

LẬP TRÌNH TRUY XUẤT HỆ THỐNG TẬP TIN (IO STREAM)

PGS. TS. Huỳnh Công Pháp

I. Tóm tắt lý thuyết

- Java cung cấp gói java.io gồm tập hợp các lớp và interface cho phép thao tác với hệ thống tệp tin.
- Lớp File và FileDescriptor được dùng để lấy thông tin và thao tác với hệ thống tệp tin và thư mục.
- Luồng (stream) được hiểu như một đường ống nối từ chương trình ra bên ngoài cho phép chương trình ghi hoặc đọc dữ liệu với các thiết bị (như màn hình, máy in, ổ đĩa).
- Các luồng InputStream và OutputStream dùng để đọc và ghi dữ liệu dạng bytes.
- Các luồng Reader và Writer dùng để đọc và ghi dữ liệu dạng ký tự.
- FileInputStream và FileOutputStream là hai luồng chính để đọc và ghi dữ liệu dạng bytes với tệp tin.
- FileReader và FileWriter là hai luồng cho phép đọc và ghi dữ liệu dạng ký tự với tệp tin.

II. Bài tập thực hành mẫu

1. Sử dụng lớp File, viết chương trình đọc và in ra màn hình nội dung một thư mục (tên các tệp tin hoặc thư mục con)

```
/*Cần import gói java.io để truy xuất hệ thống tệp tin và thư mục*/
import java.io.*;

public class ReadDir {

    public ReadDir(String path)
    {

        /*Trở đến thư mục có đường dẫn path*/
        File f = new File(path);

        /*Lấy danh sách tên tệp tin hoặc thư mục mà f đang trở đến*/
        String[] filenames = f.list();

        /*Duyệt danh sách và in tên ra màn hình*/
        for (int i=0;i<filenames.length;i++)
            System.out.println(filenames[i]);
    }
}
```

```

public static void main(String[] arg)
{
    /*đường dẫn là thư mục hiện hành*/
    new ReadDir("./");
}
}

```

Kết quả chương trình:

```

<terminated> ReadDir [Java Application] C:\Program Files (x86)\Java\jre8\bin\javaw.exe (Nov 12, 2013, 4:20:49 PM)
Bai_tap_thuc_hanh_Java_IO.JDBC.doc
Bai_tap_thuc_hanh_Java_IO.JDBC.pdf
Bai_Tap_Thuc_Hanh
Basic Java
Core Java (Starting With Java)
Core Java (Starting With Java).rar
CoreJava
Examples
Slides_tiang_Viet

```

2. Sử dụng lớp File, viết chương trình đếm số lượng tệp tin có trong một thư mục (kể cả các tệp tin có trong các thư mục con)

```

import java.io.File;
public class CountFolder {

    public static int count(String path)
    {
        /*Dùng để đếm số lượng tệp tin*/
        int c =0;

        File f = new File(path);

        String[] filenames = f.list();

        for (int i=0;i<filenames.length;i++)
        {
            /*Tạo fi để kiểm tra phần tử hiện tại là tệp hay thư mục*/
            File fi = new File(path+"\""+filenames[i]);

            /*Nếu là tệp tin, tăng đếm lên 1. */
            if (fi.isFile()) c++;

            else /*nếu là thư mục gọi lại hàm đếm thư mục*/
                c+=count(fi.getAbsolutePath());
        }

        /*Trả lại kết quả đếm được*/
        return c;
    }
    public static void main(String[] arg)
    {

```

```

        /*Gọi hàm đếm và in ra kết quả của thư mục hiện hành*/
        System.out.println(count("./"));
    }
}

```

Kết quả chương trình:

```

Problems @ Javadoc Declaration Console Console
<terminated> CountFolder [Java Application] C:\Program Files (x86)\Java\jre8\bin\javaw.exe (Nov 12, 2013, 4:28:37 PM)
229
|

```

3. Viết chương trình đếm số lượng ký tự 'i' có trong một tệp tin.

```

import java.io.FileInputStream;
import java.io.IOException;
public class CountCharFile
{
    public static int count(String path, char ch)
    {
        /*Dùng làm biến đếm*/
        int c=0;

        /*Dùng đọc nội dung tệp tin*/
        FileInputStream fis = null;

        try
        {
            /* Mở tệp tin cần đọc nội dung*/
            fis = new FileInputStream(path);

            /*Đọc dữ liệu từ tệp và kiểm tra, kết thúc nhận được mã -1*/
            int data;
            data = fis.read();

            while(data!=-1)
            {
                if (data == ch) c++;
                data = fis.read();
            }
        }
        catch(IOException e)
        {
            System.out.println("Lỗi IOException");
        }
        finally
        {
            try
            {
                fis.close();
            }
            catch(IOException e)
            {
                System.out.println("Lỗi IOException");
            }
        }
    }
}

```

```

    }
    }
    return c;
}
public static void main(String args[])
{
    /*Thử kiểm tra tệp chương trình hiện tại*/
    System.out.println(count("./src/CountCharFile.java",'i' ));
}
}

```

Kết quả chương trình:

```

<terminated> CountCharFile (1) [Java Application] C:\Program Files\Java\jdk1.7.0_4
40

```

4. Viết chương trình hiển thị nội dung tệp tin ra màn hình.

Đáp án 1: FileInputStream

```

import java.io.FileInputStream;
import java.io.IOException;
public class ReadFile
{
    public static void readFile(String path)
    {
        FileInputStream fis = null;
        try
        {
            /*Mở tệp tin cần đọc*/
            fis = new FileInputStream(path);

            /*Đọc nội dung và hiển thị*/
            int data;
            data = fis.read();
            while(data!=-1)
            {
                System.out.print((char)data);
                data = fis.read();
            }
        }
        catch(IOException e)
        {
            System.out.println("Lỗi IOException");
        }
        finally
        {
            try
            {
                fis.close();
            }
        }
    }
}

```

```

        catch(IOException e)
        {
            System.out.println("Lỗi IOException");
        }
    }
    public static void main(String args[])
    {
        readFile("./src/ReadFile.java");
    }
}

```

Đáp áp 2: Sử dụng FileReader

```

import java.io.*;
class FileReaderDemo
{
    public static void main(String args[]) throws Exception
    {
        /*Sử dụng Loại Luồng làm việc với tệp ký tự FileReader*/
        FileReader fr = new FileReader("./src/ReadFile.java");

        /*Sử dụng Luồng trung gian để đọc cả dòng*/
        BufferedReader br = new BufferedReader(fr);

        /*Đọc từng dòng rồi in ra màn hình*/
        String s;
        while((s = br.readLine()) != null) {
            System.out.println(s);
        }
        fr.close();
    }
}

```

Kết quả chương trình:

```

<terminated> ReadFile (1) [Java Application] C:\Program Files\Java\jdk1.7.0_45\jre\bin
import java.io.FileInputStream;
import java.io.IOException;
public class ReadFile
{
    public static void readFile(String path)
    {
        FileInputStream fis = null;
        try
        {
            fis = new FileInputStream(path);
            int data;
            data = fis.read();
            while(data!=-1)
            {
                System.out.print((char)data);
                data = fis.read();
            }
        }
        catch(IOException e)
        {
            System.out.println("Lá»i IOException");
        }
        finally
        {
            try
            {
                fis.close();
            }
            catch(IOException e)

```

5. Viết chương trình tạo giao diện gồm 1 Textarea và một Button có tên "Open..". Khi người dùng ấn nút "Open..", chương trình hiển thị cửa sổ cho phép người dùng chọn một tệp để hiển thị nội dung trong Textarea.

```

import java.io.*;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class OpenFile extends JFrame implements ActionListener
{
    /*Tạo các thành phần trên giao diện*/
    JTextArea ta = new JTextArea();
    JButton open = new JButton("Open..");

    /*Sử dụng Lớp JFileChooser cho phép mở hộp thoại mở hệ thống tệp tin*/
    JFileChooser chooser;

    /*Sử dụng các Lớp này để đọc nội dung tệp tin */
    FileReader fr;
    BufferedReader br;

    public OpenFile()
    {

    }

    /*Thêm các thành phần vào tầng ContentPane của JFrame*/
    Container cont = this.getContentPane();

    open.addActionListener(this);
    cont.add(ta);

```

```

        cont.add(open,"South");

        this.setSize(400,400);
        this.setVisible(true);
    }

    /*Hàm này được thực hiện khi ấn nút Open*/
    public void actionPerformed(ActionEvent e) {

        /* Tạo đối tượng JFileChooser để hiển thị hộp thoại mở tệp tin*/
        chooser = new JFileChooser();

        /*Mở thư mục hiện hành*/
        chooser.setCurrentDirectory(new java.io.File("."));

        /*Tiêu đề của hộp thoại*/
        chooser.setDialogTitle("Open file");

        /*Nếu đã chọn tệp tin*/
        if (chooser.showOpenDialog(this) == JFileChooser.APPROVE_OPTION) {

            /*Lấy tên tệp tin đã chọn*/
            String filename =chooser.getSelectedFile().getAbsolutePath();

            /*Đọc nội dung tệp và hiển thị ở TextArea*/
            readFile(filename);

        }
    }

    /*Đọc nội dung tệp và hiển thị ở TextArea*/
    public void readFile(String filename)
    {
        try
        {

            /*Xoa hết noi dung hien co o Textarea*/
            ta.setText("");

            /*Mở tệp và đọc*/
            fr = new FileReader(filename);
            br = new BufferedReader(fr);
            String s;
            while((s = br.readLine()) != null)
            {

                /*Chèn thêm nội dung vào TextArea*/
                ta.append(s+"\n");

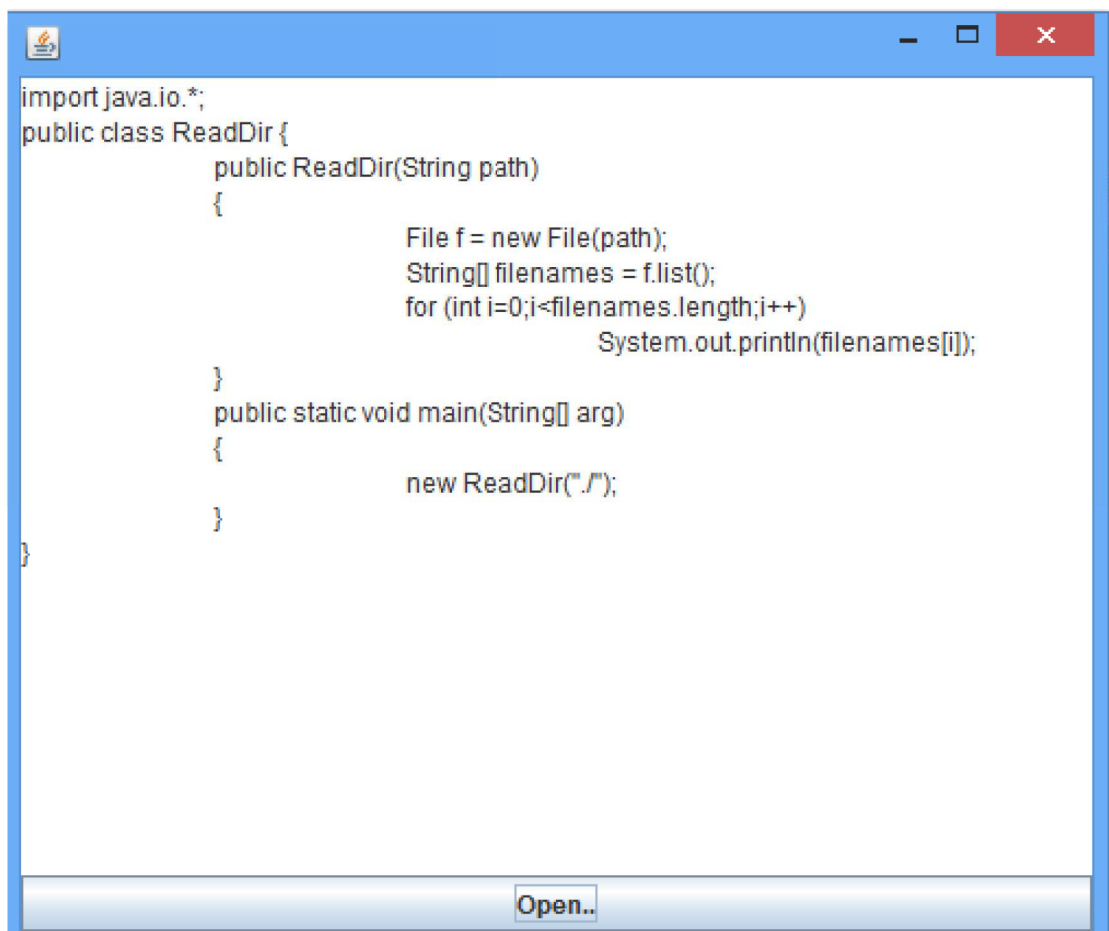
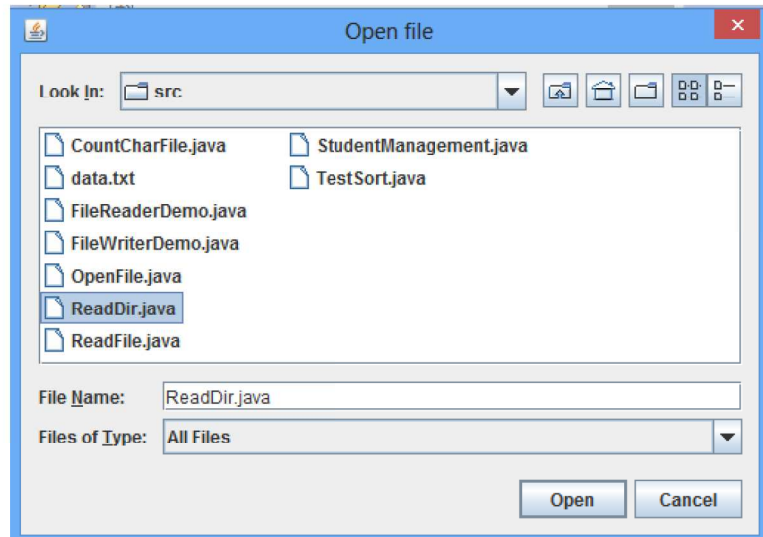
            }
            fr.close();

        }catch(Exception e)
        {
            e.printStackTrace();
        }
    }
}

```

```
public static void main(String[] a)
{
    new OpenFile();
}
```

Kết quả chương trình:



- Viết chương trình ghi nội dung *"programming with IO Stream in java"* xuống một tệp tin trên ổ đĩa.

Đáp án 1: Sử dụng FileOutputStream

```
import java.io.FileOutputStream;
import java.io.IOException;
public class WriteFile
{
    public static void writeFile(String path)
    {
        /* Dùng để ghi dữ liệu lên tệp tin*/
        FileOutputStream fos = null;

        /*Chuỗi dữ liệu cần ghi*/
        String msg = "programming with IO Stream in java";
        try
        {
            /*Mở tệp tin cần ghi*/
            fos = new FileOutputStream(path);

            /*Ghi từng ký tự của chuỗi lên tệp tin*/
            for (int i=0;i<msg.length();i++)
                fos.write((int)msg.charAt(i));

            /* Đẩy dữ liệu còn ở bộ đệm xuống tệp*/
            fos.flush();
        }
        catch(IOException e)
        {
            System.out.println("Lỗi IOException");
        }
        finally
        {
            /*Khởi finally Luôn Luôn thực hiện*/
            try
            {
                fos.close();
            }
            catch(IOException e)
            {
                System.out.println("Lỗi IOException");
            }
        }
    }

    public static void main(String args[])
    {
        writeFile("D:\\Test.txt");
    }
}
```

Đáp án 2: Sử dụng FileWriter

```
import java.io.*;
```

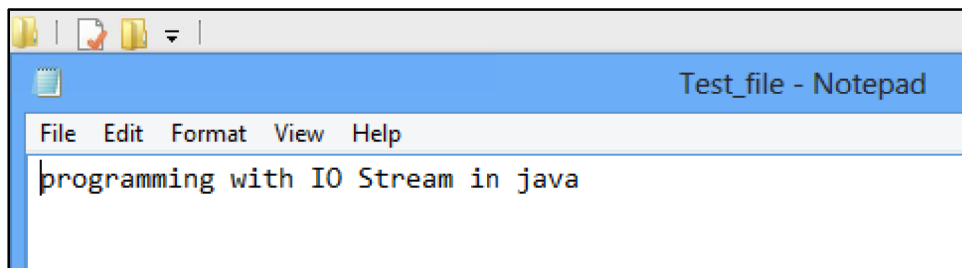
```

class FileWriterDemo {
public static void main(String args[]) throws Exception
{
String source = "programming with IO Stream in java";

/*Sử dụng Lớp FileWriter để mở và ghi dữ liệu*/
FileWriter f = new FileWriter("./Test_file.txt");
f.write(source);
f.flush();
f.close();
}
}

```

Kết quả chương trình:



7. Cải tiến đáp án 1 ở bài tập 6, cho phép chương trình ghi cả một chuỗi xuống tệp tin.

```

import java.io.*;
import java.io.IOException;
public class WriteFile
{
    public static void writeFile(String path)
    {
        /*Sử dụng Luồng trung gian cho ghi cả chuỗi xuống tệp*/
        DataOutputStream bf=null;
        String msg = "programming with IO Stream in java";
        try
        {
            /*Nối FileOutputStream vào Luồng DataOutputStream*/
            bf = new DataOutputStream(new FileOutputStream(path));

            /*Ghi cả dòng xuống tệp*/
            bf.writeChars(msg);

            /*Đẩy dữ liệu còn ở bộ đệm xuống tệp*/
            bf.flush();
        }
        catch(IOException e)
        {
            System.out.println("Lỗi IOException");
        }
        finally
        {
            try
            {

```

```

        bf.close();
    }
    catch(IOException e)
    {
        System.out.println("Lỗi IOException");
    }
    }
}
public static void main(String args[])
{
    writeFile("D:\\Test.txt");
}
}

```

8. Viết chương trình tạo giao diện gồm 1 Textarea và một Button có tên "Save as..". Khi người dùng ấn nút "Save as..", chương trình hiển thị cửa sổ cho phép người dùng đặt tên tệp để lưu nội dung của Textarea xuống ổ đĩa.

```

import java.io.*;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class SaveFile extends JFrame implements ActionListener
{
    JTextArea ta = new JTextArea();
    JButton saveAs = new JButton("Save as..");

    /*Sử dụng Lớp JFileChooser để hiển thị hộp thoại Lưu tệp tin*/
    JFileChooser chooser;

    /*Sử dụng Lớp FileWriter để ghi nội dung xuống tệp*/
    FileWriter f;

    public SaveFile()
    {
        /*Tạo giao diện có nút Save as và TextArea*/
        Container cont = this.getContentPane();

        saveAs.addActionListener(this);
        cont.add(ta);
        cont.add(saveAs, "South");

        this.setSize(400,400);
        this.setVisible(true);
    }

    /*Hàm này thực thi khi ấn nút Save as..*/
    public void actionPerformed(ActionEvent e) {

        /*Tạo đối tượng JFileChooser để mở hộp thoại ghi tệp*/
        chooser = new JFileChooser();
    }
}

```

```

/*Thư mục hiện hành được mở */
    chooser.setCurrentDirectory(new java.io.File("."));

/*Tiêu đề của hộp thoại*/
    chooser.setDialogTitle("Save as..");

    if (chooser.showSaveDialog(this) == JFileChooser.APPROVE_OPTION) {

/*Lấy tên tệp đã đặt hoặc chọn*/
        String filename = chooser.getSelectedFile().getAbsolutePath();

        saveFile(filename);

    }
}

/*Hàm đọc nội dung từ TextArea rồi ghi xuống tệp tin*/
public void saveFile(String filename)
{
    try
    {
        String content = ta.getText();

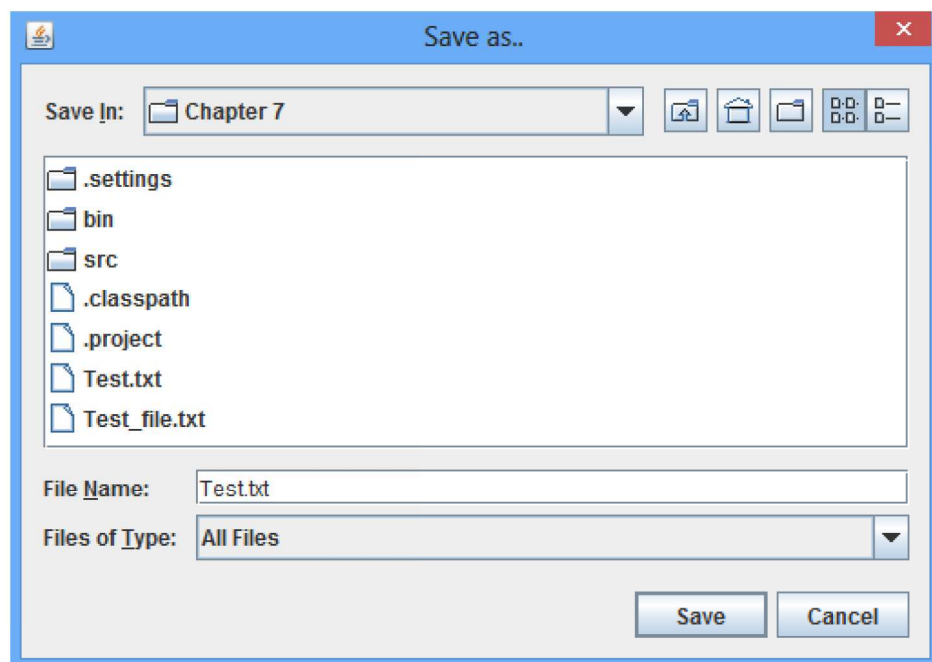
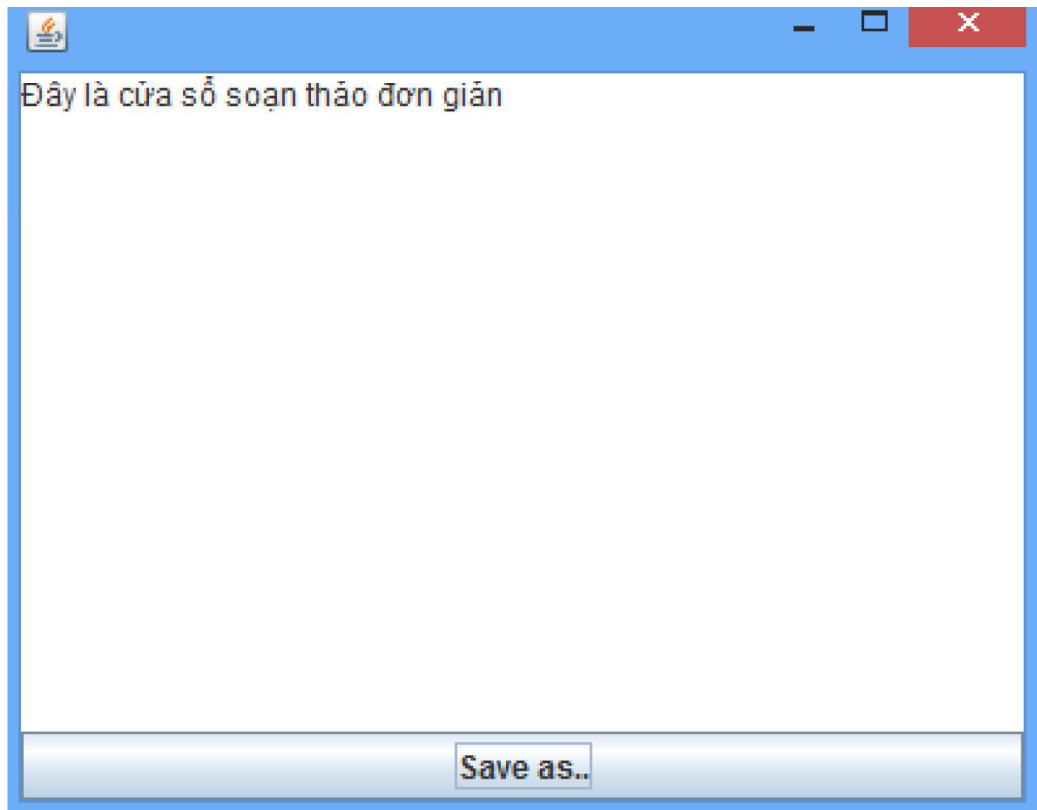
        f = new FileWriter(filename);
        f.write(content);
        f.flush();
        f.close();

    }catch(Exception e)
    {
        e.printStackTrace();
    }
}

public static void main(String[] a)
{
    new SaveFile();
}
}

```

Kết quả chương trình:



9. Viết chương trình sao chép nội dung tệp tin từ thư mục này sang tệp tin ở thư mục khác.

```
import java.io.*;
```

```
/*Copy tệp tin là mở tệp nguồn để đọc nội dung vào chương trình và từ chương trình ghi nội dung đọc được vào tệp đích*/
```

```
public class CopyFile
```

```

{
    public static void main(String args[])
    {
        /* fis dùng để đọc nội dung tệp, fos dùng để ghi nội dung lênh tệp*/
        FileInputStream fis = null;
        FileOutputStream fos = null;

        try
        {
            /*Mở tệp để đọc và ghi*/
            fis = new FileInputStream("./src/CopyFile.class");
            fos = new FileOutputStream("./CopyFile1.class");

            /*Đọc dữ liệu từ tệp fis và ghi xuống tệp fos*/
            int data;
            data = fis.read();
            while(data!=-1)
            {
                fos.write(data);
                data = fis.read();
            }
            fos.flush();
        }
        catch(IOException e)
        {
            System.out.println("Lỗi IOException");
        }
        finally
        {
            try
            {
                fos.close();
                fis.close();
            }
            catch(IOException e)
            {
                System.out.println("Lỗi IOException");
            }
        }
    }
}

```

10. Viết chương trình quản lý sinh viên có giao diện đơn giản. Chương trình gồm các chức năng cơ bản:

- Tạo giao diện cho phép nhập danh sách sinh viên;
- Lưu danh sách sinh viên xuống tệp tin khi chương trình kết thúc;
- Hiển thị danh sách sinh viên theo chiều tăng dần của điểm trung bình;
- Tìm kiếm sinh viên theo tên.

Giả sử mỗi sinh viên được mô tả gồm 3 thuộc tính: tên, ID và điểm trung bình.

```

import java.awt.*;
import java.awt.event.*;
import java.io.*;
import java.util.*;

public class StudentManagement extends Frame implements ActionListener
{
    /*dung he thong menu de chua cac chuc nang chuong trinh*/
    private MenuBar bar;
    private Menu action;
    private MenuItem input, search, sort_view, exit;

    FileWriter f ;
    PrintWriter pr;

    /*dung de chua danh sach sinh vien*/
    Vector list = new Vector();

    public StudentManagement(String s)
    {
        super(s);

        /*doc du lieu tu tep tin va tao danh sach sinh vien*/
        loadData();

        /*tao he thong menu*/
        bar = new MenuBar();
        setMenuBar(bar);
        action = new Menu("Action");
        bar.add(action);
        action.addActionListener(this);
        input = new MenuItem("Input");
        action.add(input);
        input.addActionListener(this);

        action.addSeparator();
        sort_view = new MenuItem("Sort and View");
        action.add(sort_view);
        sort_view.addActionListener(this);

        action.addSeparator();
        search = new MenuItem("Search");
        action.add(search);
        search.addActionListener(this);

        action.addSeparator();
        exit = new MenuItem("Exit");
        action.add(exit);
        exit.addActionListener(this);

        /*Thiet lap kich thuoc va hien thi cua so chinh*/
        setSize(400,300);
        setVisible(true);
        setResizable(false);
    }
    public void actionPerformed(ActionEvent ae)
    {

```

```

if(ae.getActionCommand().equals("Exit"))
{
    /*Truoc khi ket thuc Luu dl xuong tep tin*/
    saveFile();
    System.exit(0);
}
if(ae.getActionCommand().equals("Input"))
{
    /*Tao cua so nhap thong tin sinh vien*/
    InputForm f = new InputForm(this, "Input information");
}
if(ae.getActionCommand().equals("Sort and View"))
{
    /*Goi ham sap xep sinh vien theo chieu tang danh cua aver*/
    sort();
}
if(ae.getActionCommand().equals("Search"))
{
    /*Tao cua so tim kiem sinh vien theo ten*/
    new SearchForm("Search", list);
}

}
/*Cap nhat doi tuong sinh vien vao danh sach sau khi chon nut Save o cua so nhap sinh vien*/
public void updateList(String n, String i, float a)
{
    Student st = new Student(n,i,a);
    list.add(st);
}

/*Mo tep tin de ghi moi khi chuong trinh ket thuc*/
public void saveFile()
{
    try{

        f = new FileWriter("./src/data.txt", false);
        pr = new PrintWriter(f);

        Enumeration vEnum = list.elements();

        while(vEnum.hasMoreElements())
        {
            Student st = (Student)vEnum.nextElement();

            /*Gia tri cac thuoc tinh cua mot sinh vien duoc ghi xuong tep thanh 1 chuoai, phan cach nhau bang ki tu &*/
            String toString = st.getName()+"&"+st.getId()+"&"+st.getAver();
            pr.println(toString);
            pr.flush();

        }
    }catch(Exception e)
    {
        e.printStackTrace();
    }
}

```



```

    }
}

/*Mo tep tin de doc va nap vao danh sach ngay khi chuong trinh khoi dong*/
public void loadData()
{
    try{
        FileReader fr = new FileReader("./src/data.txt");

        BufferedReader br = new BufferedReader(fr);
        String s;
        while((s = br.readLine()) != null) {

            /*Thong tin sinh vien doc tu tep tin la 1 chuoì, phan cach nhau bang ki tu
            &. Tach thong tin de tao lai doi tuong Student*/

            String[] data = s.split("&") ;

            Student st=new Student(data[0], data[1], Float.parseFloat(data[2]));

            /*Them vao danh sach*/
            list.add(st);
        }
        catch(Exception e)
        {
            e.printStackTrace();
        }
    }

    /*sap xep danh sach theo chieu tang dan cua diem trung binh
    * su dung ham sort cuar Lop Arrays */
    public void sort()
    {
        /*Tao mang Student tu danh sach de sap xep*/
        Student[] sts = new Student[list.size()];
        int index=0;

        Enumeration vEnum = list.elements();
        while(vEnum.hasMoreElements())
        {
            sts[index] = (Student)vEnum.nextElement();
            index++;
        }

        /*Sap xep theo chieu tang dan cua aver*/
        Arrays.sort(sts);

        /*Hien thi danh sach da sap xep vao TextArea*/

        TextArea ta = new TextArea("Name \t id \t aver\n");
        for(index=0; index < sts.length; index++)
        {

            ta.append(sts[index].getName()+"\t"+sts[index].getId()+"\t"+sts[index].getAver()+"\n");
        }
    }
}

```

```

        this.add(ta);
        this.validate();//cap nhat lai giao dien
    }
    public static void main(String[] args)
    {
        new StudentManagement("Sudent Management");
    }
}

/* Tao Lop de bieu dien sinh vien*/
class Student implements Comparable
{
    private String name;
    private String id;
    private float aver;

    public Student()
    {
        name = new String("");
        id = new String("");
        aver=0;
    }
    public Student(String n, String i, float a)
    {
        name = n;
        id = i;
        aver=a;
    }
    public String getName()
    {
        return name;
    }
    public String getId()
    {
        return id;
    }
    public float getAver()
    {
        return aver;
    }
    public int compareTo(Object other)
    {
        {
            Student otherRect = (Student)other;
            return (int)(this.aver-otherRect.aver);
        }
    }
}

class InputForm extends Frame implements ActionListener
{
    private Label namelb;
    private TextField name;
    private Label idlb;
    private TextField id;
    private Label averlb;
    private TextField aver;
    Button save, newInput, cancel;
    StudentManagement main_frame;
}

```

```

public InputForm(StudentManagement a, String s)
{
    super(s);

    /*Lay doi tuong cua so chinh de goi ham cap nhat thong tin sinh
    vien duoc nhap o cua so nay vao danh sach list o chuong trinh
    chinh*/
    main_frame = a;

    /*Tao giao dien nhap thong tin sinh vien*/

    setLayout(new BorderLayout());
    Panel p1 = new Panel();
    p1.setLayout(new GridLayout(3,2));

    nameLabel=new Label("Name");
    name = new TextField(20);
    p1.add(nameLabel);
    p1.add(name);
    idlb=new Label("Id");
    id = new TextField(20);
    p1.add(idlb);
    p1.add(id);
    averlb=new Label("Average");
    aver = new TextField(20);
    p1.add(averlb);
    p1.add(aver);

    this.add(p1,"North");

    Panel p2 = new Panel();
    save = new Button("Save");
    newInput = new Button("New");
    cancel = new Button("Cancel");
    save.addActionListener(this);
    newInput.addActionListener(this);
    cancel.addActionListener(this);
    p2.add(save);
    p2.add(newInput);
    p2.add(cancel);

    this.add(p2,"South");
    setSize(300,300);
    setVisible(true);
    setResizable(false);
}
public void actionPerformed(ActionEvent ae)
{
    if(ae.getActionCommand().equals("Save"))
    {
        /*truyen va cap nhat sinh vien moi vao danh sach o chuong trinh chinh*/
        main_frame.updateList(name.getText(), id.getText(),
        Float.parseFloat(aver.getText()));
    }
    if(ae.getActionCommand().equals("New"))

```

```

    {
        reset();
    }
    if(ae.getActionCommand().equals("Cancel"))
    {
        this.dispose();
    }
}

public void reset()
{
    name.setText("");
    id.setText("");
    aver.setText("0");
}
}

class SearchForm extends Frame implements ActionListener
{
    private Label nameLb;
    private TextField name;
    private TextArea ta ;

    private Button search, cancel;
    private Vector list;

    public SearchForm(String s, Vector v)
    {
        super(s);

        /*Lay danh sach sinh vien tu chuong trinh chinh*/
        list =v;

        /*Tao giao dien cho cua so tim kiem*/
        setLayout(new BorderLayout());
        Panel p1 = new Panel();
        p1.setLayout(new GridLayout(3,2));
        nameLb=new Label("Name");
        name = new TextField(20);
        p1.add(nameLb);
        p1.add(name);
        this.add(p1,"North");

        ta = new TextArea("Name \t Id \t Aver \n");
        this.add(ta,"Center");

        Panel p2 = new Panel();
        search = new Button("Search");

        cancel = new Button("Cancel");
        search.addActionListener(this);

        cancel.addActionListener(this);
        p2.add(search);

        p2.add(cancel);
    }
}

```

```

        this.add(p2, "South");
        setSize(300, 300);
        setVisible(true);
        setResizable(false);
    }
    public void actionPerformed(ActionEvent ae)
    {
        if(ae.getActionCommand().equals("Search"))
        {
            /*goi ham timkiem ten sinh vien*/
            search(name.getText());
        }

        if(ae.getActionCommand().equals("Cancel"))
        {
            this.dispose();
        }
    }

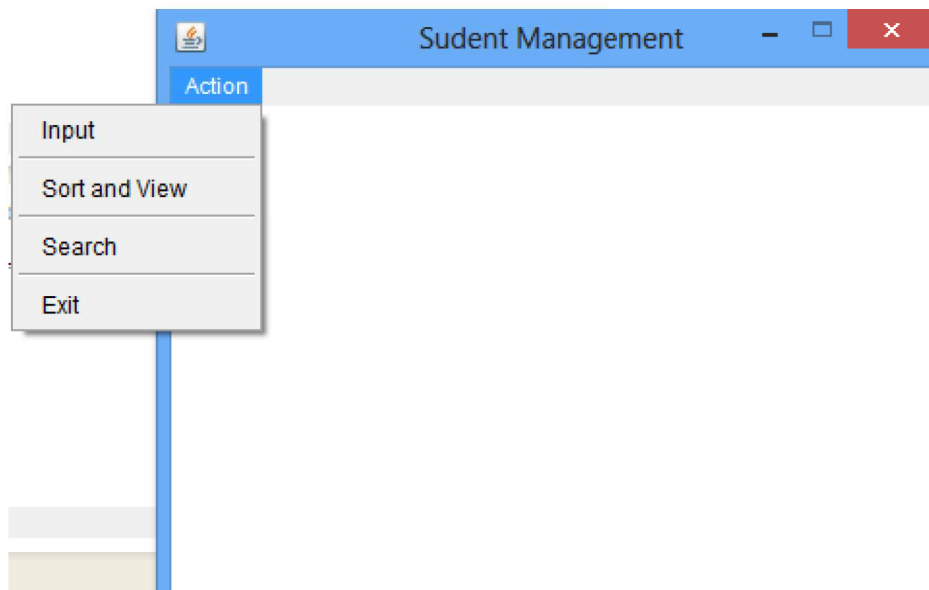
    /*timkiem ten sinh vien tu danh sach co chua chuoi s*/
    public void search(String s)
    {
        Enumeration vEnum = list.elements();

        while(vEnum.hasMoreElements())
        {
            Student st = (Student)vEnum.nextElement();
            if (st.getName().indexOf(s)!=-1)
            {
                ta.append(st.getName()+"\t"+st.getId()+"\t"+st.getAver()+"\n");
            }
        }
        this.validate();
    }
}

```

Kết quả chương trình:

Cửa sổ chính của chương trình



Cửa sổ nhập dữ liệu:

The "Input information" dialog box contains the following data:

Name	hcephap
Id	a123
Average	8

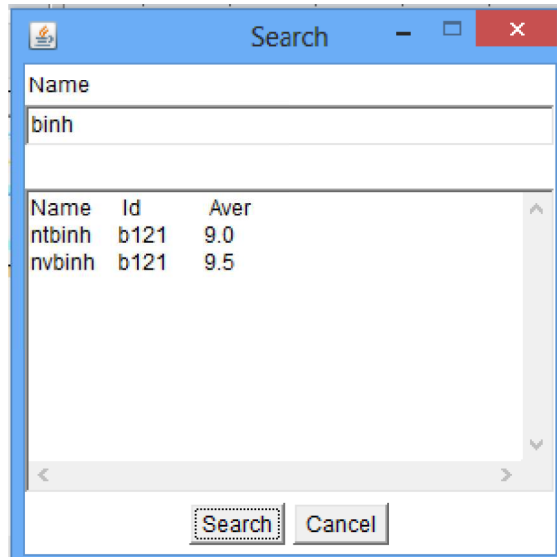
Buttons: Save, New, Cancel

Cửa sổ hiển thị danh sách sinh viên đã được sắp xếp:

The screenshot shows the "Student Management" window with a list of students sorted by average score. The list is as follows:

Name	id	aver
nhung	b234	7.0
hcephap	a123	8.0
nvanh	id	8.5
ntbinh	b121	9.0
nvbinh	b121	9.5

Cửa sổ tìm kiếm sinh viên theo tên:



III. Bài tập tự giải

1. Viết chương trình tính kích thước một thư mục. Gợi ý: tham khảo bài tập 2 và sử dụng hàm `length()` của lớp `File`.
2. Viết chương trình tạo giao diện gồm 4 textfield cho phép nhập thông tin: họ tên, ngày sinh, địa chỉ email, số điện thoại và hai nút có tên "Save" và "Cancel". Khi chương trình chạy lần đầu tiên, giao diện chứa các textfield trống cho phép người dùng nhập dữ liệu và khi ấn nút Save, chương trình sẽ đọc thông tin ở các textfield để ghi xuống tệp tin. Kể từ lần chạy thứ 2, chương trình sẽ đọc thông tin đã ghi ở tệp tin để điền vào các textfield của giao diện chương trình. Nút Cancel dùng để tắt chương trình.
3. Viết chương trình tạo giao diện gồm 1 Textarea và hai nút có tên "Save as.." và "Open". Khi người dùng ấn nút "Open...", chương trình hiển thị cửa sổ cho phép người dùng chọn một tệp để hiển thị nội dung trong Textarea. Khi người dùng ấn nút "Save as...", chương trình hiển thị cửa sổ cho phép người dùng đặt tên tệp để lưu nội dung của Textarea xuống ổ đĩa.
4. Viết chương trình cho phép sao chép nội dung thư mục.
5. Viết chương trình tạo một trình soạn thảo, gồm có 1 TextArea tạo không gian soạn thảo và hệ thống menu gồm 2 mục menu có tên là File và Edit. Trong đó,
 - Mục menu File gồm các mục con:
 - New: Mở mới không gian soạn thảo
 - Save as: Tạo tệp tin và ghi nội dung ở không gian soạn thảo xuống tệp tin
 - Save: Ghi nội dung ở không gian soạn thảo xuống tệp tin đang mở

- Open: Mở hệ thống tệp tin để tìm tệp tin cần đọc nội dung và điền vào không gian soạn thảo.
 - Mục menu Edit gồm các mục con Cut, Copy, Paste lần lượt cho phép cắt, sao chép và dán nội dung đoạn văn bản trong không gian soạn thảo.
6. Viết chương trình quản lý nhân sự của một công ty gồm các loại nhân sự Giám đốc, Quản đốc và nhân viên. Chương trình cung cấp giao diện cho phép thực hiện các công việc sau:
- a. Nhập dữ liệu cho các nhân sự và ghi dữ liệu xuống tệp tin để lưu trữ
 - b. Đọc thông tin từ tệp tin để hiển thị lên giao diện, đồng thời cho phép:
 - Lọc thông tin các nhân sự có tổng lương cao nhất.
 - Lọc tên các nhân sự có sinh nhật trong tháng 12
 - Lọc tất cả nhân viên thuộc một phòng nào đó.
 - Hiển thị nhân sự theo chiều tăng dần của tổng lương
 - Tìm kiếm nhân viên theo tên hoặc năm sinh.

Gợi ý: Tạo các lớp nhân sự gồm các thuộc tính như bên dưới:

- Giám đốc gồm các thuộc tính: họ tên, ngày sinh, hệ số lương, hệ số chức vụ;
- Quản đốc gồm các thuộc tính: họ tên, ngày sinh, hệ số lương, số lượng nhân viên quản;
- Nhân viên gồm các thuộc tính: họ tên, ngày sinh, hệ số lương, tên đơn vị (phòng/ban).

