

Bài 09:

XỬ LÝ FILE VÀ I/O TRONG

JAVA SWING

Giảng Viên: ThS. Giang Hào Côn

9.0/ FileInputStream và FileOutputStream

a) FileInputStream trong Java là gì?

b) Cách sử dụng FileInputStream:

- **Tạo đối tượng FileInputStream**: Sử dụng `new FileInputStream(String fileName)` hoặc `new FileInputStream(File file)` để tạo một đối tượng FileInputStream. Thay thế fileName bằng tên file hoặc file bằng đối tượng File đại diện cho file bạn muốn đọc.
- **Đọc dữ liệu**: Sử dụng các phương thức như `read()` hoặc `read(byte[] b)` để đọc dữ liệu byte từ file.

9.0/ FileInputStream và FileOutputStream

b) Cách sử dụng FileInputStream:

- Đóng file: Sử dụng phương thức `close()` để đóng file và giải phóng tài nguyên.

```
try (FileInputStream fis = new FileInputStream("myfile.txt")) {  
    int data;  
    while ((data = fis.read()) != -1) {  
        System.out.print((char) data);  
    }  
} catch (IOException e) {  
    e.printStackTrace();  
}
```

9.0/ FileInputStream và FileOutputStream

a) FileOutputStream trong Java là gì?

b) Cách sử dụng FileOutputStream:

- **Tạo đối tượng FileOutputStream:** Sử dụng `new FileOutputStream(String fileName)` hoặc `new FileOutputStream(File file)` để tạo một đối tượng FileOutputStream. Thay thế fileName bằng tên file hoặc file bằng đối tượng File đại diện cho file bạn muốn ghi..
- Ghi dữ liệu: Sử dụng các phương thức như `write(byte b)` hoặc `write(byte[] b)` để ghi dữ liệu byte vào file..

9.0/ FileInputStream và FileOutputStream

b) Cách sử dụng FileOutputStream:

- Đóng file: Sử dụng phương thức `close()` để đóng file và giải phóng tài nguyên.

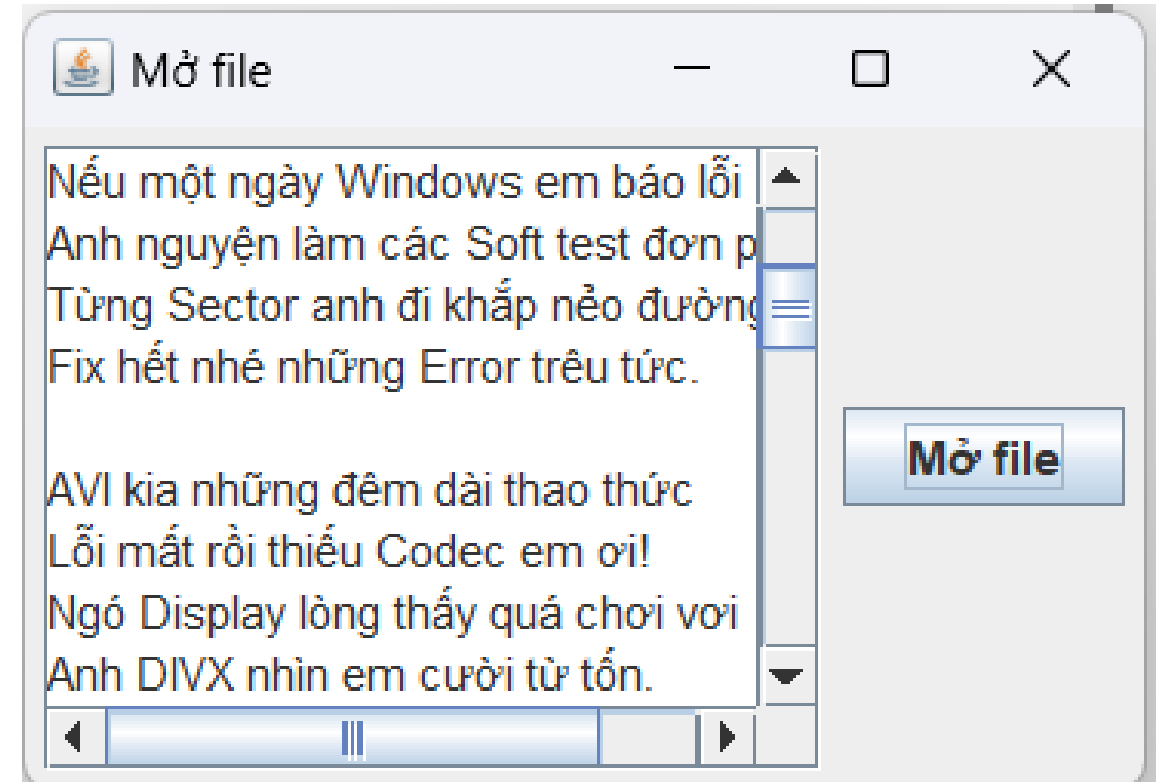
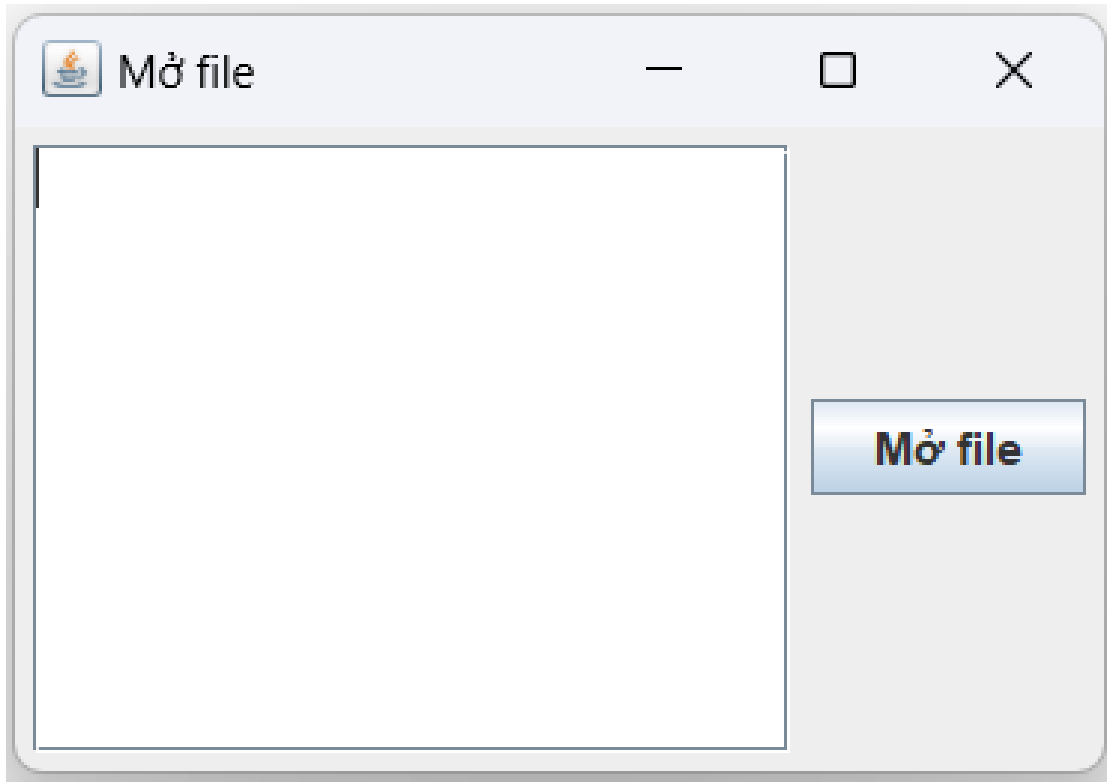
```
try (FileOutputStream fos = new FileOutputStream("myfile.txt")) {  
    String message = "Hello, world!";  
    byte[] bytes = message.getBytes();  
    fos.write(bytes);  
} catch (IOException e) {  
    e.printStackTrace();  
}
```

9.1/ Byte stream trong Java

- a) Byte stream trong Java là gì?
 - b) Ưu điểm và nhược điểm của byte stream
 - c) Các phương thức sử dụng Byte Stream
 - Để đọc byte stream, sử dụng phương thức `read()` hoặc `read(byte[])`.
 - Để ghi byte stream, sử dụng phương thức `write(byte)` hoặc `write(byte[])`
- Lưu ý

9.1/ Byte stream trong Java

Ví dụ: Đọc file văn bản hiện ra màn hình



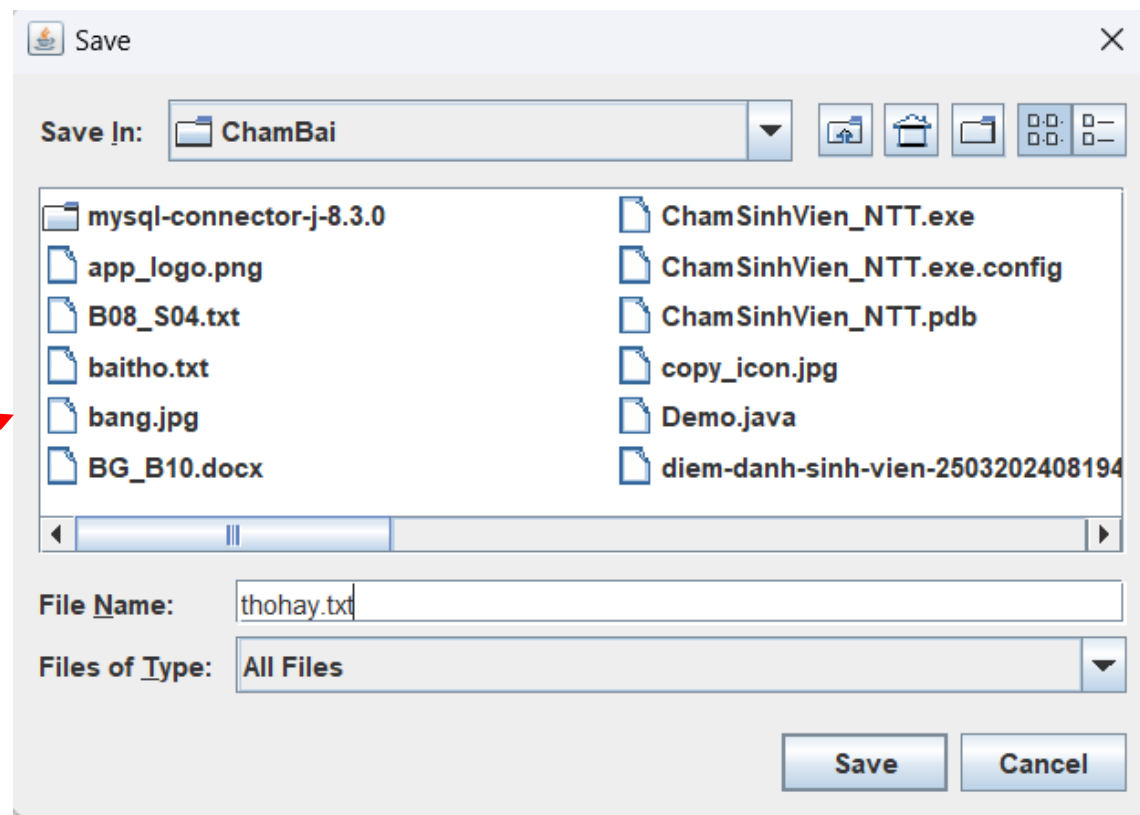
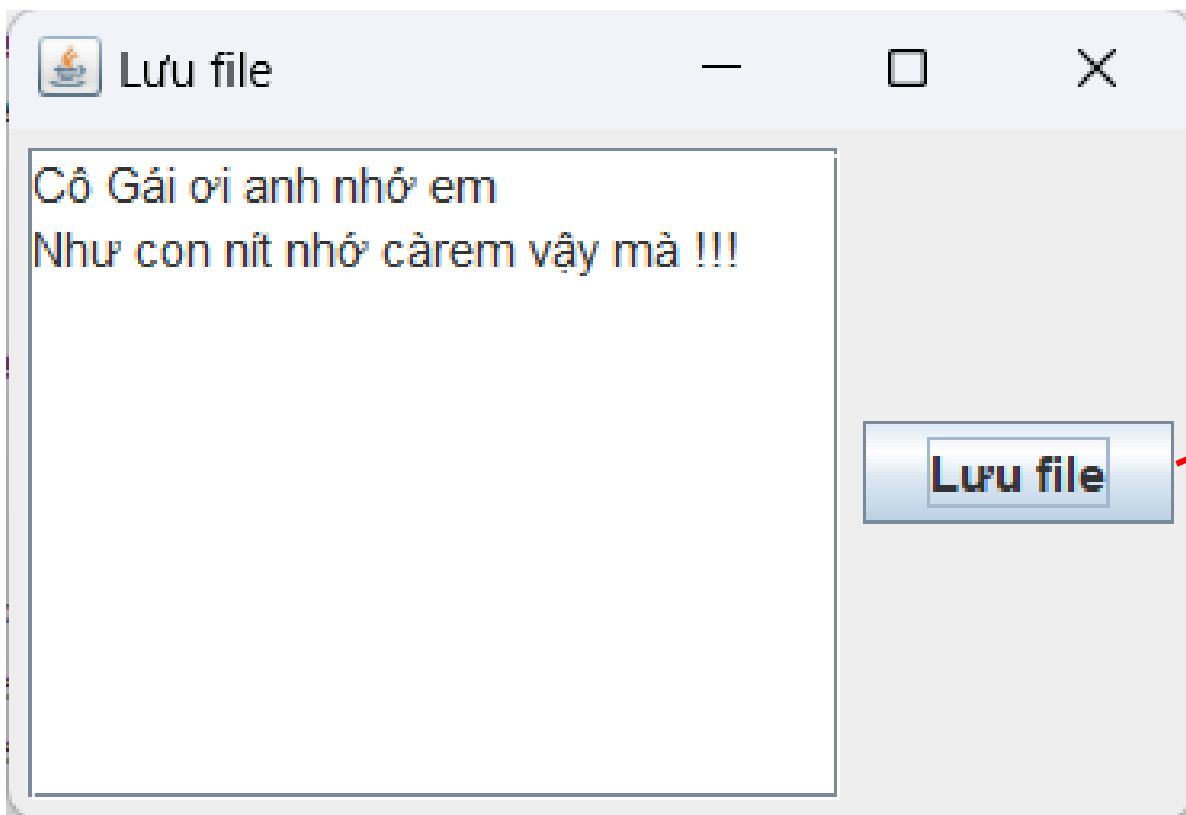
9.1/ Byte stream trong Java

Ví dụ: Đọc file văn bản hiện ra màn hình

```
try (FileInputStream fis = new FileInputStream("myfile.txt");  
    BufferedReader br = new BufferedReader(new InputStreamReader(fis)))  
String line;  
while ((line = br.readLine()) != null) {  
    textArea.append(line + "\n");  
}  
} catch (IOException e) {  
    e.printStackTrace();  
}
```


9.1/ Byte stream trong Java

Ví dụ: Đọc file văn bản hiện ra màn hình



9.1/ Byte stream trong Java

Ví dụ: Đọc file văn bản hiện ra màn hình

```
public void actionPerformed(ActionEvent e) {  
    if (e.getSource() == saveButton) {  
        JFileChooser fileChooser = new JFileChooser();  
        int result = fileChooser.showSaveDialog(this);  
  
        if (result == JFileChooser.APPROVE_OPTION) {  
            File selectedFile = fileChooser.getSelectedFile();  
            try (FileOutputStream fos = new FileOutputStream(selectedFile);  
                BufferedWriter bw = new BufferedWriter(new OutputStreamWriter(fos))) {  
                bw.write(textArea.getText());  
            } catch (IOException ex) {  
                ex.printStackTrace();  
            }  
        }  
    }  
}
```

9.2/ Character Stream trong Java

- a) Character Stream trong Java là gì?
- b) Ưu điểm và nhược điểm của character stream
- c) Các phương thức sử dụng character Stream
 - Để đọc character stream, sử dụng phương thức `read()` hoặc `read(char[])`.
 - Để ghi character stream, sử dụng phương thức `write(char)` hoặc `write(char[])`.

■ Lưu ý

9.2/ Character Stream trong Java

- Đọc file văn bản in ra màn hình

```
try (FileReader fr = new FileReader("myfile.txt");  
    BufferedReader br = new BufferedReader(fr)) {  
    String line;  
    while ((line = br.readLine()) != null) {  
        System.out.println(line);  
    }  
} catch (IOException e) {  
    e.printStackTrace();  
}
```

9.2/ Character Stream trong Java

- Ghi một chuỗi vào tập tin text

```
try (FileWriter fw = new FileWriter("myfile.txt");  
    BufferedWriter bw = new BufferedWriter(fw)) {  
    String message = "Hello, world!";  
    bw.write(message);  
} catch (IOException e) {  
    e.printStackTrace();  
}
```

9.3/ Standard Stream trong Java Swing

- a) Standard Stream trong Java Swing là gì?
- b) Cách sử dụng Standard Stream:
 - Để đọc dữ liệu từ `System.in`, sử dụng các lớp `Scanner` hoặc `BufferedReader`.
 - Để ghi dữ liệu ra `System.out`, sử dụng các lớp `System.out.print()` hoặc `System.out.println()`.
 - Để ghi thông báo lỗi ra `System.err`, sử dụng các lớp `System.err.print()` hoặc `System.err.println()`

9.3/ Standard Stream trong Java Swing

Ví dụ

```
import java.util.Scanner;

public class StandardStreamExample {

    public static void main(String[] args) {
        // Đọc dữ liệu từ bàn phím
        Scanner scanner = new Scanner(System.in);
        System.out.print("Nhập tên của bạn: ");
        String name = scanner.nextLine();

        // Ghi dữ liệu ra màn hình
        System.out.println("Xin chào, " + name + "!");

        // Ghi thông báo lỗi ra màn hình
        System.err.println("Có lỗi xảy ra!");
    }
}
```

Ưu điểm và
nhược điểm ?

Bài Tập

- 1) Stream (luồng) là gì ? Trình bày các loại Stream trong Java.
- 2) Byte Stream trong Java là gì ? Trình ưu điểm, nhược điểm và cách sử dụng Byte Stream trong Java.
- 3) Standard Stream trong Java là gì ? Trình bày ưu điểm, nhược điểm và cách sử dụng Standard Stream trong Java.