

Bài 37: Ghi file với PrintWriter

- ✓ Thao tác với thư mục, file
- ✓ Tìm hiểu lớp PrintWriter
- ✓ Tìm hiểu lớp FileWriter
- ✓ Ghi dữ liệu ra màn hình
- ✓ Ghi dữ liệu ra file
- ✓ Ví dụ minh họa và bài tập

Thao tác với thư mục, file

➤ Một số phương thức của lớp File dùng cho mục đích này:

Phương thức	Mô tả
renameTo(newFile)	Đổi tên file được trỏ tới bởi đối tượng file hiện thời thành file mới với tên được cung cấp bởi đối tượng newFile trong tham số. Nếu việc đổi tên thực hiện thành công thì trả về true và ngược lại trả về false.
createNewFile()	Tạo file mới với tên được cung cấp trong tham số của constructor nếu như file được chỉ ra đó chưa tồn tại trong đường dẫn. Trả về true nếu quá trình tạo file mới thành công và false trong trường hợp ngược lại.
mkdir()	Tạo thư mục đích với tên được trỏ đến bởi đối tượng file này. Trả về true nếu tạo file thành công và false nếu ngược lại.
mkdirs()	Tạo thư mục đích và tất cả các thư mục cha trong đường dẫn được trỏ tới bởi đối tượng file này. Trả về true nếu tạo thành công và false nếu ngược lại.
delete()	Xóa tệp hoặc thư mục chỉ ra trong đường dẫn của đối tượng file hiện thời. Nếu là thư mục thì muốn xóa được nó phải rỗng. Trả về true nếu xóa thành công và false nếu ngược lại.
deleteOnExist()	Xóa tệp hay thư mục nếu nó tồn tại.

Ví dụ

➤ Ví dụ tạo file:

```
var filePath = "output.txt";  
File file = new File(filePath);  
var isCreateNewFileSuccess = file.createNewFile();  
System.out.println("Tạo mới file thành công? "  
    + isCreateNewFileSuccess);
```

➤ Ví dụ đổi tên file:

```
var filePath = "output.txt";  
File file = new File(filePath);  
File newFile = new File("OUTPUT.OUT");  
file.renameTo(newFile);  
System.out.println("File cũ: " + file.getName());  
System.out.println("File mới: " + newFile.getName());
```

Lớp PrintWriter

- Mục đích sử dụng của file: lưu trữ lâu dài dữ liệu. Thuận tiện cho việc vận chuyển, chia sẻ...
- Lớp PrintWriter sử dụng để ghi dữ liệu dạng text ra file.
- Dữ liệu dạng text là dữ liệu dạng tường minh con người có thể đọc và hiểu được
- Sử dụng các phương thức như print, println, printf để ghi file

Một số phương thức

Phương thức	Mô tả
<code>PrintWriter(String fileName)</code>	Tạo một đối tượng <code>PrintWriter</code> để dữ liệu vào tên file đã chỉ ra trong tham số. Không tự động xóa bộ đệm.
<code>PrintWriter(String fileName, String csn)</code>	Tạo ra một đối tượng <code>PrintWriter</code> để dữ liệu vào file có tên cho trong tham số thứ nhất và hỗ trợ tập csn trong tham số thứ hai. Không tự động xóa bộ đệm. (csn = <i>character-encoding scheme: lược đồ mã hóa kí tự</i>)
<code>PrintWriter(File file)</code>	Tạo một đối tượng <code>PrintWriter</code> trở đến đối tượng <code>File</code> đã được chỉ ra trong tham số. Không tự động xóa bộ đệm.
<code>PrintWriter(File file, String csn)</code>	Tạo một đối tượng <code>PrintWriter</code> trở đến đối tượng <code>File</code> đã được chỉ ra trong tham số thứ nhất và hỗ trợ tập csn cho trong tham số thứ hai. Không tự động xóa bộ đệm.
<code>PrintWriter(OutputStream out)</code>	Tạo một đối tượng <code>PrintWriter</code> mới từ đối tượng <code>OutputStream</code> cho trước. Không tự động xóa bộ đệm.
<code>PrintWriter(OutputStream out, boolean autoFlush)</code>	Tạo một đối tượng <code>PrintWriter</code> mới từ đối tượng <code>OutputStream</code> cho trước và thiết lập kích hoạt hoặc không tự động xóa bộ đệm.
<code>PrintWriter(Writer out)</code>	Tạo một đối tượng <code>PrintWriter</code> từ một đối tượng <code>Writer</code> cho trước.
<code>PrintWriter(Writer out, boolean autoFlush)</code>	Tạo một đối tượng <code>PrintWriter</code> từ một đối tượng <code>Writer</code> cho trước và thiết lập auto flush.

Ví dụ về PrintWriter

➤ Sau đây là ví dụ về sử dụng PrintWriter:

```
PrintWriter printWriter = new PrintWriter("output.txt", "UTF_8" );  
// tạo đối tượng PrintWriter để in ra màn hình  
PrintWriter printWriter = new PrintWriter(System.out);  
// tạo đối tượng PrintWriter sử dụng đối tượng File  
var filePath = "output.txt";  
File file = new File(filePath);  
PrintWriter printWriter = new PrintWriter(file, "UTF_8" );  
// tạo đối tượng PrintWriter sử dụng đối tượng của FileWriter  
var filePath = "output.txt";  
File file = new File(filePath);  
// sử dụng FileWriter nếu muốn thêm dữ liệu vào file  
// ở chế độ append  
FileWriter fileWriter = new FileWriter(file, true);  
PrintWriter printWriter = new PrintWriter(fileWriter, true);
```

Lớp FileWriter

- Là một lớp tiện ích dùng để ghi dữ liệu dạng text ra file. Giả định rằng các mặc định mã hóa kí tự và mặc định kích thước bộ đệm đã được chấp thuận
- Nếu file cần thao tác với mà chưa tồn tại sẽ được tạo mới
- Một file tại 1 thời điểm chỉ được sử dụng bởi 1 đối tượng FileWriter duy nhất
- Do đó việc tạo đối tượng mới trở đến một file đang được sử dụng bởi một object FileWriter khác sẽ thất bại

Một số constructor

➤ Các constructor và mô tả:

Phương thức	Mô tả
<code>FileWriter(String fileName)</code>	Tạo đối tượng <code>FileWriter</code> từ một tên file cho trước trong tham số.
<code>FileWriter(String fileName, boolean append)</code>	Tạo đối tượng <code>FileWriter</code> từ một tên file nào đó cho trước trong tham số. Đồng thời xác định xem có kích hoạt chế độ ghi thêm vào cuối file hay không.
<code>FileWriter(File file)</code>	Tạo đối tượng <code>FileWriter</code> từ một đối tượng file cho trước trong tham số.
<code>FileWriter(File file, boolean append)</code>	Tạo đối tượng <code>FileWriter</code> từ một đối tượng file cho trước trong tham số. Đồng thời xác định xem có kích hoạt chế độ ghi thêm vào cuối file hay không.

Ghi dữ liệu ra màn hình

- Trước hết ta tạo một đối tượng của `PrintWriter` nhận tham số là `System.out`
- Sau đó sử dụng các phương thức `print`, `printf`, `println` như bình thường
- Ví dụ:

```
// tạo đối tượng PrintWriter để đẩy dữ liệu ra màn hình console  
PrintWriter printWriter = new PrintWriter(System.out, true);  
// sử dụng như ghi dữ liệu ra màn hình trước nay  
printWriter.println("Hello World!");  
printWriter.println("Hey, who you are?");  
printWriter.printf("My average grade is %5.2f", 3.26f);
```

Lưu ý

- Nếu không để auto flush thì dữ liệu sẽ không tự động được đẩy ra và bộ đệm sẽ không bị xóa đi
- Ta phải thực hiện điều này bằng cách gọi phương thức flush()
- Ví dụ:

```
// tạo đối tượng PrintWriter để đẩy dữ liệu ra màn hình console
PrintWriter printWriter = new PrintWriter(System.out);
// sử dụng như ghi dữ liệu ra màn hình trước nay
printWriter.println("Hello World!");
printWriter.print("Hey, who you are?");
printWriter.flush(); // đẩy dữ liệu ra và xóa bộ đệm
printWriter.printf("\nMy average grade is %5.2f", 3.26f);
```

Ghi dữ liệu ra file

- Các bước để ghi dữ liệu ra file:
 - Tạo file
 - Mở file
 - Ghi dữ liệu
 - Đóng file
- Khi đọc dữ liệu ra ta cần đọc đúng theo thứ tự và kiểu khi đã ghi vào file nếu không sẽ bị lỗi
- Sử dụng các phương thức `parseX(String str)` của các kiểu lớp bao để thực hiện việc chuyển đổi dữ liệu dạng string sang kiểu mong muốn.
- Ví dụ: `int number = Integer.parseInt(str);`

Ví dụ về ghi file

➤ Ví dụ sau ghi một số thông tin đơn giản vào file:

```
var filePath = "output.txt";
File file = new File(filePath);
FileWriter fileWriter = new FileWriter(file, true);

PrintWriter printWriter = new PrintWriter(fileWriter, true);
// sử dụng như ghi dữ liệu ra màn hình trước nay
printWriter.println("Hello World!");
printWriter.print("Hey, who you are?");
printWriter.flush(); // đẩy dữ liệu ra và xóa bộ đệm
printWriter.printf("\nMy average grade is %5.2f", 3.26f);

printWriter.close();
fileWriter.close();
```



Ví dụ minh họa

- Tạo thư mục, tạo file, đổi tên file
- Ghi thông tin ra màn hình
- Ghi thông tin ra file
- Đọc file và chuyển String thành số

Nội dung tiếp theo

Lớp Date