

**MÔN : CÔNG NGHỆ JAVA****Bài thực hành 1.3 : Viết chương trình thực hiện giải thuật lập****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), tính 3 hàm số theo công thức lập sau đây :

$$S1(x) = 1 + x + x^2/2! + x^3/3! + \dots + x^n/n!$$

$$S2(x) = 1 - x + x^2/2! - x^3/3! + \dots + (-1)^n x^n/n!$$

$$S3(x) = 1 + \sin(x) + \sin(2x) + \dots + \sin(nx)$$

**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.

**IV. Phân tích :**

Điểm chung của 3 hàm S1, S2, S3 là dùng cùng công thức lập có dạng sau :

$$S(x) = 1 + \sum_{n=1}^{\infty} e_n$$

trong đó mỗi thành phần  $e_n$  phụ thuộc vào  $x$  và  $n$ ,  $e_n$  thường nhỏ dần khi  $n$  tăng lên đến khi đủ nhỏ (nhỏ hơn sai số epsilon qui định) thì ta bỏ đi :

- đối với S1(x) thì  $e_n = x^n/n!$
- đối với S2(x) thì  $e_n = (-1)^n x^n/n!$
- đối với S3(x) thì  $e_n = \sin(nx)$

Thuật giải lập để tính công thức trên gồm các bước sau :

1. Lúc đầu, thiết lập  $S = 1$  và  $n = 1$
2. Tính  $e_n$ , tích lũy  $e_n$  vào S.
3. Kiểm tra  $e_n < \text{epsilon}$ . Nếu thỏa thì dừng, nếu không thì tăng  $n$  1 đơn vị và thực hiện lại bước 2.

Lập trình chẳng qua là dịch thuật giải ở dạng ngôn ngữ tự nhiên thành ngôn ngữ lập trình cho máy hiểu và thực thi tự động dùm con người khi cần. Sau đây là qui trình điển hình để viết chương trình nhỏ thực hiện 3 công thức lập trên bằng NetBeans :

**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 "NBTinhLap" 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 NBTinhLap hiển thị.
4. Viết code cho thân của class NBTinhLap như sau :

```

public class NBTinhLap {
    //định nghĩa hằng gọi nhớ miêu tả sai số mong muốn
    static final double EPSILON = 0.0000001;
    static Scanner input;    //định nghĩa biến đối tượng phục vụ đọc dữ liệu

    //định nghĩa hàm tính n!
    static long giaithua(int n)
    {
        if (n <= 1) return 1;
        return n*giaithua(n-1);
    }

    //định nghĩa hàm tính thành phần  $x^n/n!$ 
    static double EnS1(double x, int n)
    {
        return (Math.pow(x,n))/giaithua(n);
    }

    //định nghĩa hàm tính công thức S1(x)
    static double S1(double x)
    {
        //1. Lúc đầu, thiết lập S = 1 và n = 1
        double S = 1;
        int n = 1;
        double En;
        do {
            //2. Tính En, tích lũy En vào S
            En = EnS1(x,n);
            S = S + En;
            n++;
        } while (Math.abs(En) >= EPSILON); //3. Lặp bước 2 nếu En >= epsilon
        //gửi kết quả về lệnh gọi hàm này
        return S;
    }

    //định nghĩa hàm tính thành phần  $(-1)^n x^n/n!$ 
    static double EnS2(double x, int n)
    {
        return Math.pow(-1,n)*Math.pow(x,n)/giaithua(n);
    }

    //định nghĩa hàm tính công thức S2(x)
    static double S2(double x)
    {
        //1. Lúc đầu, thiết lập S = 1 và n = 1
        double S = 1;
        int n = 1;
        double En;
        do {
            //2. Tính En, tích lũy En vào S

```

```

    En = EnS2(x,n);
    S = S + En;
    n = n + 1;
} while (Math.abs(En) >= EPSILON); //3. Lặp bước 2 nếu En >= epsilon
//gửi kết quả về lệnh gọi hàm này
return S;
}

//định nghĩa hàm tính công thức S3(x)
static double S3(double x)
{
    //1. Lúc đầu, thiết lập S = 1 và n = 1
    double S = 1;
    int n = 1;
    double En;
    do {
        //2. Tính En, tích lũy En vào S
        En = Math.sin(n*x);
        S = S + En;
        n++;
    } while (Math.abs(En) >= EPSILON); //3. Lặp bước 2 nếu En >= epsilon
    //gửi kết quả về lệnh gọi hàm này
    return S;
}

//chương trình chính thử dùng các hàm được viết
public static void main(String[] args)
{
    double x;
    String buf;
    //tạo đối tượng nhập dữ liệu thuộc kiểu cơ bản
    input = new Scanner(System.in);
    //yêu cầu người dùng nhập x
    System.out.print("Nhập x : ");
    x = input.nextDouble();
    //gọi thử hàm S1(x)
    System.out.println("S1(" + x + ") = " + S1(x));
    //gọi thử hàm S2(x)
    System.out.println("S2(" + x + ") = " + S2(x));
    //gọi thử hàm S3(x)
    System.out.println("S3(" + x + ") = " + S3(x));
}
} //kết thúc 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. 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.

7. Nếu chương trình hết lỗi, chương trình sẽ chạy, cửa sổ output sẽ hiển thị thông báo Nhập x :", bạn hãy dời cursor về bên phải thông báo rồi nhập giá trị x. Chương trình sẽ tính 3 công thức lập theo biến x và hiển thị kết quả.
8. Lập lại bước 6, 7 nhiều lần để tính 3 công thức lập theo nhiều biến x khác nhau và đánh giá kết quả.