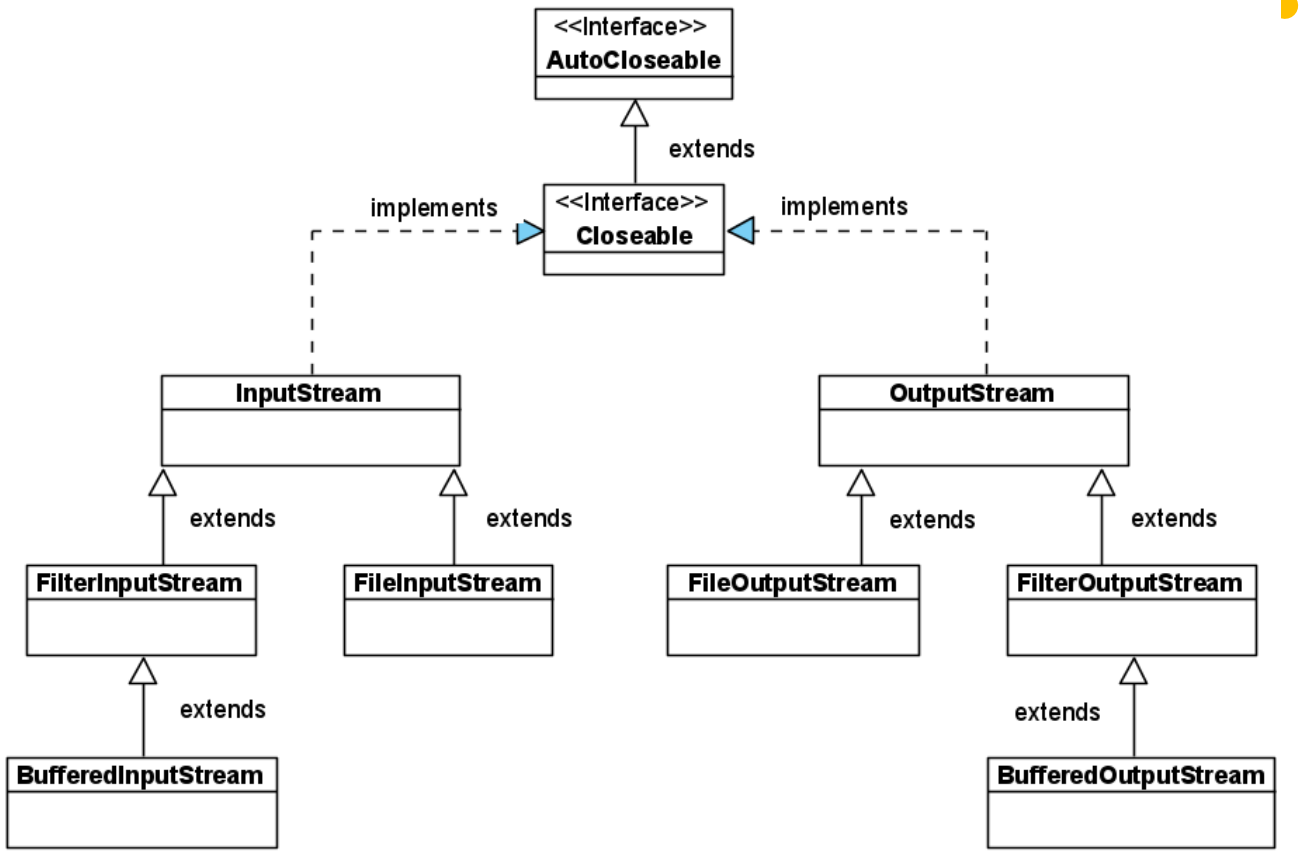
Bài 72: Đọc ghi file nhị phân

- ✓ Giới thiệu
- ✓ FileInputStream vs FileOutputStream
- ✓ BufferedInputStream vs BufferedOutputStream
- ✓ Minh họa

Giới thiệu

- Ta đã biết cách đọc ghi file text sử dụng Scanner và PrintWriter
- Nội dung này nói về việc đọc ghi file nhị phân
- File nhị phân là file mà dữ liệu của nó chỉ chứa các bit nhị phân. Chỉ máy tính mới hiểu nội dung file
- Các file ảnh, âm thanh, video là một ví dụ về file nhị phân
- Khi ta thao tác với file nhị phân, ta sử dụng luồng các byte để đọc ghi dữ liệu dạng nhị phân

Sơ đồ phân cấp



Lớp InputStream

Là lớp cha trừu tượng của tất cả các lớp đại diện cho một luồng byte đầu vào. Một số phương thức:

Phương thức	Mô tả
int available()	Trả về một ước lượng của số byte có thể đọc từ luồng input này mà không bị chặn bởi lệnh kế tiếp trên luồng input này.
void close()	Đóng luồng đầu vào này và giải phóng bất kì tài nguyên hệ thống liên kết với luồng hiện tại.
void mark(int readlimit)	Đánh dấu vị trí hiện tại trong luồng input.
abstract int read()	Đọc byte kế tiếp của luồng đầu vào. Trả về byte dữ liệu kế tiếp hoặc -1 nếu đọc đến cuối file.
int read(byte[] b)	Đọc một lượng các byte từ luồng input và lưu kết quả vào mảng bộ đệm b. Trả về tổng số byte dữ liệu đọc được hoặc -1 nếu đọc đến cuối file.
int read(byte[] b, int off, int len)	Đọc tối đa len byte dữ liệu từ luồng input vào trong mảng bộ đệm b. Trả về tổng số byte dữ liệu đọc được hoặc -1 nếu đọc đến cuối file.
void reset()	Reset lại vị trí trong luồng hiện tại về vị trí lần cuối gọi phương thức mark().
long skip(long n)	Bỏ qua và loại bỏ n byte dữ liệu từ luồng input này.

Lớp FileInputStream

Nhận dữ liệu đầu vào từ một file trong hệ thống file. Chủ yếu để đọc các dữ liệu thô như âm thanh, hình ảnh. Một số phương thức của lớp và mô tả:

Phương thức	Mô tả
FileInputStream(File file)	Tạo đối tượng FileInputStream bằng cách mở một kết nối đến đối tượng file cho trong tham số.
FileInputStream(String name)	Tạo một đối tượng FileInputStream bằng cách mở một kết nối đến file có đường dẫn được cho trong tham số.
void finalize()	Để chắc chắn rằng phương thức close() của đối tượng hiện thời được gọi khi không có bất kì tham chiếu nào đến chính nó.
FileChannel getChanel()	Trả về một đối tượng duy nhất liên kết với đối tượng hiện thời.
int read()	Đọc 1 byte của luồng input này. Trả về byte dữ liệu kế tiếp hoặc -1 nếu đọc đến cuối file.
int read(byte[] b)	Đọc tối đa b.length byte dữ liệu từ luồng input này và lưu vào mảng b. Trả về tổng số byte dữ liệu đọc được hoặc -1 nếu đọc đến cuối file.
int read(byte[] b, int off, int len)	Đọc tối đa len byte dữ liệu từ luồng input vào mảng b. Trả về tổng số byte dữ liệu đọc được hoặc -1 nếu đọc đến cuối file

Lớp OutputStream

Là lớp cha trừu tượng đại diện cho luồng file đầu ra. Một luồng đầu ra nhận dữ liệu từ một nguồn nào đó rồi gửi đến nơi lưu trữ dữ liệu. Một số phương thức của lớp và mô tả:

Phương thức	Mô tả
void close()	Đóng luồng đầu ra và giải phóng bất cứ tài nguyên hệ thống nào đang liên kết với luồng này.
void flush()	Làm rỗng luồng byte đầu ra và đẩy các byte trong bộ đệm của luồng output này ra nơi chứa.
void write(byte[] b)	Ghi b.length byte từ tham số vào luồng output.
void write(byte[] b, int off, int len)	Ghi len byte từ tham số thứ nhất, bắt đầu từ vị trí off vào luồng output.
abstract void write(int b)	Ghi byte được chỉ định vào luồng output này.

Lớp FileOutputStream

- Dùng để ghi dữ liệu ra file hoặc FileDescriptor.
- File đầu ra có thể có sẵn hoặc được tạo mới tùy thuộc hệ điều hành
- Ở một số nền tảng mỗi file chỉ cho phép 1 đối tượng FileOutputStream thao tác tại một thời điểm do đó có thể nếu ta tạo đối tượng của lớp này trỏ đến file đang mở bởi một đối tượng lớp FileOutputStream khác thì việc tạo sẽ thất bại
- Lớp này chủ yếu dùng để ghi luồng các byte thô như dữ liệu hình ảnh, âm thanh ra file

Các phương thức

Phương thức	Mô tả
FileOutputStream(File file)	Tạo một đối tượng của lớp này để ghi luồng output vào file được đại diện bởi đối tượng file trong tham số.
FileOuputStream(File file, boolean append)	Tạo đối tượng để ghi dữ liệu ra file trong tham số thứ nhất cùng với chế độ ghi file ở tham số thứ hai.
FileOutputStream(FileDescriptor fdObj)	Tạo một đối tượng ghi file để ghi dữ liệu vào một file descriptor trong tham số.
FileOutputStream(String name)	Tạo một đối tượng ghi file để ghi dữ liệu vào file cho trong đường dẫn cung cấp bởi tham số.
FileOutputStream(String name, boolean append)	Tạo một đối tượng ghi file để ghi dữ liệu vào file cho trong đường dẫn cung cấp bởi tham số thứ nhất cùng với chế độ ghi file ở tham số thứ hai.
void close()	Đóng luồn file đầu ra và giải phóng các tài nguyên hệ thống liên kết với luồng hiện tại.
void finalize()	Dọn dẹp các kết nối đến file và đảm bảo rằng lời gọi tới phương thức close() của đối tượng hiện thời được thực hiện khi không còn bất kì tham chiếu nào đến luồng hiện thời.
FileChanel getChannel()	Trả về đối tượng FileChannel duy nhất được liên kết với luồng file đầu ra này.
FileDescriptor getFD()	Trả về file descriptor liên kết với luồng hiện tại.
void write(int b)	Ghi byte trong tham số vào luồng file đầu ra này.
void write(byte[] b)	Ghi b.length byte từ tham số vào luồng output hiện tại.
void write(byte[] b, int off, int len)	Ghi len byte từ tham số thứ nhất bắt đầu từ vị trí off vào luồng file đầu ra hiện thời.

Ví dụ

```
try {
    File file = new File(inputFile);
    FileInputStream fileInputStream = new FileInputStream(file);
    FileOutputStream fileOutputStream =
                                new FileOutputStream(outputFile);
    final int SIZE = 8 * 1024; // bộ đệm 8KB
    byte[] buffer = new byte[SIZE];
    var start = new Date().getTime();
    while ((fileInputStream.read(buffer)) != -1) {
        fileOutputStream.write(buffer);
    }
    var end = new Date().getTime();
    System.out.println("Copy file "
            + inputFile.substring(inputFile.lastIndexOf("\\") + 1)
            + " successfully!");
    System.out.println("File size: "
              + (file.length() / 1024) + "KB");
    System.out.println("Time to complete: "
              + (end - start) + "ms");
} catch (IOException e) {
    e.printStackTrace();
}
```

Lớp BufferedInputStream

- Bổ sung chức năng đệm, đánh dấu và thiết lập lại vị trí cho một luồng input
- Khi đối tượng của lớp được tạo, một mảng đệm bên trong sẽ được tạo ra
- Khi các byte từ luồng được đọc hoặc bỏ qua, bộ đệm bên trong sẽ được cấp lại nếu cần
- Thao tác đánh dấu dùng để ghi nhớ một điểm trong luồng input
- Thao tác thiết lập lại sẽ làm cho tất cả các byte đã đọc kể từ lần đánh dấu gần nhất được đọc lại

Lớp BufferedInputStream

➢Sau đây là một số phương thức và mô tả:

Trường/phương thức	Mô tả
byte[] buf	Mảng bộ đệm nội bộ nơi lưu trữ dữ liệu.
int count	Chỉ số lớn hơn giá trị chỉ số của byte hợp lệ cuối cùng trong bộ đệm.
int pos	Vị trí hiện tại trong bộ đệm.
int markpos	Giá trị của trường pos tại thời điểm lời gọi cuối đến phương thức mark() được thực hiện.
BufferedInputStream(InputStream)	Tạo một đối tượng của lớp hiện thời và lưu giá trị tham số nhận được để sử dụng về sau.
BufferedInputStream(InputStream in, int size)	Tạo đối tượng của lớp hiện thời với kích thước bộ đệm cho trước và lưu tham số thứ nhất để sử dụng về sau.
Int available()	Trả về một ước lượng của số lượng các byte có thể đọc từ luồng input mà không bị block bởi lời gọi kế tiếp trên chính luồng input này.
Void close()	Đóng luồng input và giải phóng các tài nguyên hệ thống được liên kết với nó.
Boolean markSupport()	Test xem liệu luồng input có hỗ trợ phương thức mark() và reset() không.

Lớp BufferedOutputStream

- Là lớp thực thi của luồng đầu ra. Bằng cách thiết lập luồng đầu ra với lớp này ta có thể ghi các byte vào luồng đầu ra ở tầng dưới mà không cần thực hiện lời gọi hệ thống cho mỗi byte được ghi

Trường/phương thức	Mô tả
Byte[] buf	Bộ đệm nội bộ nơi lưu trữ dữ liệu.
Int count	Số lượng byte hợp lệ trong bộ đệm.
BufferedOutputStream(OutputStream out)	Tạo đối tượng của lớp hiện thời để ghi dữ liệu.
BufferedOutputStream(OutputStream out, int size)	Tạo đối tượng của lớp hiện thời với kích thước bộ đệm cho trước trong tham số thứ hai.
Void flush()	Xóa bộ đệm(làm sạch) luồng bộ đệm đầu ra.
Void write(int b)	Ghi byte được chỉ định vào luồng bộ đệm đầu ra.
Void write(byte[] b, int off, int len)	Ghi len byte từ tham số thứ nhất bắt đầu từ vị trí off của tham số thứ hai vào luồng bộ đệm hiện tại.

Ví dụ

```
try {
    File file = new File(inputFile);
    FileInputStream fileInputStream = new FileInputStream(file);
    FileOutputStream fileOutputStream =
                                    new FileOutputStream(outputFile);
    BufferedInputStream bufferedInputStream =
                            new BufferedInputStream(fileInputStream);
    BufferedOutputStream bufferedOutputStream =
                          new BufferedOutputStream(fileOutputStream);
    final int SIZE = 8 * 1024; // bộ đệm 8KB
    byte[] buffer = new byte[SIZE];
    var start = new Date().getTime();
    while ((bufferedInputStream.read(buffer)) != -1) {
        bufferedOutputStream.write(buffer);
    }
    var end = new Date().getTime();
    System.out.println("Copy file "
            + inputFile.substring(inputFile.lastIndexOf("\\") + 1)
            + " successfully!");
    System.out.println("File size: "
              + (file.length() / 1024) + "KB");
    System.out.println("Time to complete: "
              + (end - start) + "ms");
} catch (IOException e) {
    e.printStackTrace();
}
```

Minh họa

- Thực hiện trong công cụ lập trình

Nội dung tiếp theo

Câu lệnh try-with-resource