

```

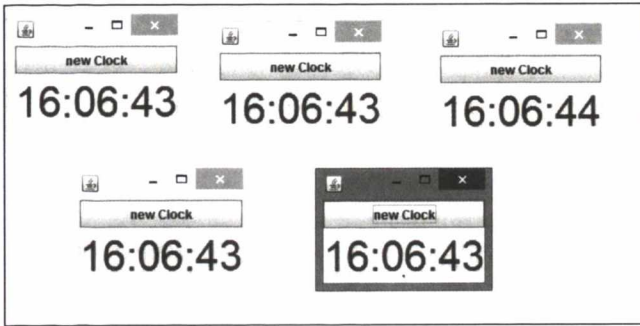
/*Tuyến có tên t*/
Thread t;
public MultiClock()
{
    /*Lấy tầng ContentPane của JFrame để chứa các đối tượng hiển thị*/
    Container cont = this.getContentPane();
    /*Lấy thời gian hiện tại*/
    Calendar cal = Calendar.getInstance();
    /*Thiết lập định dạng hiển thị thời gian giờ:phút:giây*/
    SimpleDateFormat sdf = new SimpleDateFormat("HH:mm:ss");
    /*Hiển thị thời gian nhờ JLabel clock, canh lề giữa*/
    clock = new JLabel(sdf.format(cal.getTime()),JLabel.CENTER);
    /*Thiết lập kích thước và màu chữ*/
    clock.setFont(new Font(clock.getFont().getName(), Font.PLAIN, 40));
    clock.setForeground(Color.RED);
    /*Đặt nút vào phía trên cùng của giao diện*/
    cont.add(createClock,"North");
    /*Đặt JLabel clock ở vùng trung tâm của giao diện*/
    cont.add(clock);
    this.pack();
    this.setVisible(true);
    /*Gắn ống nghe cho nút ấn*/
    createClock.addActionListener(this);
    /*Tạo một tuyến để hiển thị đồng hồ*/
    Thread t = new Thread(this);
    /*Khởi động tuyến và gọi hàm run()*/
    t.start();
}
/*Hàm tick() dùng để cập nhật lại thời gian sau 1 giây*/
public void tick()
{
    try
    {

```

```
    /*Lấy thời gian hiện tại và thiết lập định dạng hiển thị
    thời gian*/
    Calendar cal = Calendar.getInstance();
    SimpleDateFormat sdf = new SimpleDateFormat("HH:mm:ss");
    /*Cập nhật lại giá trị ở JLabel cLock tức là cập nhật
    thời gian*/
    cLock.setText(sdf.format(cal.getTime()));
    /*Dừng 1000 mili giây tức 1 giây trước khi cập nhật thời
    gian mới*/
    Thread.sleep(1000);
}
catch(Exception e)
{
    e.printStackTrace();
}
}
/*Tuyến hoạt động*/
public void run()
{
    /*Vòng lặp vô hạn*/
    while (true)
    {
        /*Cập nhật lại thời gian*/
        tick();
    }
}
/*Tạo một tuyến mới (đồng hồ mới) khi ấn nút New CLock*/
public void actionPerformed(ActionEvent e)
{
    /*Tạo một tuyến biểu diễn đồng hồ*/
    Thread t = new Thread(new MultiClock());
    t.start();
}
public static void main(String[] a)
{
```

```
new MultiClock();  
}  
}
```

Kết quả chương trình:



### 6.3. BÀI TẬP TỰ GIẢI

**Bài 1:** Viết chương trình tạo 3 tuyến: Tuyến 1 cho phép người dùng nhập vào 2 số thực là hai cạnh của hình chữ nhật, tuyến 2 và tuyến 3 sẽ đồng thời tính diện tích và chu vi của hình chữ nhật với 2 cạnh nhập được ở tuyến 1. Tuyến chính chờ các tuyến hoàn thành rồi in kết quả chu vi và diện tích của hình chữ nhật ra màn hình.

**Bài 2:** Viết chương trình Applet hiển thị chuỗi "Multithread programming" chạy từ bên trái cửa sổ sang phải và ngược lại.

**Bài 3:** Viết chương trình tạo 7 tuyến, mỗi tuyến tạo ngẫu nhiên một số từ 1 đến 30. Tuyến chính chờ tất cả các tuyến hoàn thành, tính tổng các số đã được tạo ra từ các tuyến rồi in ra màn hình.

**Bài 4:** Viết chương trình hiển thị 10 hình tròn tô màu khác nhau và xuất phát từ các vị trí ngẫu nhiên, di chuyển và phản xạ theo một quy luật di chuyển và phản xạ của quả bóng bida. Mỗi quả bóng tương ứng có 2 nút nhấn "Stop" và "Start" để điều khiển việc dừng và di chuyển tương ứng.

**Bài 5:** Phát triển bài tập 10 của phần bài tập thực hành mẫu bằng cách thêm một lựa chọn hiển thị thời gian của một thành phố ở các quốc gia khác nhau.

**Bài 6:** Viết chương trình nhân 2 ma trận vuông  $N \times N$ , chương trình có  $N \times N$  tuyến, mỗi tuyến thực hiện việc tính giá trị một phần tử của ma trận kết quả. Tuyến chính sẽ nhận kết quả của các tuyến con để tạo ma trận kết quả và in ra màn hình.

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

---

---

### 7.1. TÓM TẮT LÝ THUYẾT

- Java cung cấp gói `java.io` gồm tập hợp các lớp và interface cho phép thao tác với hệ thống tập tin.
- Lớp `File` và `FileDescriptor` được dùng để lấy thông tin và thao tác với hệ thống tập 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 tập tin.
- `FileReader` và `FileWriter` là hai luồng cho phép đọc và ghi dữ liệu dạng ký tự với tập tin.

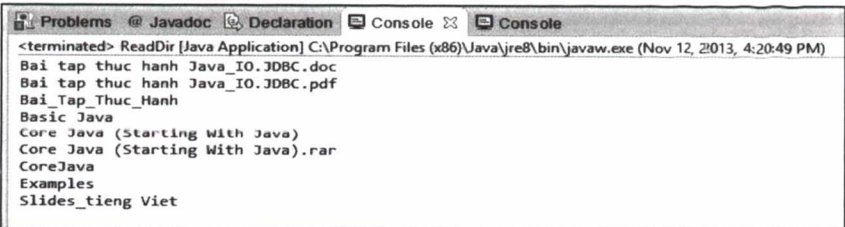
### 7.2. BÀI TẬP THỰC HÀNH MẪU

**Bài 1:** Sử dụng lớp `File`, viết chương trình đọc và in ra màn hình nội dung một thư mục (tên các tập tin hoặc thư mục con)

```
/*Cần import gói java.io để truy xuất hệ thống tập 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 trỏ 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
```

**Bài 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]);
    if (fi.isFile())
        /*Nếu là tệp tin, tăng đếm lên 1. */
        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:*

```

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

```

**Bài 3:** Viết chương trình đếm số lượng ký tự ‘i’ có trong một tệp tin.

```

import java.io.FileInputStream;
import java.io.IOException;

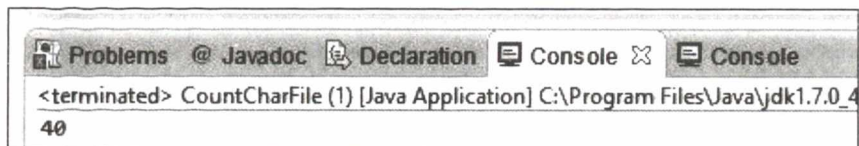
```

```
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:



The screenshot shows an IDE interface with a console window. The console output is: `<terminated> CountCharFile (1) [Java Application] C:\Program Files\Java\jdk1.7.0_40`. The number 40 is printed on the line below.

**Bài 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("Loi IOException");
    }
    finally
    {
        try
        {
            fis.close();
        }
        catch(IOException e)
        {
            System.out.println("Loi 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*/
    }
}
```