

Lập trình Java

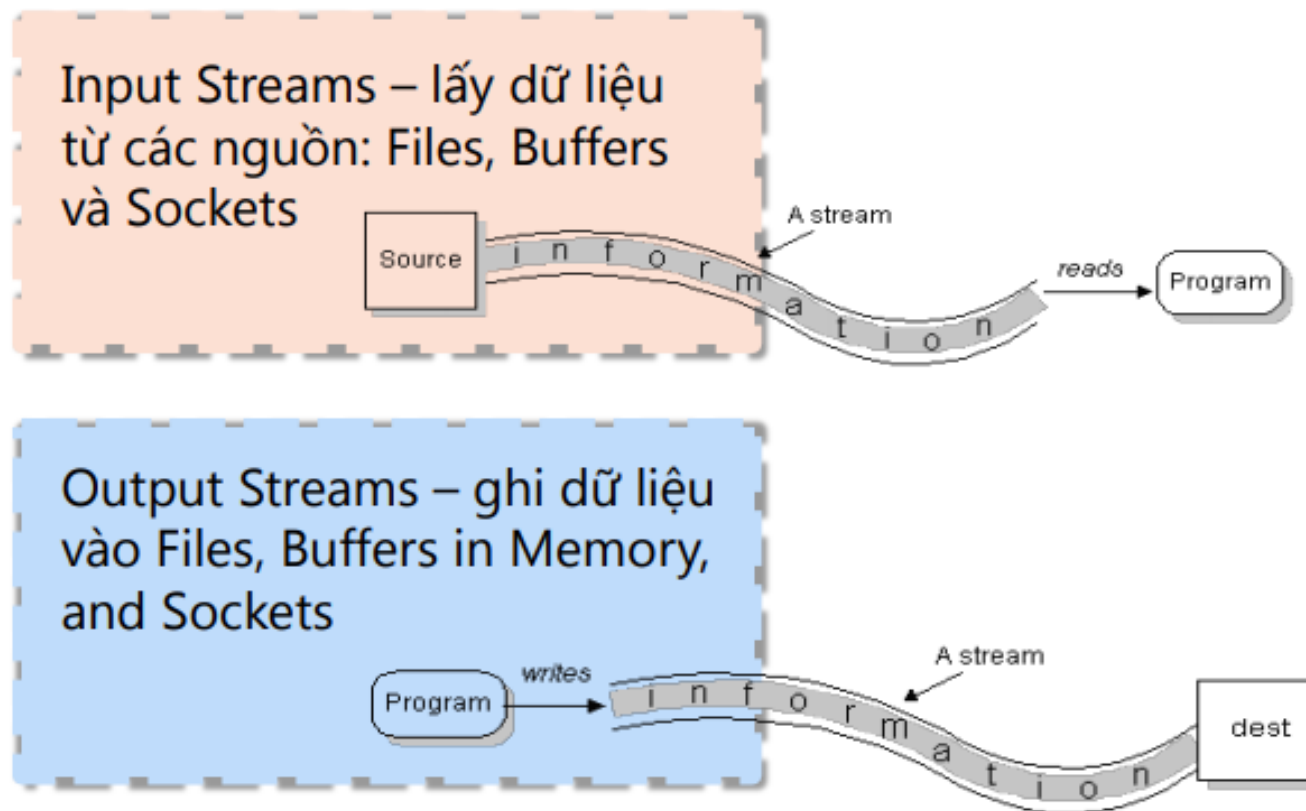
Tuần 7: File và Input, Output

Nội dung

- Các loại luồng dữ liệu, truy cập và ghi file

Các loại luồng dữ liệu

Luồng (Stream): dãy liên tục các dữ liệu, các hoạt động nhập/ xuất dữ liệu như nhập liệu từ bàn phím, đọc dữ liệu từ file có sẵn, ghi dữ liệu ra file, ra đĩa,...

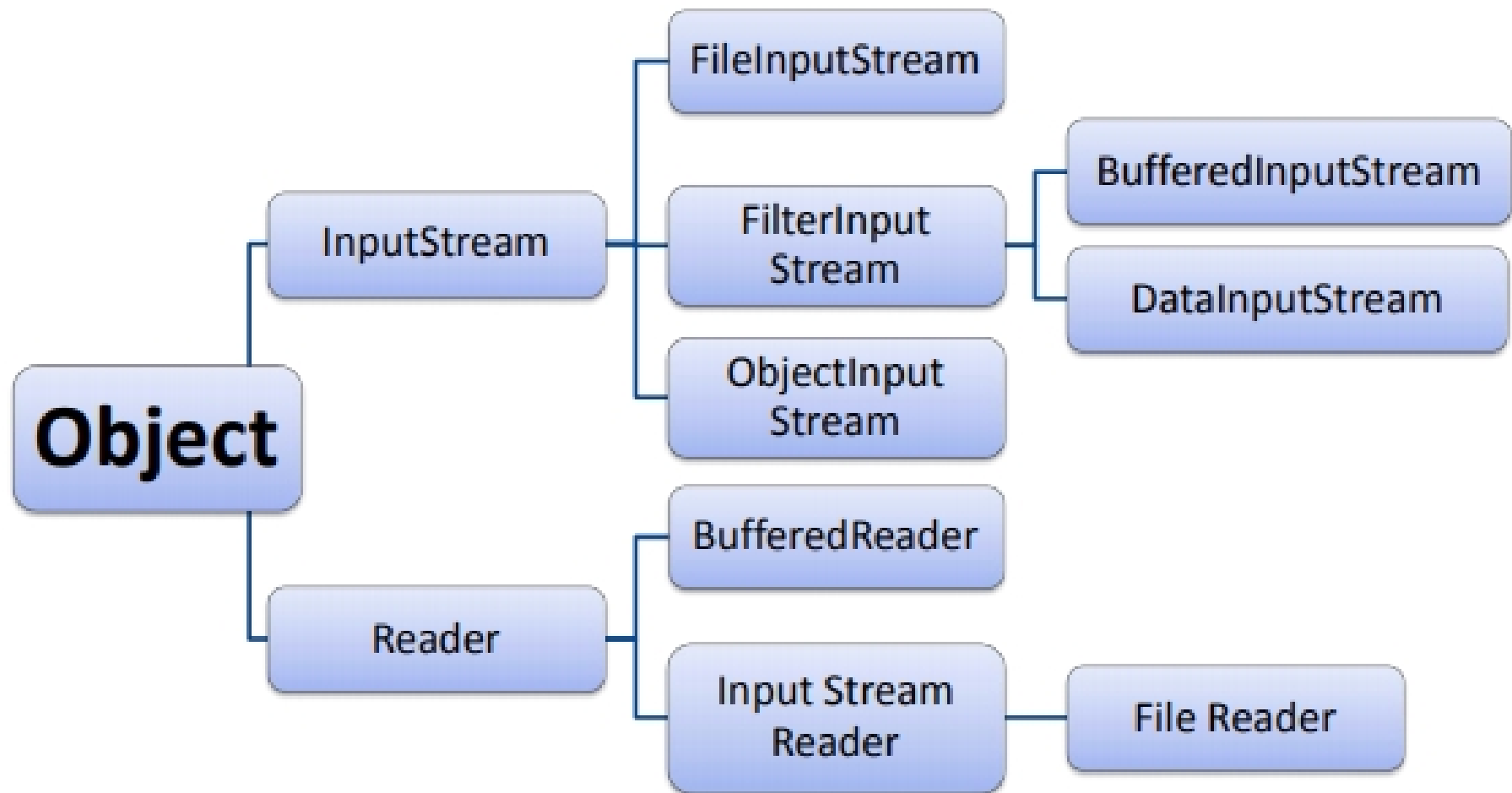


Các loại luồng dữ liệu

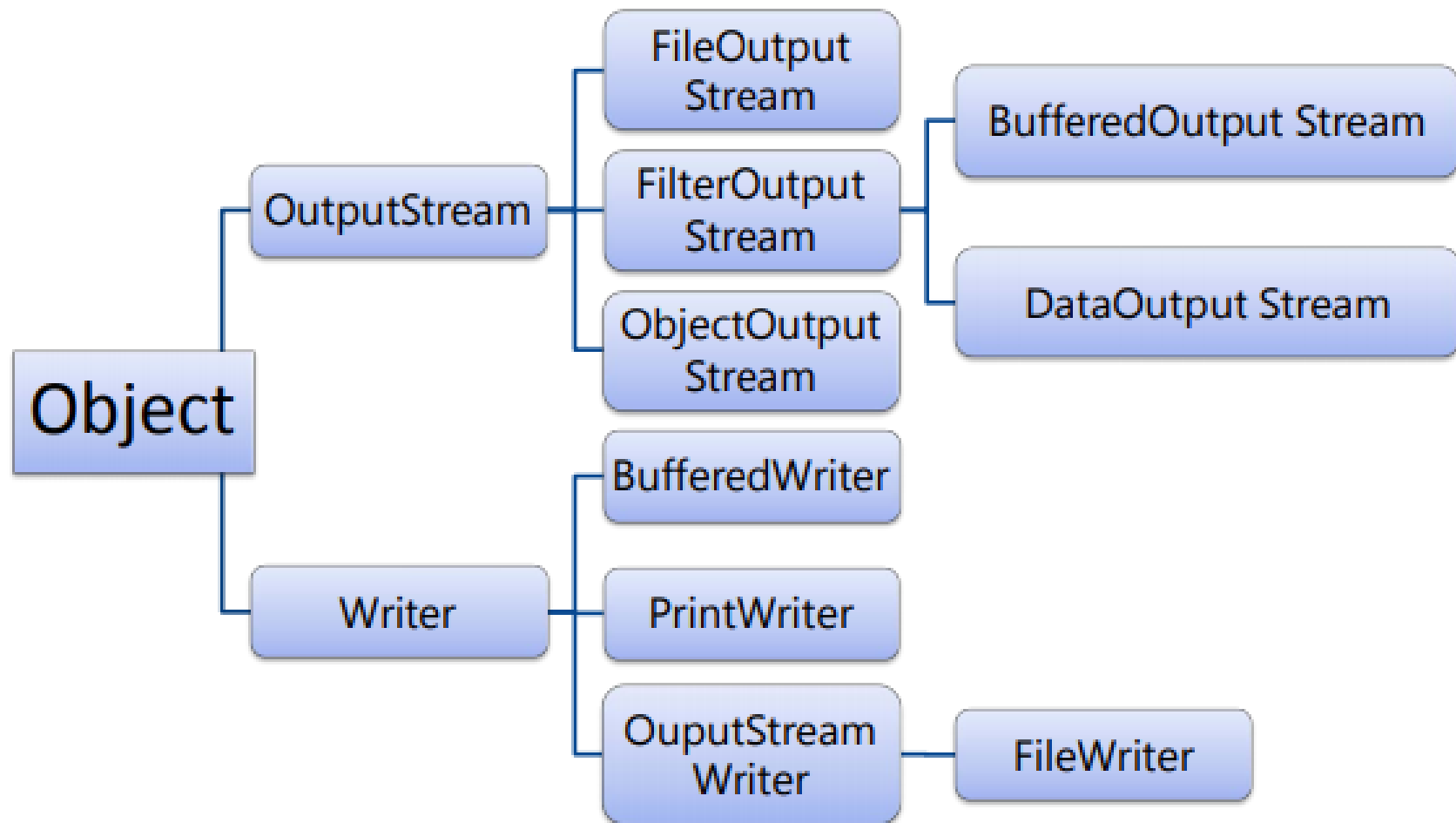
- Luồng byte: thực hiện I/O các byte (8 bit) thường dùng để đọc dữ liệu nhị phân. Lớp thường dùng là `FileInputStream` và `FileOutputStream`.
 - Luồng character: hỗ trợ việc nhập xuất dữ liệu kiểu ký tự (Unicode 16 bit). Lớp thường dùng là `FileReader` và `FileWriter`.
- => dùng `InputStream` để đọc các file (file ảnh, phim....), còn dùng `FileReader` sẽ tốt hơn khi đọc các đoạn text (file txt...)

Luồng byte	Luồng character
<ul style="list-style-type: none">▪ Dữ liệu dạng nhị phân▪ 2 class abstract:<ul style="list-style-type: none">• <code>InputStream</code>• <code>OutputStream</code>	<ul style="list-style-type: none">▪ Dữ liệu dạng ký tự Unicode▪ 2 class abstract:<ul style="list-style-type: none">• <code>Reader</code>• <code>Writer</code>

Các loại luồng dữ liệu



Các loại luồng dữ liệu



Các loại luồng dữ liệu

- Các bước xử lý:
 1. import java.io*
 2. Tạo một đối tượng Stream và liên kết với nguồn dữ liệu
 3. Đọc hoặc ghi dữ liệu
 4. Đóng Stream

Luồng byte

- Nhập xuất kiểu dữ liệu nguyên thủy: int, float, double, String, Boolean và kiểu đối tượng object.
- Ví dụ khởi tạo một đối tượng Input Stream để đọc file:

```
InputStream in = new FileInputStream("C:/java/hello");
```

Hoặc

```
File file = new File("C:/java/hello");  
InputStream in = new FileInputStream(file);
```


Luồng byte

- Ví dụ: tạo file 'filejava.txt' và ghi dữ liệu

```
import java.io*;  
public class VD1{  
    public static void main(String []args){  
        FileOutputStream fou =new FileOutputStream("filejava.txt");  
  
        String text = "Java buoi 7";  
        byte[] textbyte = text.getBytes(); // converting string into byte array  
        fou.write(textbyte);  
    }  
}
```

Luồng byte

- Ví dụ: đọc file 'filejava.txt' và in ra màn hình

```
import java.io*;
public class VD2{
    public static void main(String []args){
        FileInputputStream fin =new FileInputStream("filejava.txt");

        int c;
        While ((c=fin.read())!=-1){
            Sytem.out.print((char) c);
            Fin.close();
        }
    }
}
```

Luồng byte

- Các phương thức của InputStream

Phương thức	Mô tả
<code>public abstract int read()throws IOException</code>	Đọc byte kế tiếp của dữ liệu từ input stream. Nó trả về -1 khi đọc đến vị trí cuối tập tin.
<code>public int available()throws IOException</code>	Trả về một ước tính về số byte có thể đọc được từ input stream hiện tại.
<code>public void close()throws IOException</code>	được sử dụng để đóng input stream hiện tại.
<code>protected void finalize()throws IOException {}</code>	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

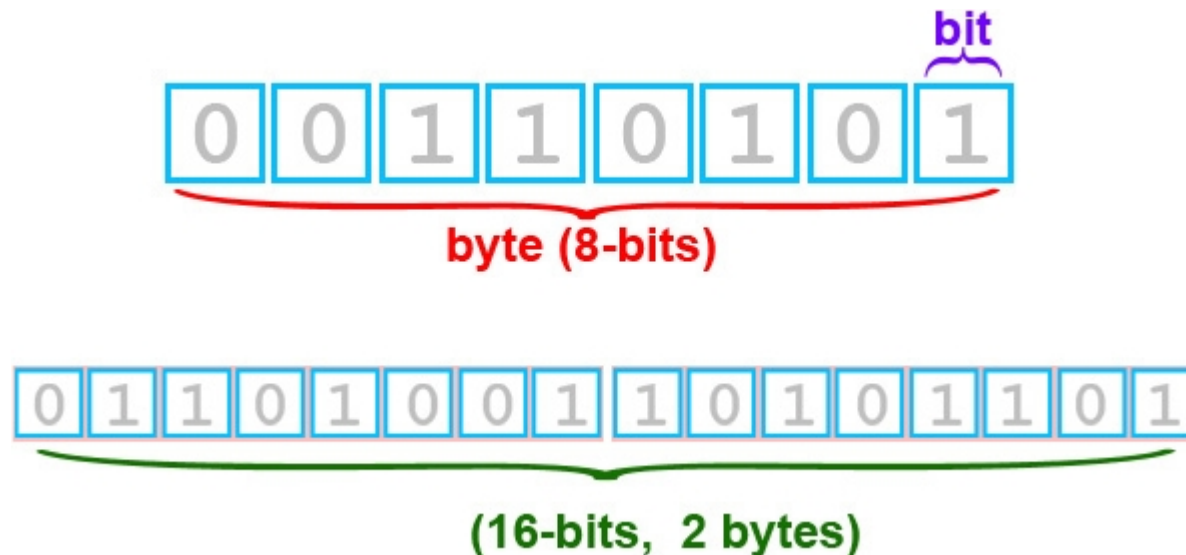
Luồng byte

- Các phương thức của OutputStream

Phương thức	Mô tả
<code>public void write(int) throws IOException</code>	được sử dụng để ghi một byte đến output stream hiện tại.
<code>public void write(byte[]) throws IOException</code>	được sử dụng để ghi một mảng các byte đến output stream hiện tại.
<code>protected void finalize() throws IOException {}</code>	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
<code>public void close() throws IOException</code>	được sử dụng để đóng output stream hiện tại.

Luồng character

- Ưu tiên khi muốn thao tác với file chứa văn bản unicode vì nó cho phép thao tác trực tiếp trên ký tự unicode.
- Được kế thừa từ 2 abstract: Reader và Writer



Luồng character

Luồng nhị phân (binary stream) mỗi một lần đọc/ghi một byte
Luồng character mỗi lần đọc/ghi một ký tự, tùy thuộc vào kiểu mã hóa (encoding) (UTF-8, UTF-16,..) mà ký tự đó tương đương với 1, 2 hoặc 3 byte

Luồng character

- Các phương thức của FileReader

Phương thức	Mô tả
<code>public int read() throws IOException</code>	Đọc một ký tự đơn. Trả về một int, mà biểu diễn ký tự đã đọc trong mẫu ASCII. Nó trả về -1 nếu không có dữ liệu.
<code>public int read(char [] c, int offset, int len)</code>	Đọc các ký tự bên trong một mảng. Trả về số ký tự đã đọc
<code>public void close()</code>	Nó được sử dụng để đóng lớp FileReader.

Luồng character

```
import java.io.*;
public class HelloReader {
    public static void main(String[] args) throws IOException {
        // Tạo một Reader (Luồng ký tự), để đọc một file.
        // Với mã hóa (encoding) mặc định.
        Reader r = new FileReader("test_reader.txt");
        int i = -1;
        while ((i = r.read()) != -1) {
            // Ép về thành kiểu ký tự.
            System.out.println((char) i);
        }
        r.close();
    }
}
```


Luồng character

- Các phương thức của FileWriter

Phương thức	Mô tả
<code>public void write(String text)</code>	Nó được sử dụng để ghi string vào FileWriter.
<code>public void write(char c)</code>	Nó được sử dụng để ghi char vào FileWriter.
<code>public void write(char[] c)</code>	Nó được sử dụng để ghi mảng char vào FileWriter.
<code>public void write(char [] c, int offset, int len)</code>	Ghi một phần của mảng các ký tự bắt đầu từ offset và với độ dài len
<code>public void write(String s, int offset, int len)</code>	Ghi một phần của String bắt đầu từ offset và với độ dài len
<code>public void flush()</code>	Nó được sử dụng để xả dữ liệu của FileWriter.
<code>public void close()</code>	Nó được sử dụng để đóng FileWriter.

Luồng character

```
import java.io.FileWriter;
public class FileWriterExample {
    public static void main(String args[]) {
        try {
            FileWriter fw = new FileWriter("D:\\testout.txt");
            fw.write("Welcome to java.");
            fw.close();
        } catch (Exception e) {
            System.out.println(e);
        }
        System.out.println("Success...");
    }
}
```

Bài tập trên lớp

Tạo class NhanVien gồm các thuộc tính:

String manv

String hoten

Int tuoi

Float luong

1. viết các phương thức setter,getter, constructor 4 tham số, toString
2. Nhập nhanvien từ bàn phím sau đó tạo file luunhanvien để lưu thông tin vào file.
3. Từ file luunhanvien.txt, thực hiện truy cập vào file và tìm nhân viên theo mã số.