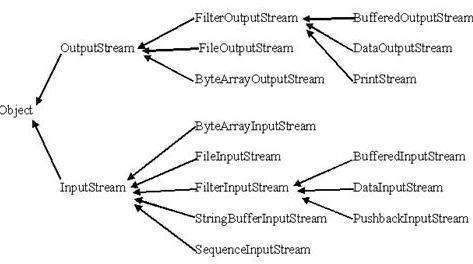
**Java.io và đọc ghi file trong java**

Java.io chứa tất cả những lớp cần thiết để thực hiện input và output trong java. Trogn những Stream này biểu diễn một nguồn Input đến đích đến Output. Một stream có thể được định nghĩa như một dãy liên tục. InputStream được dùng để đọc dữ liệu từ một nguồn và OutputStream được dùng để ghi dữ liệu tới 1 đích.



Tùy vào loại dữ liệu muốn đọc mà ta chọn những loại đọc và ghi khác nhau:

+ Byte Stream: dùng để đọc và ghi các bytes 8bit, 2 lớp thường sử dụng nhất trong byte Stream là : FileInputStream và FileOutputStream.

+ Character Stream: dùng để input và output các bytes Unicode 16bit, 2 lớp thường được sử dụng nhất trong Character Stream là FileReader và FileWriter. Mặc dù về bản chất FileReader và FileWriter sử dụng FileInputStream và FileOutputStream, nhưng sự khác biệt là FileReader và FileWriter đọc ghi 1 lần 2 bytes.

+ Standard Stream: cũng giống như bao ngôn ngữ khác java cũng cung cấp bộ nhập xuất chuẩn từ bàn phim:

* Chuẩn đầu vào : System.in
* Chuẩn đầu ra : System.out
* Lỗi chuẩn : System.err

1. Đọc ghi file với FileInputStream và FileOutputStream

1.1 FileInputStream

FileInputStream đọc dữ liệu trực tiếp từ file nên ta phải có đường dẫn file khi khai báo.

*\* ta có thể khai báo đường dẫn file vào 1 biến và sau đó truyền biến đó vào thì sẽ thuận tiện hơn cho việc thêm sửa xóa sau này.*



Những phương thức :

|  |  |
| --- | --- |
| **STT** | **Phương thức và Miêu tả** |
| 1 | **public void close() throws IOException{}**  Phương thức này đóng output stream. Giải phóng bất kỳ nguồn hệ thống nào liên kết với file. Ném một IOException |
| 2 | **protected void finalize()throws IOException {}**  Phương thức này xóa sự kết nối tới File đó. Bảo đảm rằng phương thức close của output stream này được gọi khi không có tham chiếu nào nữa tới stream này. Ném một IOException |
| 3 | **public int read(int r)throws IOException{}**  Phương thức này đọc byte dữ liệu đã xác định từ InputStream. Trả về một int. Trả về byte dữ liệu tiếp theo và -1 sẽ được trả về nếu kết thúc file. |
| 4 | **public int read(byte[] r) throws IOException{}**  Phương thức này đọc r byte từ input stream vào trong một mảng. Trả về tổng số byte đã đọc. Nếu kết thúc file, -1 được trả về. |
| 5 | **public int available() throws IOException{}**  Cung cấp số byte mà được đọc từ input stream này. Trả về một int |

2.2 FileOutputStream

FileOutputStream sử dụng file để ghi dữ liệu nếu file chưa tồn tại thì nó sẽ tự động tạo file đó và ghi dữ liệu vào.

*\* ta cũng nên khai báo đường dẫn file vào 1 biến và sau đó truyền biến đó vào thì sẽ thuận tiện hơn cho việc thêm sửa xóa sau này.*



Phương thức để thao tác với việc ghi file:

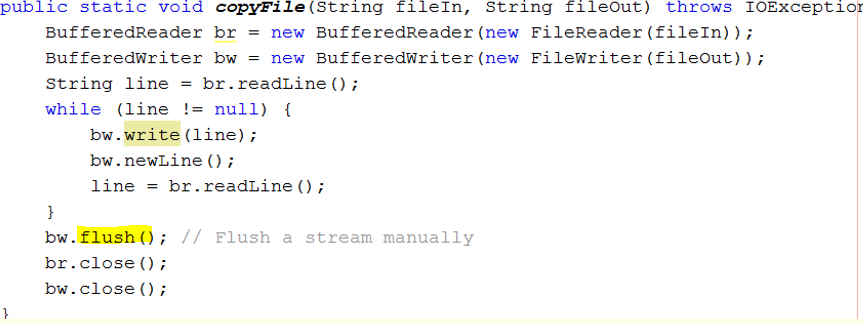
|  |  |
| --- | --- |
| **STT** | **Phương thức và Miêu tả** |
| 1 | **public void close() throws IOException{}**  Phương thức này đóng output stream. Giải phóng bất kỳ nguồn hệ thống nào liên kết với file. Ném một IOException |
| 2 | **protected void finalize()throws IOException {}**  Phương thức này xóa sự kết nối tới File đó. Bảo đảm rằng phương thức close của output stream này được gọi khi không có tham chiếu nào nữa tới stream này. Ném một IOException |
| 3 | **public void write(int w)throws IOException{}**  Phương thức này ghi byte đã xác định tới output stream |
| 4 | **public void write(byte[] w)**  Ghi w byte từ mảng byte đã đề cập tới OutputStream. |

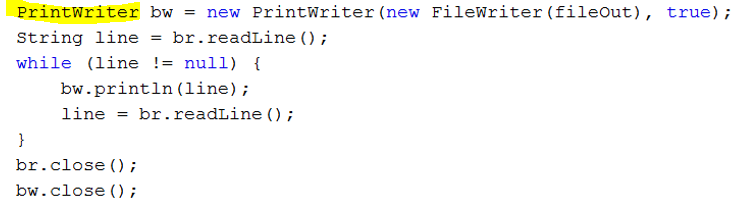
2. Sử dụng luồng đệm (buffered Stream) để đọc ghi file

- I/O của luồng không đệm kiểm soát trực tiếp bởi HDH, ràng buộc với thao tác vật lý gây ra việc tốn chi phí và kém hiệu quả.

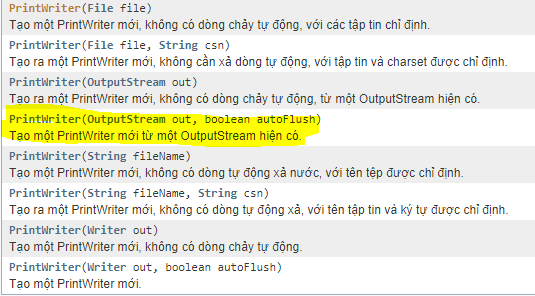
- I/O sử dụng luồng đệm sẽ dùng bộ nhớ (Ram) như vung đệm để I/O data. Native API (lời gọi hệ thống) được gọi khi buffer trống khi đọc và đầy khi ghi.

\* lưu ý khi sử dụng luồng đệm để đọc ghi dữ liệu ta phải dùng phương thức flush() để đẩy dữ liệu vào vùng đệm khi ghi file. Những nếu ta sử dụng PrintWriter thì ta không cần dùng phương thức này.





3. PrintWriter (ghi file với luồng đệm auto flush)



Khi ta thêm biến true vào sau thì PrintWriter sẽ auto flush cho ta.

- 2 phương thức sử dụng chủ yếu của PrintWriter là print và write.

4. Các thao tác với File

Lớp File cung cấp cho ta 3 constructor:

+ File (“đường dẫn tuyệt đối tới file”);

+ File (“tên đường dẫn”, “tên file”): đường dẫn tới file và tên file tách biệt

+ File (File, “tên file con”): tạo 1 file trong File.

Một số phương thức thường dùng với File:

