

MÔN : CÔNG NGHỆ JAVA

Bài thực hành 7.3 : Viết phần mềm demo sự tương tranh giữa các threads

I. Mục tiêu :

- Giúp SV làm quen với việc dùng class Thread của package java.lang để quản lý thread.
- Giúp SV thấy được vấn đề tương tranh giữa các thread khi chúng cùng truy xuất tài nguyên dùng chung.

II. Nội dung :

- Xây dựng chương trình cho phép người dùng thực hiện quản lý các thread có tên là A-Z như kích hoạt chạy, tạm dừng, chạy lại, tăng/giảm quyền ưu tiên, dừng và xóa thread... bằng các thao tác ấn phím. Mỗi thread khi chạy sẽ hiển thị icon miêu tả mình lên form, icon sẽ chạy theo 1 phương xác định và khi dừng thành form thì dời lại theo nguyên lý vật lý.
- Quan sát cảnh icon của thread này đè mất icon của thread khác, đây là vấn đề tương tranh giữa các thread trong việc truy xuất các cell hiển thị trên form.

III. Chuẩn đầu ra :

- Sinh viên nắm vững và sử dụng thành thạo class Thread để quản lý thread.
- Sinh viên nắm vững vấn đề tương tranh giữa các thread khi chúng truy xuất tài nguyên dùng chung.

IV. Qui trình :

0. Dùng chương trình soạn thảo đồ họa như Paint, CorelDraw, Photoshop vẽ ảnh bitmap miêu tả từng thread từ A-Z, mỗi ảnh có độ lớn 30*30 pixel như sau :



Cắt các ảnh bitmap lên các file có tên tương ứng là ImageA.jpg đến ImageZ.jpg.

Vẽ tiếp 1 ảnh bitmap miêu tả màu nền đen như sau :



Cắt ảnh bitmap nền lên file có tên Black.jpg.

1. Chạy NetBean 7.3.1, nếu cửa sổ Project có hiển thị các Project cũ hãy đóng chúng 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 Java Application".
3. Xác định thư mục chứa Project ở textbox "Project Location", nhập "NBThreadDemo2" vào textbox "Project Name", click button Finish để máy tạo thực sự Project. Cửa sổ mã nguồn của class chương trình NBThreadDemo2 hiển thị. Soạn code cho hàm main như sau :

```
public static void main(String[] args) {  
    //tạo và hiển thị Form giao diện cho ứng dụng  
    MainForm dlg = new MainForm();  
    dlg.show();  
}
```

4. Dùng trình duyệt file duyệt đến thư mục quản lý Project vừa tạo ra, bạn thấy có nhiều thư mục con, vào thư mục con tên là src (chứa package mã nguồn của chương trình), tạo thư mục con mới với tên là images. Copy 27 file ảnh vừa vẽ ở bước 0 vào thư mục images.

5. Trong cửa sổ quản lý Project, làm hiển thị chi tiết package nbthreaddemo2 (hiện đang chứa file miêu tả chương trình NBThreadDemo2.java). Ấn phải chuột trên folder nbthreaddemo2, chọn chức năng New.Java Class để máy hiển thị cửa sổ "New Java Class", nhập tên class mới là MyThread, click chuột vào button Finish để máy tạo ra class tương ứng, cửa sổ soạn mã nguồn của class sẽ hiển thị, viết code cho class MyThread như sau :

```
package nbthreaddemo2;
//import các package cần dùng
import javax.swing.*;
import java.awt.*;
import java.util.*;
import java.awt.image.*;
import java.util.concurrent.TimeUnit;

//đặc tả class quản lý thread được dùng trong chương trình
public class MyThread extends Thread {
    //định nghĩa các hằng, thuộc tính cần dùng
    final double PI = 3.1416;
    final int xCell = 30;    //độ rộng của mỗi cell (pixel)
    final int yCell = 30;    //độ cao của mỗi cell (pixel)
    public Boolean fstart;    //trạng thái Start của thread
    public Boolean fstop;    //trạng thái Stop của thread
    public Boolean fsuspend;    //trạng thái Suspend của thread
    public Boolean WaitOne = false;    //trạng thái chờ truy xuất cell
    public BufferedImage Pic;    //icon miêu tả thread
    int xCount;    //độ rộng vùng chạy của thread (số cell)
    int yCount;    //độ cao vùng chạy của thread (số cell)
    int tgchay;    //thời gian tính trước mỗi bước chạy của thread
    public Point Pos = new Point(); //vị trí của thread trong vùng chạy
    double dblGocChay;    //góc chạy của thread
    double tx, ty;    //bước tăng theo x và y
    MainForm frm;    //đối tượng quản lý Form ứng dụng
    //hàm khởi tạo các thông số của thread
    public MyThread(Random rnd, int xMax, int yMax, MainForm frm)
    {
        xCount = xMax; yCount = yMax; this.frm = frm;
        Pos.x = (int)(rnd.nextInt(xCount));
        Pos.y = (int)(rnd.nextInt(yCount));
        dblGocChay = ChinhGocChay(rnd.nextInt(360));
        tgchay = 1500+100*rnd.nextInt(100);
    }

    //định nghĩa hàm giả lập công việc tính toán của thread
    void MySleep(long count)
    {
        long i, j, k = 0;
        for (i = 0; i < count; i++)
            for (j = 0; j < 64000; j++) k = k + 1;
    }
}
```

```

//tác vụ chức năng của thread
public void run() {
    int x1, y1;
    int x2, y2;
    int x, y;
    boolean kq = true;
    try {
        while (fstart) { //lặp trong khi chưa có yêu cầu kết thúc
            //xác định tọa độ hiện hành của thread
            x1 = Pos.x; y1 = Pos.y;
            //hiển thị icon của thread ở (x1,y1)
            frm.gh.drawImage(Pic, xCell * x1, yCell * y1, null);
            //xác định màu vẽ hướng chạy của icon
            Color c = new Color(Pic.getRGB(1, 1), true);
            int yR, yG, yB;
            if (c.getRed() > 128) yR = 0; else yR = 255;
            if (c.getGreen() > 128) yG = 0; else yG = 255;
            if (c.getBlue() > 128) yB = 0; else yB = 255;
            frm.gh.setColor(new Color(yR, yG, yB));
            if (tx >= 0 && ty >= 0) { //hiện mũi tên góc dưới phải
                x = xCell * x1 + xCell - 2;
                y = yCell * y1 + yCell - 2;
                frm.gh.drawLine(x, y, x - 10, y);
                frm.gh.drawLine(x, y, x, y - 10);
            }
            else if (tx >= 0 && ty < 0) { //hiện mũi tên góc trên phải
                x = xCell * x1 + xCell - 2;
                y = yCell * y1 + 2;
                frm.gh.drawLine(x, y, x - 10, y);
                frm.gh.drawLine(x, y, x, y + 10);
            }
            else if (tx < 0 && ty >= 0) { //hiện mũi tên góc dưới trái
                x = xCell * x1 + 2;
                y = yCell * y1 + yCell - 2;
                frm.gh.drawLine(x, y, x + 10, y);
                frm.gh.drawLine(x, y, x, y - 10);
            }
            else { //hiện mũi tên góc trên trái
                x = xCell * x1 + 2;
                y = yCell * y1 + 2;
                frm.gh.drawLine(x, y, x + 10, y);
                frm.gh.drawLine(x, y, x, y + 10);
            }
            //giả lập thực hiện công việc của thread
            MySleep(tgchay);
            //xác định vị trí mới của thread
            HieuchinhVetri();
            x2 = Pos.x; y2 = Pos.y;
            //Xóa icon ở vị trí cũ
            frm.gh.drawImage(frm.blackPic, xCell * x1, yCell * y1, null);
        }
    }
}

```

```

        if (kq == false && fstart == false) { //xóa thread
            this.stop();
            this.fstop = true;
            return;
        }
    }
}
catch (Exception e) { this.stop(); }
//xóa icon của thread trước khi ngừng
x1 = Pos.x; y1 = Pos.y;
frm.gh.drawImage(frm.blackPic, xCell * x1, yCell * y1, null);
//dùng Thread
fstop = true;
stop();
}

//=====
//Hiệu chỉnh góc chạy của thread
//để tránh các trường hợp thread chạy thẳng đứng hay ngang
//=====
double ChinhGocChay(double dblGocChay) {
    double goc = dblGocChay;
    if (0 <= goc && goc < 90) return 45;
    if (90 <= goc && goc < 180) return 135;
    if (180 <= goc && goc < 270) return 225;
    if (270 <= goc) return 315;
    return goc;
}

//=====
//Tính góc phản xạ mới khi thread đụng thành đứng (bên trái hay phải).
//=====
double DoiGocChayX(double dblGocChay) {
    double goc;
    if (dblGocChay > 0 && dblGocChay < 180) goc = 180 - dblGocChay;
    else goc = 180 + 360 - dblGocChay;
    return ChinhGocChay(goc);
}

//=====
//Tính góc phản xạ mới khi thread đụng thành ngang (trên hay dưới).
//=====
double DoiGocChayY(double dblGocChay) {
    return ChinhGocChay(360 - dblGocChay);
}

//=====
//Hiệu chỉnh vị trí của thread
//=====
public void HieuchinhVetri() {
    int x, y;
    x = Pos.x;
    y = Pos.y;

```

```

if (x == 0 || x == xCount - 1 || y == 0 || y == yCount - 1) {
    //icon dừng thành ngang hay dọc -> thay đổi góc chạy
    if (x == 0 || x == xCount - 1) dblGocChay = DoiGocChayX(dblGocChay);
    else if (y == 0 || y == yCount - 1) dblGocChay = DoiGocChayY(dblGocChay);
}
//Hiệu chỉnh tọa độ x của thread
tx = 2 * Math.cos(dblGocChay * PI / 180);
x = x + (int)tx;
if (x < 0) x = 0;
else if (x >= xCount) x = xCount - 1;
//Hiệu chỉnh tọa độ y của thread
ty = 2 * Math.sin(dblGocChay * PI / 180);
y = y + (int)ty;
if (y < 0) y = 0;
else if (y >= yCount) y = yCount - 1;
//chỉnh góc chạy khi dừng 1 trong 4 góc
if (x == 0 && y == 0) //góc trên trái
    ChinhGocChay(dblGocChay + 45);
else if (x == 0 && y == yCount - 1) //góc dưới trái
    ChinhGocChay(dblGocChay + 45);
else if (x == xCount - 1 && y == 0) //góc trên phải
    ChinhGocChay(dblGocChay + 45);
else if (x == xCount - 1 && y == yCount - 1) //góc dưới phải
    ChinhGocChay(dblGocChay + 45);
//Lưu vị trí mới
Pos.x = (int)x;
Pos.y = (int)y;
}
}

```

6. Trong cửa sổ quản lý Project, làm hiển thị chi tiết package nbthreaddemo2 (hiện đang chứa file miêu tả chương trình NBThreadDemo2.java). Ấn phải chuột trên folder nbthreaddemo2, chọn chức năng New.JFrame Form để máy hiển thị cửa sổ "New JFrame Form", nhập tên Form mới là MainForm, click chuột vào button Finish để máy tạo ra Frame tương ứng, cửa sổ thiết kế Form sẽ hiển thị.
7. Click chuột vào icon Source nằm ở phía trên trái của sổ thiết kế form để hiển thị cửa sổ soạn mã nguồn cho form. Viết code cho form như sau :

```

package nbthreaddemo2;
//import các package cần dùng
import java.util.*;
import java.awt.*;
import java.net.*;
import javax.imageio.*;
import javax.swing.*;
import java.util.concurrent.*;

//định nghĩa form ứng dụng
public class MainForm extends javax.swing.JFrame {
    //định nghĩa các thuộc tính cần dùng
    public Image blackPic;

```

```

MyThread[] threadLst;
final int xCell = 30;
final int yCell = 30;
final int xCount = 25;
final int yCount = 20;
Graphics gh;
//tạo đối tượng sinh số ngẫu nhiên
public Random rnd = new Random();
//tác vụ khởi tạo form
public MainForm() {
    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    //thiết lập tác vụ xử lý sự kiện ấn phím trên form
    addKeyListener(new java.awt.event.KeyAdapter() {
        public void keyPressed(java.awt.event.KeyEvent evt) {
            formKeyPressed(evt);
        }
    });
    //thiết lập lại kích thước form theo yêu cầu
    this.setSize(xCount * 30, yCount * 30);
    this.setLocation(0, 0);
    //thiết lập màu nền đen cho form
    this.getContentPane().setBackground(Color.BLACK);
    //đọc bitmap miêu tả cell nền từ file
    URL url;
    try {
        url = this.getClass().getClassLoader().getResource
            ("images/Black.jpg");
        blackPic = ImageIO.read(url);
    } catch (Exception e) {}
    //tạo danh sách chứa 26 thread từ A-Z
    threadLst = new MyThread[26];
    int i;
    //Lặp thiết lập trạng thái ban đầu cho 26 thread từ A-Z
    for (i = 0; i < 26; i++) {
        threadLst[i] = new MyThread(rnd, xCount, yCount, this);
        threadLst[i].fstop = threadLst[i].fsuspend = threadLst[i].fstart = false;
        char c = (char)(i + 65);
        try { //đọc bitmap miêu tả thread c từ file
            url = this.getClass().getClassLoader().getResource
                ("images/Image" + c + ".jpg");
            threadLst[i].Pic = ImageIO.read(url);
        } catch (Exception e) {
            System.out.println(e.toString());
        }
    }
}
//tác vụ xử lý việc ấn phím
private void formKeyPressed(java.awt.event.KeyEvent evt) {
    //lưu đối tượng Graphics của form để vẽ icon khi cần
    gh = this.getGraphics();
}

```

```

//xác định mã phím ấn, nếu không phải từ A-Z thì phớt lờ
int newch = evt.getKeyCode();
if (newch < 0x41 || newch > 0x5a) return;
//xác định chức năng mà user muốn và thực hiện
if (evt.isControlDown() && evt.isShiftDown()) { //dùng Thread
    threadLst[newch - 65].fstart = false;
}
else if (evt.isControlDown()) { //giảm độ ưu tiên tối thiểu
    threadLst[newch - 65].setPriority(1);
}
else if (evt.isControlDown() && evt.isAltDown()) { //tạm dừng thread
    if (threadLst[newch - 65].fstart && !threadLst[newch - 65].fsuspend)
    {
        threadLst[newch - 65].suspend();
        threadLst[newch - 65].fsuspend = true;
    }
}
else if (evt.isAltDown()) { //cho thread chạy lại
    if (threadLst[newch - 65].fstart && threadLst[newch - 65].fsuspend)
    {
        threadLst[newch - 65].resume();
        threadLst[newch - 65].fsuspend = false;
    }
}
else if (evt.isShiftDown()) { //tăng độ ưu tiên tối đa
    threadLst[newch - 65].setPriority(31);
}
else { //tạo mới thread và bắt đầu chạy
    if (!threadLst[newch - 65].fstart) {
        threadLst[newch - 65].fstart = true;
        threadLst[newch - 65].fsuspend = false;
        threadLst[newch - 65].start();
    }
}
}
}
}

```

8. 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.
9. Nếu chương trình hết lỗi, cửa sổ chương trình sẽ hiển thị với cửa sổ có nền đen.
10. Hãy tạo thử từng thread (gõ phím chữ tương ứng) và quan sát quỹ đạo chạy và tốc độ chạy của chúng. Quan sát hiện tượng đè nhau giữa các thread ngày càng nhiều khi số thread tăng lên.
11. Hãy thử thực hiện 1 số chức năng quản lý thread bằng cách nhập tổ hợp phím tương ứng như qui định trong mục II. Nội dung.