package com.java;

import java.awt.FlowLayout;

import java.awt.Font;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.BufferedInputStream;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.IOException;

import java.io.InputStream;

import java.io.RandomAccessFile;

import java.net.HttpURLConnection;

import java.net.URL;

import java.net.URLConnection;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JTextField;

public class DownMultiThreadFrame extends JFrame implements ActionListener{

    private static final long serialVersionUID = 9055094226537574068L;

    public DownMultiThreadFrame(){

        panel.setLayout(new FlowLayout());

        label1.setFont(new Font("雅黑", Font.BOLD, 15));

        panel.add(label1);

        panel.add(label2);

        panel.add(urlField);

        panel.add(StartButton);

        panel.add(resetButton);

        panel.add(exitButton);

        setContentPane(panel);

        StartButton.addActionListener(this);

        resetButton.addActionListener(this);

        exitButton.addActionListener(this);

        setSize(400, 400);

        setVisible(true);

        setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

    }

    public static void main(String[] args){

        new DownMultiThreadFrame();

    }

    private final JPanel panel = new JPanel();

    private final JLabel label1 = new JLabel("网络资源的多线程下载：");

    private final JLabel label2 = new JLabel("网络资源的网址：");

    JButton StartButton = new JButton("点击开始下载");

    JButton resetButton = new JButton("清空");

    JButton exitButton = new JButton("退出");

    JTextField urlField = new JTextField(20);

    @Override

    public void actionPerformed(ActionEvent e){

        if(e.getSource() == StartButton){

            if("".equals(urlField.getText())){

                JOptionPane.showMessageDialog(this, "请输入资源地址");

            }

            String url = urlField.getText();

            int pos = url.lastIndexOf("/");

            String fileName = url.substring(pos + 1);

            try{

                download(url, "d:\\" + fileName, 2);

            }catch(Exception e1){

                e1.printStackTrace();

            }

        }else if(e.getSource() == resetButton){

            urlField.setText("");

        }else{

            System.exit(0);

        }

    }

    /\*\*

     \* 下载网络资源

     \* \*/

    public void download(String url, String dest, int threadNum)

            throws Exception{

        URL downURL = new URL(url);

        // 打开网络连接

        HttpURLConnection conn = (HttpURLConnection) downURL.openConnection();

        // 创建文件长度的变量

        long fileLength = -1;

        // 获得连接状态标记代码

        int stateFlagCode = conn.getResponseCode();

        if(stateFlagCode == 200){

            // 网络连接正常

            fileLength = conn.getContentLength();

            // 取笑网络连接

            conn.disconnect();

        }

        if(fileLength > 0){

            // 计算每个线程的字节数

            long byteCounts = fileLength / threadNum + 1;

            File file = new File(dest);

            int i = 0;

            while(i < threadNum){

                long startPosition = byteCounts \* i;

                long endPosition = byteCounts \* (i + 1);

                if(i == threadNum - 1){

                    DownMulltiThread fileThread = new DownMulltiThread(url,

                            file, startPosition, 0);

                    new Thread(fileThread).start();

                }else{

                    DownMulltiThread fileThread = new DownMulltiThread(url,

                            file, startPosition, endPosition);

                    new Thread(fileThread).start();

                }

                i++;

            }

        }

    }

}// end class

/\*\*

 \* 用于实现网络资源的下载

 \* \*/

class DownMulltiThread implements Runnable{

    public DownMulltiThread(){

    }

    /\*\*

     \* @param sUrl

     \*            网络资源位置

     \* @param desFile

     \*            需要写入的文件对象

     \* @param startPos

     \*            写入的开始位置

     \* @param endPos

     \*            写入文件的结束位置

     \* \*/

    public DownMulltiThread(String sUrl, File desFile, long startPos,

            long endPos){

        this.desFile = desFile;

        this.sUrl = sUrl;

        this.startPos = startPos;

        this.endPos = endPos;

    }

    private String sUrl = "";

    private File desFile;

    private long startPos;

    private long endPos;

    @Override

    public void run(){

        RandomAccessFile out;

        try{

            out = new RandomAccessFile(desFile, "rw");

            out.seek(startPos);

            URL url = new URL(sUrl);

            URLConnection urlcon = url.openConnection();

            urlcon.connect();

            InputStream in = urlcon.getInputStream();

            BufferedInputStream buf = new BufferedInputStream(in);

            byte[] bytes = new byte[1024];

            int len = buf.read(bytes);

            long diffLength=endPos-startPos;

            while(len != -1){

                out.write(bytes, 0, len);

                diffLength-=1024;

                if(diffLength<=0){

                    break;

                }

                len = buf.read(bytes);

            }

        }catch(FileNotFoundException e){

            e.printStackTrace();

        }catch(IOException e){

            e.printStackTrace();

        }

    }

}