

MÔN : CÔNG NGHỆ JAVA

Bài thực hành 2.1 : Viết chương trình sắp 8 con hậu

I. Mục tiêu :

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

II. Nội dung :

- Dùng NetBeans viết chương trình chạy ở chế độ textmode (console), thực hiện các yêu cầu sau :
 - tim tất cả phương án để sắp 8 con hậu vào bàn cờ 8x8 ô sao cho chúng không thể ăn được nhau (thỏa mãn đồng thời các điều kiện chúng không trên cùng hàng, cùng cột hay cùng đường chéo).
 - in các phương án dạng ma trận 2 chiều để người dùng xem dễ dàng.

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 thuật giải back-tracking và biết cách dùng các kiểu dữ liệu như array và class.

IV. Phân tích :

- Ta dùng thuật giải backtracking để tìm các phương án sắp 8 con hậu như sau :

+ Lập tìm vị trí con hậu ở từng cột, từ trái sang phải :

- Nếu sắp được cho cột c thì tăng c 1 đơn vị để tiếp tục cột kế tiếp, nếu cột c đã vượt qua MAX thì đã tìm được 1 cách mới.
- Nếu không sắp được cho cột c thì giảm c 1 đơn vị để thử tiếp các khả năng còn lại của cột cũ này. Nếu c < 0 thì hết cách.

- Để lưu vết xử lý trên từng cột, ta dùng biến array hang[MAX], mỗi phần tử hang[c] chứa chỉ số hàng đã xử lý lần cuối trước đây (-1 : chưa, MAX : hết).

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

V. Qui trình :

- 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.
- 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".
- Xác định thư mục chứa Project ở textbox "Project Location", nhập "NBSap8Conhau" 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 NBSap8Conhau hiển thị.
- Viết code cho thân của class NBSap8Conhau như sau :

```
public class NBSap8Conhau {  
    //định nghĩa các thuộc tính dữ liệu  
    static final int MAX = 7; //chỉ số max của hàng và cột  
    static int cachso = 0; //số cách sắp đã tìm được
```

```
static int[] hang = new int[MAX+1]; //danh sách chỉ số hàng đã thử trong từng cột
static int c; //vị trí cột đang sắp
```

```
//Điểm nhập của chương trình
```

```
public static void main(String[] args)
{
    // khởi động trạng thái xuất phát
    for (c = 0; c <= MAX; c++) hang[c] = -1;
    cachso = 0;
    //bắt đầu đặt con hậu ở cột 0 vào hàng 0
    hang[0] = 0;
    //bắt đầu tìm hàng cho con hậu ở cột thứ 2 (chỉ số là 1)
    c = 1;
    while (Tim1cach())
    {
        // tìm được cách sắp 8 con hậu
        InKetqua();
        //lùi lại cột trước để tiếp tục tìm cách sắp khác
        c--;
    }
    // hết cách --> dừng chương trình
    System.out.println("Số nghiệm là " + cachso);
    return;
}
```

```
//cố gắng tìm 1 phương án sắp 8 con hậu
```

```
static boolean Tim1cach()
{
    int h;
    while (c <= MAX)
    {
        // tìm chỉ số hàng cho con hậu ở cột c
        h = timhang(c);
        if (h >= 0)
        { //nếu tìm được
            //lưu hàng vào danh sách kết quả rồi tăng chỉ số cột để tiếp tục
            hang[c++] = h;
            continue;
        }
        //trường hợp không tìm được hàng cho con hậu ở cột c
        if (c > 0)
        {
            //nếu cột c không phải là cột đầu tiên thì :
            //xét lại cột c từ hàng đầu rồi lùi cột c
            hang[c--] = -1;
            continue;
        }
        //hết cách
        return false;
    }
}
```

```

    return true;
} // hết hàm Tim1cach

// hàm cố gắng tìm chỉ số hàng cho con hậu ở cột c
// trả về chỉ số hàng tìm được hay -1 để báo sai
static int timhang(int c)
{
    int h, hmin = hang[c] + 1;
    for (h = hmin; h <= MAX; h++) // thử hàng h
        if (testvitri(h, c)) return h;
    return -1;
}

// hàm kiểm tra xem có thể đặt con hậu ở vị trí h,c ?
// trả về TRUE nếu được, FALSE nếu không
static boolean testvitri(int h, int c) {
    int c1, h1;
    // cột c đang kiểm tra nên chưa có con hậu nào
    // xem có con hậu nào nằm ở hàng h ?
    for (c1 = 0; c1 < c; c1++)
        if (hang[c1] == h) return false;
    // xem có con hậu nào nằm trên đường chéo trên trái - dưới phải
    c1 = c-1; h1 = h-1;
    while (c1 >= 0 && h1 >= 0) {
        if (hang[c1] == h1) return false;
        h1--; c1--;
    }
    // xem có con hậu nào nằm trên đường chéo trên phải - dưới trái
    c1 = c-1; h1 = h+1;
    while (c1 >= 0 && h1 <= MAX) {
        if (hang[c1] == h1) return false;
        h1++; c1--;
    }
    return true;
}

// hàm in kết quả 1 cách sắp 8 con hậu theo yêu cầu
static void InKetqua() {
    int h, c;
    System.out.println("Cách sắp thứ : " + (++cachso));
    for (h = 0; h <= MAX; h++) {
        // Hiển thị hàng lưới ngang của bàn cờ
        for (c = 0; c <= MAX; c++) System.out.print(" +---");
        System.out.println(" +");
        // Hiển thị nội dung hàng thứ h của bàn cờ
        for (c = 0; c <= MAX; c++)
            if (hang[c] == h) System.out.print("| x ");
            else System.out.print("| ");
        System.out.println("|");
    }
}

```

```
// Hiển thị hàng lưới ngang cuối cùng của bàn cờ  
for (c = 0; c <= MAX; c++) System.out.print("+---");  
System.out.println("+");  
}  
} //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. 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 92 phương án xếp 8 con hậu khác nhau thỏa mãn yêu cầu. Hãy kiểm tra kết quả và đánh giá tính đúng đắn của chương trình.