

MÔN : CÔNG NGHỆ JAVA

Bài thực hành 1.4 : Viết chương trình xử lý đa thức

I. Mục tiêu :

- Giúp SV làm quen với môi trường lập trình trực quan NetBeans (version 7.3.1) để viết các ứng dụng đơn giản chạy ở chế độ text.

II. Nội dung :

- Dùng NetBeans viết chương trình nhỏ chạy ở chế độ textmode (console), thực hiện các yêu cầu sau :
 1. miêu tả kiểu để lưu giữ từng toán hạng của đa thức gồm 2 thông tin : hệ số và số mũ.
 2. sắp xếp danh sách các toán hạng của đa thức theo thứ tự giảm dần của số mũ, phương pháp sắp xếp là Insertion Sort.
 3. chuẩn hóa đa thức : mỗi số mũ chỉ xuất hiện trong đa thức 1 lần.

Thí dụ có đa thức gốc $R(x) = x^3 + x^2 + x + 2x^2 + 2x + 2$

sau khi chuẩn hóa thành $R(x) = x^3 + 3x^2 + 3x + 2$
 4. Nhân 2 đa thức.

III. Chuẩn đầu ra :

- Sinh viên làm quen môi trường NetBeans, biết dùng nó để viết chương trình đơn giản chạy ở chế độ text.
- Sinh viên nắm vững cách viết đoạn lệnh thực hiện các công thức lập và biết cách dùng các kiểu dữ liệu cơ bản định sẵn khi cần.

IV. Phân tích :

1. Ta dùng phát biểu class để miêu tả thông tin về mỗi số hạng của đa thức như sau :

```
class Sohang {  
    public int somu; //số mũ  
    public float heso; //hệ số  
};
```

2. Ý tưởng sắp xếp danh sách theo giải thuật "Insertion Sort" như sau :

Lập xác định vị trí đúng cho từng số hạng i (từ 0 tới n) của đa thức :

- Lập tìm số hạng có số mũ lớn nhất từ vị trí i tới n
- Hoán vị nó với số hạng i

Từ thuật giải được viết bằng ngôn ngữ tự nhiên ở trên, ta sẽ dịch nó thành hàm SortItems() được viết bằng Java như trong mã nguồn dưới đây.

3. Chuẩn hóa đa thức : được thực hiện thông qua 2 bước chính :

- sắp xếp đa thức theo thứ tự số mũ giảm dần (gọi hàm SortItems() ở bước 2).
- duyệt tuần tự từ đầu đến cuối đa thức để cộng các số hạng liên tiếp có cùng số mũ lại thành 1 số hạng.

Từ thuật giải được viết bằng ngôn ngữ tự nhiên ở trên, ta dịch nó thành hàm Chuanhoa() được viết bằng Java như trong mã nguồn dưới đây.

4. Nhân đa thức : được thực hiện thông qua 2 bước chính :

- lập nhân từng số hạng của đa thức 1 với tất cả các số hạng của đa thức 2. Kết quả được lưu vào đa thức kết quả.
- Chuẩn hóa đa thức kết quả.

Từ thuật giải được viết bằng ngôn ngữ tự nhiên ở trên, ta dịch nó thành hàm Nhan() được viết bằng Java như trong mã nguồn dưới đây.

V. Qui trình :

1. Chạy NetBeans, nếu cửa sổ Project bên trái màn hình còn hiển thị Project nào đó (của lần cuối cùng trước đó), ấn phải chuột trên phần tử gốc của cây Project để hiển thị menu các chức năng, rồi chọn option Close để đóng Project tương ứng lại.
2. Chọn menu File.New Project để máy hiển thị cửa sổ "New Project", chọn mục "Java" trong Listbox Categories, chọn mục "Java Application" trong Listbox Projects rồi click button Next để hiển thị cửa sổ "New Application".
3. Xác định thư mục chứa Project ở textbox "Project Location", nhập "NBDathuc" vào textbox "Project Name", click button Finish để máy tạo thực sự Project. Cửa sổ soạn mã nguồn của class chương trình có tên là NBDathuc hiển thị.

4. Viết code cho thân của class NBDathuc như sau :

```
public class NBDathuc {
    //hàm sắp xếp đa thức theo thứ tự số mũ giảm dần
    static void SortItems (Sohang[] dathuc, int soluong) {
        //định nghĩa các biến cần dùng
        int i,j, max;
        Sohang temp;
        //Lập xác định vị trí đúng cho từ số mũ
        for (i = 0; i < soluong-1; i++) {
            //Tìm phần tử có số mũ lớn nhất từ vị trí i
            max = i;
            for (j = i + 1; j < soluong; j++)
                if (dathuc[max].somu < dathuc[j].somu)
                    max = j;
            //hoán vị phần tử có số mũ lớn nhất về vị trí i
            temp = dathuc[i];
            dathuc[i] = dathuc[max];
            dathuc[max] = temp;
        }
    }

    //hàm chuẩn hóa đa thức, mỗi số mũ chỉ có 1 phần tử
    static int Chuanhoa (Sohang[] dathuc, int soluong) {
        //định nghĩa các biến cần dùng
        int i,j;
        Sohang[] temp = new Sohang[soluong];
        //sắp xếp tự các phần tử trong đa thức
        SortItems(dathuc,soluong);
        //Lập việc gộp các phần tử có số mũ giống nhau
        j = 0; temp[0] = dathuc[0];
        i = 1;
        while (i < soluong)
            if (temp[j].somu == dathuc[i].somu) {
                temp[j].heso += dathuc[i].heso;
                i++;
            } else {
```

```

    j++;
    temp[j] = dathuc[i];
    i++;
}
//copy kết quả về dathuc gốc
for (i = 0; i <= j ; i++)
    dathuc[i] = temp[i];
//return số phần tử trong đa thức chuẩn hóa
return j+1;
}

//hàm in đa thức để kiểm tra
static void Indathuc(Sohang[] dathuc, int soluong) {
    int i;
    if (soluong == 0) return;
    //in phần tử đầu tiên của đa thức
    System.out.print(dathuc[0].heso + "x ^" + dathuc[0].somu + " ");
    //lập in từng phần tử còn lại của đa thức
    for (i = 1; i < soluong; i++)
        if (dathuc[i].heso < 0)
            System.out.print(dathuc[i].heso + "x ^" + dathuc[i].somu + " ");
        else
            System.out.print(" + " + dathuc[i].heso + "x ^" + dathuc[i].somu + " ");
    //in ký tự xuống dòng
    System.out.println();
}

//hàm nhân 2 đa thức
static int Nhan (Sohang[] dathuc1, int soluong1, Sohang[] dathuc2, int soluong2, Sohang[]
dathuc3) {
    //định nghĩa các biến cần dùng
    int i,j,k;
    k = 0;
    //lập nhân từng số hạng của đa thức 1
    for (i=0; i < soluong1; i++)
        //với tất cả các số hạng của đa thức 2
        for (j = 0; j < soluong2; j++) {
            dathuc3[k].somu = dathuc1[i].somu+dathuc2[j].somu;
            dathuc3[k].heso = dathuc1[i].heso*dathuc2[j].heso;
            k++;
        }
    //chuẩn hóa đa thức kết quả
    return Chuanhoa (dathuc3,k);
}

//chương trình chính thử dùng các hàm chức năng
static void Main(string[] args) {
    Sohang[] dathuc1 = new Sohang[] { new Sohang(1,3), new Sohang(1,2), new
    Sohang(1,1),new Sohang(-2,2),new Sohang(2,1),new Sohang(2,0)};
    //1. thử sắp xếp và in ra để kiểm tra

```

```

System.out.println("Đa thức gốc là : ");
Indathuc(dathuc1,6);
SortItems(dathuc1,6);
System.out.println("Đa thức được sort là : ");
Indathuc(dathuc1,6);

```

```

Sohang[] dathuc2 = new Sohang[] { new Sohang(1,3), new Sohang(1,2), new Sohang(1,1),
new Sohang(-2,2), new Sohang(2,1),new Sohang(2,0)};

```

```

//2. thử chuẩn hóa đa thức và in ra để kiểm tra

```

```

System.out.println("Đa thức gốc là : ");
Indathuc(dathuc2,6);
int somu = Chuanhoa(dathuc2,6);
System.out.println("Đa thức được chuẩn hóa là : ");
Indathuc(dathuc2,somu);

```

```

//3. thử nhân 2 đa thức và in ra để kiểm tra

```

```

Sohang[] dathuc3 = new Sohang[] { new Sohang(1,3), new Sohang(1,2), new Sohang(1,1),
new Sohang(-2,2), new Sohang(2,1),new Sohang(2,0)};

```

```

Sohang[] dathuc4 = new Sohang[] { new Sohang(1,3), new Sohang(1,2), new Sohang(1,1),
new Sohang(-2,2), new Sohang(2,1),new Sohang(2,0)};

```

```

Sohang[] dathuc5;

```

```

int soluong;

```

```

//cấp phát bộ nhớ chứa đa thức kết quả

```

```

dathuc5 = new Sohang[6*6];

```

```

for (int i=0; i<36; i++)

```

```

    dathuc5[i] = new Sohang(0,0);

```

```

soluong = Nhan(dathuc3, 6, dathuc4, 6, dathuc5);

```

```

System.out.println("Đa thức kết quả là : ");

```

```

Indathuc(dathuc5, soluong);

```

```

} //hết hàm Main

```

```

} //hết class

```

5. Dời về đầu file mã nguồn, thêm các lệnh import sau đây vào sau lệnh package :

```

//import các package cần dùng

```

```

import java.io.*;

```

```

import java.util.*;

```

6. Trong cửa sổ quản lý Project, ấn kép chuột vào phần tử gốc có tên là NBDathuc để mở rộng nội dung của nó, bạn sẽ thấy folder "Source Packages", ấn kép chuột vào folder "Source Packages" bạn sẽ thấy folder nbdathuc. Ấn phải chuột trên folder nbdathuc, chọn chức năng New.Java Class để máy hiển thị cửa sổ "New Java class", nhập tên class mới là Sohang, click chuột vào button Finish để máy tạo ra class tương ứng. Khi cửa sổ soạn mã nguồn của class mới hiển thị, hãy viết code cho class Sohang như sau :

```

//định nghĩa kiểu miêu tả 1 số mũ

```

```

public class Sohang {

```

```

    public float heso; //hệ số

```

```

    public int somu; //số mũ

```

```

    public Sohang(float h, int s) { heso = h; somu = s; }

```

```

};

```

7. Chọn menu Run.Run Project để dịch và chạy thử chương trình. Nếu có lỗi từ vựng và cú pháp thì sửa, nếu có lỗi run-time thì debug (thông qua các chức năng trong menu Debug) để xác định lỗi rồi sửa lỗi.
8. Nếu chương trình hết lỗi, chương trình sẽ chạy, cửa sổ output sẽ hiển thị các đa thức gốc và đa thức đã sắp thứ tự, đã chuẩn hóa,... Hãy kiểm tra kết quả và đánh giá tính đúng đắn của chương trình.