項目的動態陣列  
    int terms;       // 已使用的項目數量  
    int capacity;    // 陣列容量
```

`termArray`：動態陣列，用來存儲多項式的各項

`terms`：表示目前多項式中項目的數量。

`capacity`：動態陣列的容量，當容量不足時會進行擴容。

```
public:  
    polynomial(int initialCapacity = 5) : terms(0), capacity(initialCapacity) {  
        termArray = new tern[capacity];  
    }  
    ~polynomial() {  
        delete[] termArray; // 釋放動態陣列  
    }
```

`polynomial`：初始化多項式，並為項目陣列分配內存空間，初始容量預設為 5。當多項式被創建時，它會有一個空的項目陣列。

`~polynomial`：當 `polynomial` 對象銷毀時，會釋放分配的內存

```

// 空間不足時重新配置空間
void NewTerm(const float newCoef, const int newExp) {
    for (int i = 0; i < terms; ++i) {
        if (termArray[i].exp == newExp) { // 如果指數相同，合併係數
            termArray[i].coef += newCoef;
            return;
        }
    }
    if (terms == capacity) {
        capacity *= 2;
        term* temp = new term[capacity];
        copy(termArray, termArray + terms, temp);
        delete[] termArray;
        termArray = temp;
    }
    termArray[terms].coef = newCoef;
    termArray[terms++].exp = newExp;
}

```

NewTerm 函數 是在空間不足時需要重新配置空間，如果有相同指數的項（exp 相同），會將其係數合併（coef += newCoef），否則將該項加入 termArray 陣列。處理動態擴容：當 termArray 的容量滿了，會將其容量翻倍，並重新分配內存。

```

// 輸入多項式
void input() {
    int n;
    cout << "輸入多項式的項數: ";
    cin >> n;
    for (int i = 0; i < n; ++i) {
        float coef;
        int exp;
        cout << "輸入第 " << i + 1 << " 項的係數與指數: ";
        cin >> coef >> exp;
        NewTerm(coef, exp);
    }
}

```

輸入多項式 通過 NewTerm 函數將每項加入多項式。

```
// 多項式相加
polynomial add(const polynomial& poly) const {
    polynomial result(capacity + poly.capacity); // 預留足夠容量
    for (int i = 0; i < terms; ++i) {
        result.NewTerm(termArray[i].coef, termArray[i].exp);
    }
    for (int i = 0; i < poly.terms; ++i) {
        result.NewTerm(poly.termArray[i].coef, poly.termArray[i].exp);
    }
    return result;
}
```

這是多項式相加的函數，將使用著輸入兩個多項式將兩個多項式加入到一個新的多項式 **result** 中。如果有相同指數的項，會自動合併。

```
// 多項式相乘
polynomial mult(const polynomial& poly) const {
    polynomial result(capacity * poly.capacity); // 預留足夠容量
    for (int i = 0; i < terms; ++i) {
        for (int j = 0; j < poly.terms; ++j) {
            result.NewTerm(
                termArray[i].coef * poly.termArray[j].coef,
                termArray[i].exp + poly.termArray[j].exp
            );
        }
    }
    return result;
}
```

這是多項式相乘的函數，將使用著輸入兩個多項式加入對於多項式中的每一項，將其與另一個多項式中的每一項相乘，並生成新的項目。每個項目的係數相乘，指數相加。

```
// 計算多項式在 x 值上的結果
float eval(float x) const {
    float result = 0;
    for (int i = 0; i < terms; ++i) {
        result += termArray[i].coef * pow(x, termArray[i].exp);
    }
    return result;
}
```

使用者輸入 X 的值，並帶入到第一個多項式裡，通過 `coef * x^exp` 計算每一項的值，然後將所有項的結果加總。

```
// 輸出多項式
void display() const {
    if (terms == 0) {
        cout << "0";
        return;
    }
    for (int i = 0; i < terms; ++i) {
        if (i > 0 && termArray[i].coef > 0) cout << " + ";
        if (termArray[i].exp == 0) {
            cout << termArray[i].coef;
        }
        else {
            cout << termArray[i].coef << "x^" << termArray[i].exp;
        }
    }
    cout << endl;
};
```

輸出多項式，這個函數將多項式的每一項顯示出來。

```

int main() {
    polynomial poly1, poly2;

    cout << "輸入第一個多項式:" << endl;
    poly1.input();
    cout << "輸入第二個多項式:" << endl;
    poly2.input();

    cout << "第一個多項式為: ";
    poly1.display();
    cout << "第二個多項式為: ";
    poly2.display();

    polynomial sum = poly1.add(poly2);
    polynomial product = poly1.mult(poly2);

    cout << "多項式相加結果: ";
    sum.display();
    cout << "多項式相乘結果: ";
    product.display();

    float x;
    cout << "請輸入 x 的值以求解第一個多項式: ";
    cin >> x;
    cout << "第一個多項式在 x = " << x << " 時的值為: " << poly1.eval(x) << endl;

    return 0;
}

```

主程式:使用者會先輸入兩個多項式，並顯示兩個多項式的加總和相乘，最後再叫使用者輸入  $x$  值，帶入到第一個多項式裡，最後產生帶入  $x$  值的結果。

```
輸入第一個多項式：
輸入多項式的項數：2
輸入第 1 項的係數與指數：3 2
輸入第 2 項的係數與指數：2 2
輸入第二個多項式：
輸入多項式的項數：3
輸入第 1 項的係數與指數：3 2
輸入第 2 項的係數與指數：2 3
輸入第 3 項的係數與指數：2 3
第一個多項式為：5x^2
第二個多項式為：3x^2 + 4x^3
多項式相加結果：8x^2 + 4x^3
多項式相乘結果：15x^4 + 20x^5
請輸入 x 的值以求解第一個多項式：4
第一個多項式在 x = 4 時的值為：80

C:\Users\huzhe\OneDrive\Desktop\homework2\x64\Debug\homework2.exe (流程 16972) 已結束，代碼為 0 (0x0)。
若要在偵錯停止時自動關閉主控台，請啟用【工具】->【選項】->【偵錯】->【偵錯停止時，自動關閉主控台】。
按任意鍵關閉此視窗。]
```

這是輸出的結果。

心得討論：

這次的程式作業遇到很多困難，有很多地方我還是搞不懂，我就小不太出來，但我有請教我的朋友幫忙，但也有借助一點 AI，讓我把這個難題解開來，經過這次的作業，我也學到了許多有關於類別函數和一些類別的解構子，讓我的資料處理能夠的更有效率。