

File và xử lý File

Nội dung

1. Lớp File
2. FileInputStream và FileOutputStream
3. Reader và Writer
4. Serialization

Lớp File

Java cung cấp cho ta **Lớp File** có chức năng:

- Xử lý các thao tác cơ bản trên file, ví dụ như:
 - + tạo file
 - + xoá file
 - + chèn file
 - + tìm kiếm
 - + kiểm tra tồn tại
 - +

Lớp File

Contrucstor

- `File(String dirpath)`
- `File(String dirpath, String name)`
- `File(File obj, String name)`
- `File(URL urlobj)`



Lớp File

Ví dụ

```
File f1 = new File("C:\\JavaLesson\\data.txt");  
File f2 = new File("C:\\JavaLesson", "data.txt");  
...
```

Lớp File

Phương thức quan trọng

- `renameTo(File newFile)`
- `delete()`
- `exists()`
- `isFile()`
- `mkdir()`
- `createNewFile()`
- `getPath()`

Lớp File

Ví dụ

```
File f = new File("C:\\JavaLesson\\data.txt");  
if (f.exists()) {  
    System.out.print(f.getPath());  
} else {  
    f.createNewFile();  
}
```

FileInputStream và FileOutputStream

Đây là 2 Lớp giúp đọc và ghi file sử dụng Stream

- FileInputStream
- FileOutputStream

FileInputStream

Công dụng

- Sử dụng cho việc đọc file

Contrucstor

- `FileInputStream(String filePath)`
- `FileInputStream(File objFile)`

FileInputStream

Ví dụ

```
FileInputStream fis = new  
    FileInputStream("C:\\JavaLesson\\data.txt");  
  
FileInputStream fis = new FileInputStream(  
    new File ("C:\\JavaLesson\\data.txt"));
```

FileInputStream

Phương thức quan trọng

- `read(File newFile)`
- `close()`
- `skip(long n)`
- `mark(int n)`
- `available()`
- `reset()`

FileInputStream

Ví dụ

```
try {
    FileInputStream fis = new FileInputStream("data.txt");
    int size = fis.available();
    for (int i = 0; i < size; i++) {
        System.out.print((char) fis.read());
    }
    fis.close();
} catch (FileNotFoundException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}
```

FileOutputStream

Công dụng

- Sử dụng cho việc **ghi** file

Contrucstor

- `FileOutputStream(String filePath)`
- `FileOutputStream(File objFile)`

FileOutputStream

Ví dụ

```
FileOutputStream fos = new  
    FileOutputStream("C:\\JavaLesson\\data.txt");  
  
FileOutputStream fos = new FileOutputStream(  
    new File ("C:\\JavaLesson\\data.txt"));
```

FileOutputStream

Phương thức quan trọng

- `write(int i)`
- `write(byte[] b)`
- `flush()`
- `close()`

FileOutputStream

Ví dụ

```
byte bWrite [] = {11,21,3,40,5};
try {
    FileOutputStream fos = new FileOutputStream("test.txt");
    for(int x = 0; x < bWrite.length ; x++) {
        fos.write( bWrite[x] );    // writes the bytes
    }
    fos.close();
} catch (FileNotFoundException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}
```


BufferedReader

Công dụng

- Sử dụng cho việc **đọc** file, cho phép đọc vào nhiều kiểu dữ liệu
- Sử dụng buffer để tăng tốc độ truy xuất

Contrucstor

- `BufferedReader(Reader reader)`

BufferedReader

Ví dụ

```
BufferedReader reader = new BufferedReader(  
    new InputStreamReader(  
        new FileInputStream("data.txt")) );
```

```
BufferedReader reader = new  
    BufferedReader(new InputStreamReader(System.in))
```

BufferedReader

Phương thức quan trọng

- `read()`
- `readLine(byte[] b)`
- `skip(int n)`
- `reset()`
- `close()`

BufferedReader

Ví dụ

```
BufferedReader reader = new  
    BufferedReader(new InputStreamReader(System.in));  
String line = reader.readLine();  
String age = Integer.parseInt(reader.readLine())
```

PrintWriter

Công dụng

- Sử dụng cho việc **ghi** file, cung cấp các phương thức tương tự **System.out**

PrintWriter

Contrucstor

- PrintWriter(String filePath)
- PrintWriter(File file)
- PrintWriter(OutputStream out)
- PrintWriter(OutputStream out, boolean autoFlush)
- PrintWriter(Writer out)
- PrintWriter(Writer out, boolean autoFlush)

PrintWriter

Ví dụ

```
PrintWriter writer = new PrintWriter(  
    new FileOutputStream("data.txt"));
```

PrintWriter

Phương thức quan trọng

- `print()`
- `println()`
- `printf()`
- `close()`
- `flush()`

PrintWriter

Ví dụ

```
try {
    PrintWriter pw = new PrintWriter(
        new FileOutputStream("data.dat"));
    pw.println("PrintWriter is easy to use.");
    pw.println(1234);
    pw.close();
} catch (FileNotFoundException e) {
    e.printStackTrace();
}
```

Serialization



Serialization

Công dụng

- Đọc và ghi file dưới dạng byte stream
- Hỗ trợ đọc/ghi một object
- Lưu giữ trạng thái của Object khi được lưu và có thể khôi phục lại được

ObjectInputStream

Công dụng

- Sử dụng cho việc đọc file Object

Contrucstor

-ObjectInputStream()

-ObjectInputStream(InputStream input)

ObjectInputStream

Ví dụ

```
FileInputStream fis = new  
    FileInputStream("C:\\JavaLesson\\data.txt");  
ObjectInputStream ois = new  
    ObjectInputStream(fis);
```

ObjectInputStream

Phương thức quan trọng

- `readFloat()`
- `readBoolean()`
- `readDouble()`
- `readObject()`

ObjectInputStream

Ví dụ

```
try {  
    FileInputStream fi = new FileInputStream ("message.obj");  
    ObjectInputStream oi = new ObjectInputStream (fi);  
    Message mess = (Message) oi.readObject ();  
    System.out.println ("Message:\n");  
    System.out.println ("From: " + mess.from);  
    System.out.println ("To: " + mess.to);  
    System.out.println ("Date: " + mess.when + "\n");  
    oi.close ();  
} catch (Exception e) {  
}
```

ObjectOutputStream

Công dụng

- Sử dụng cho việc ghi file Object

Contrucstor

-ObjectOutputStream()

-ObjectOutputStream(OutputStream input)

ObjectOutputStream

Ví dụ

```
FileOutputStream fos = new  
    FileOutputStream("C:\\JavaLesson\\data.txt");  
ObjectOutputStream oos = new  
    ObjectOutputStream(fos);
```

ObjectOutputStream

Phương thức quan trọng

- `writeFloat()`
- `writeObject()`

ObjectOutputStream

Ví dụ

```
try {  
    FileOutputStream fo = new FileOutputStream("Message.obj");  
    ObjectOutputStream oo = new ObjectOutputStream(fo);  
    oo.writeObject(student); // student là đối tượng cần ghi  
    oo.close();  
} catch (IOException e) {  
    e.printStackTrace();  
}
```

ObjectOutputStream

Lưu ý

- Để ghi được đối tượng ra file sử dụng ObjectOutputStream thì đối tượng đó phải được Serialize

```
class Student implements Serializable
```

File và xử lý File