cep.edu.vn

TRƯỜNG CAO ĐẲNG KINH TẾ - KẾ HOẠCH ĐÀ NẴNG
KHOA TIN HỌC & NGOẠI NGỮ

IFF

SLIDE BÀI GIẢNG
LẬP TRÌNH JAVA 2

THS. NGUYỄN ĐÌNH THÀ

Đà Nẵng – 2022

CHƯƠNG 4. SWING VÀ STREAM

MỤC TIÊU

Kết thúc bài học này, sinh viên có khả năng:

- Sử dụng thành thạo lớp FileInputStream và FileOutputStream
- Ghi được các đối tượng của lớp lên file
- Đọc dữ liệu từ từ file và hiển thị được dữ liệu lên giao diện người dùng
- Kết hợp được khi lập trình Swing và lưu trữ dữ liệu vào file

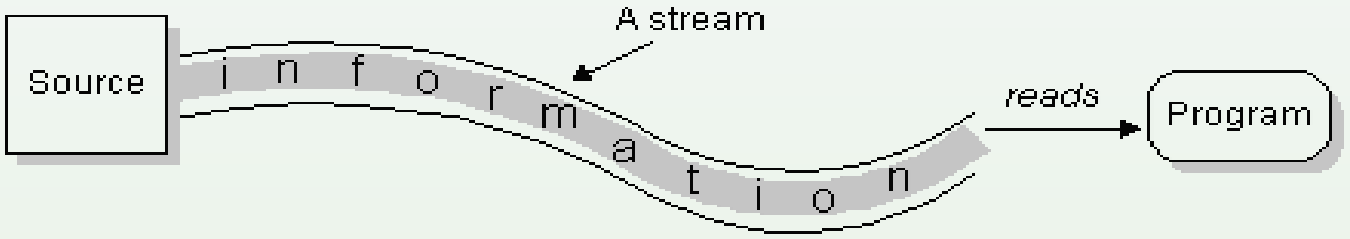
Giới thiệu luồng dữ liệu

- Các hoạt động nhập/xuất dữ liệu (nhập dữ liệu từ bàn phím, đọc dữ liệu từ file, ghi dữ liệu màn hình, ghi ra file, ghi ra đĩa, ghi ra máy in…) đều được gọi là luồng (stream).

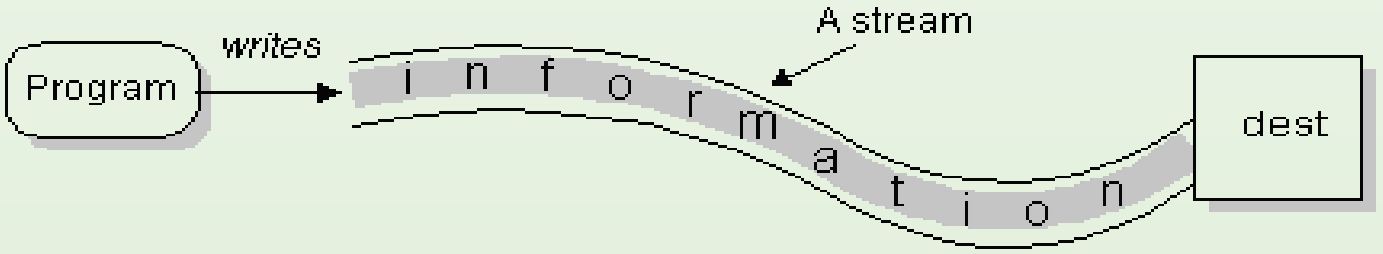
- Tất cả các luồng đều có chung một nguyên tắc hoạt động ngay cả khi chúng được gắn kết với các thiết bị vật lý khác nhau

Giới thiệu luồng dữ liệu

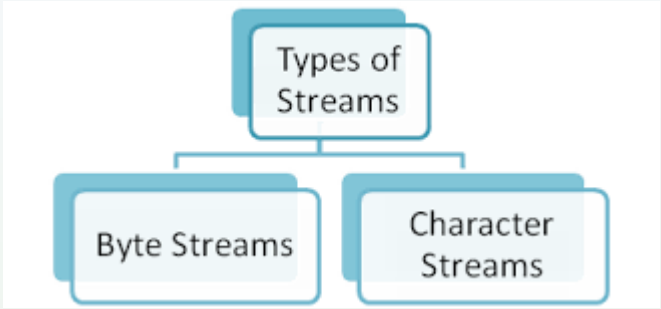
- Luồng vào là luồng cho phép chương trình đọc dữ liệu từ một nguồn nào đó: bàn phím, file, máy scan…



- Luồng ra là luồng cho phép chương trình ghi dữ liệu lên nó để chuyển đến đích nào đó: màn hình, file, máy in…



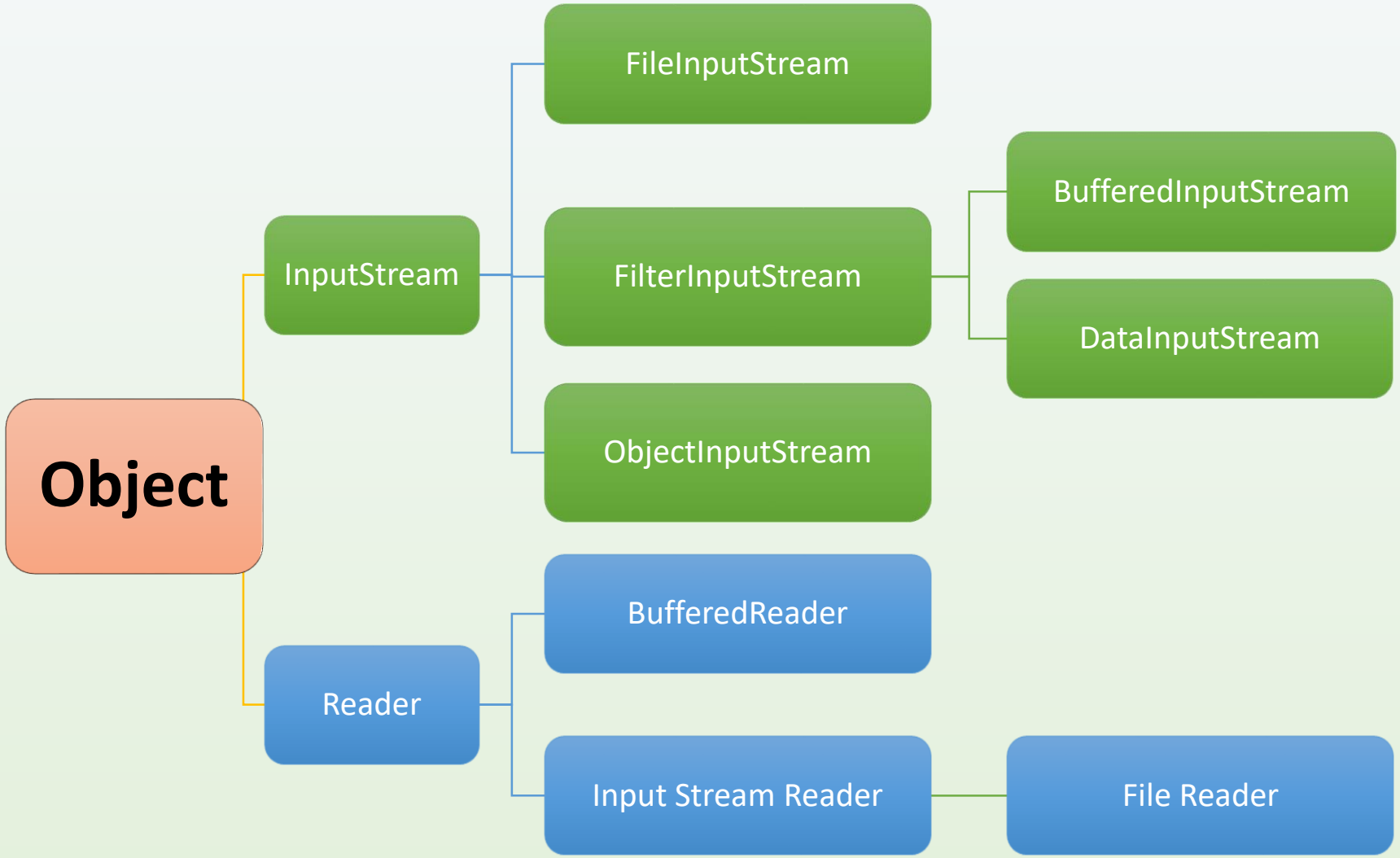
Các loại luồng dữ liệu



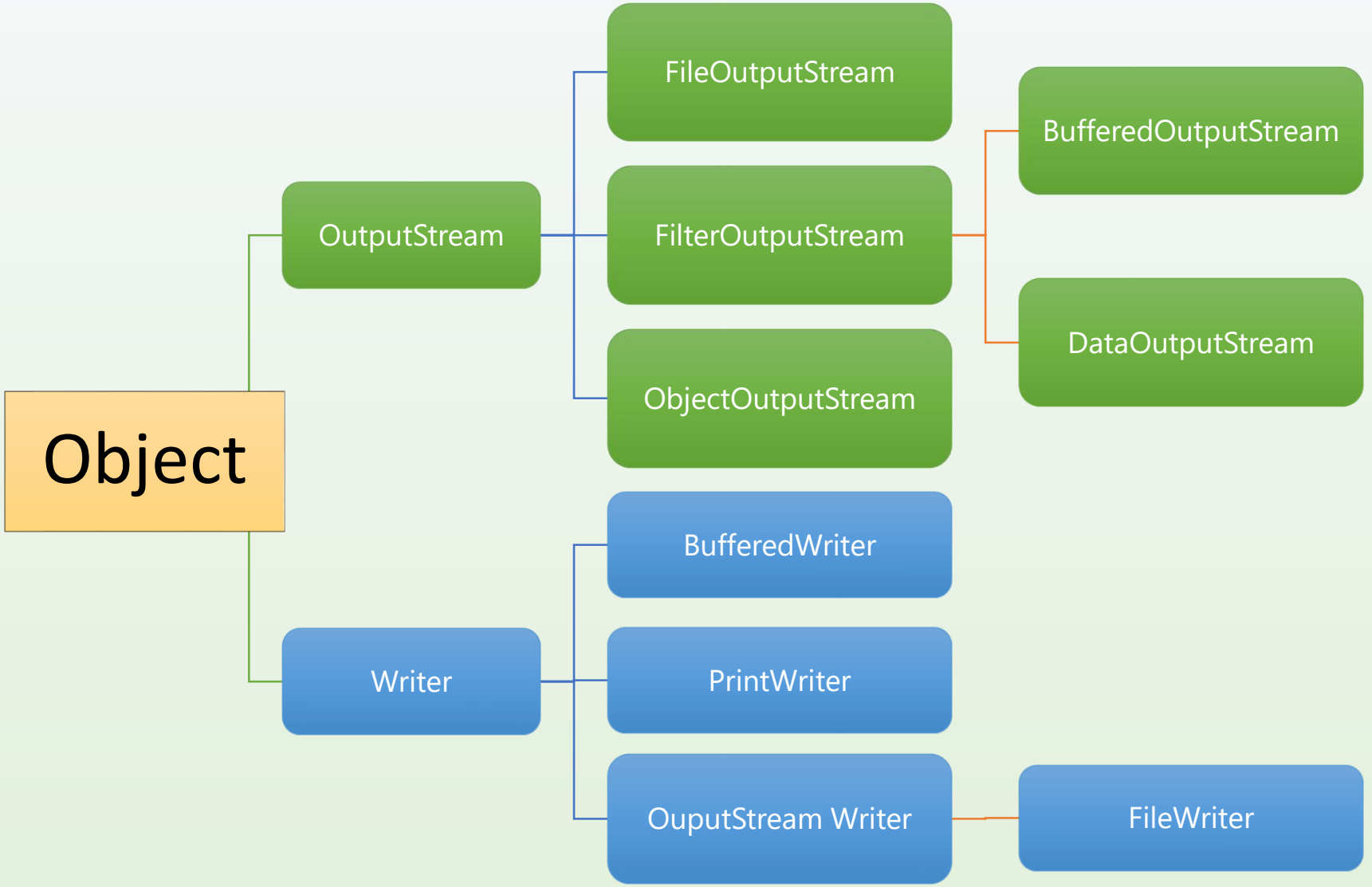
- Có 2 kiểu luồng trong Java:
  - ✓ Luồng byte (luồng nhị phân)
    - ❖ Hỗ trợ việc xuất nhập dữ liệu theo byte,
    - ❖ Thường được dùng khi đọc ghi dữ liệu nhị phân.

  - ✓ Luồng character (luồng văn bản): được thiết kế hỗ trợ việc xuất nhập dữ liệu kiểu ký tự

Kiến trúc phân cấp các loại luồng vào



Kiến trúc phân cấp các loại luồng ra



Xử lý luồng với FileInputStream/FileOutputStream

- Cặp luồng này được sử dụng để làm việc với file nhị phân
  - ✓ Sử dụng FileOutputStream để ghi dữ liệu vào file nhị phân

```
import java.io.FileOutputStream;
import java.io.IOException;

public class Example1 {

    public static void main(String[] args) throws IOException {
        FileOutputStream fos = new FileOutputStream("file1.dat");
        String text = "The quick brown fox jumped over the lazy dog";
        byte[] textAsBytes = text.getBytes();
        fos.write(textAsBytes);
    }
}
```

Xử lý luồng với FileInputStream/FileOutputStream

- Cặp luồng này được sử dụng để làm việc với file nhị phân
  - ✓ Sử dụng FileInputStream để đọc dữ liệu từ file nhị phân

```
public class Example2 {

    public static void main(String[] args) throws IOException {
        FileInputStream fis = new FileInputStream("file1.dat");
        int c;
        while ((c = fis.read()) != -1) {
            System.out.print((char) c);
        }
        fis.close();
    }
}
```

Demo1

Thực hiện code cho 2 slide ở trên

Xử lý luồng với ObjectInputStream/Object OutputStream

- Cập luồng này giúp chúng ta đọc/ghi đối tượng
- Sử dụng readObject() để đọc đối tượng từ DataInputStream
- Sử dụng writeObject(Serializable) để ghi đối tượng lên DataOutputStream
- Chú ý:
  - ✓ Chỉ các đối tượng được tạo từ các lớp có thực thi theo interface Serializable mới có thể đọc ghi được (cần phải implement).

```
public class Stock implements Serializable {
    private int id;
    private String desc;
    private double price;
    private int quantity;
    public Stock(int id, String desc, double price, int quantity){
```

```
public class Stock implements Serializable {
    private int id;
    private String desc;
    private double price;
    private int quantity;
    public Stock(int id, String desc, double price, int quantity){
        this.id = id;
        this.desc = desc;
        this.price = price;
        this.quantity = quantity;
    }
    public String toString() {
        return (id+" "+desc + " " + price + " " + quantity);
    }
}
```

Xử lý luồng với ObjectInputStream/Object OutputStream

```
public class ObjectExampleWrite {
    public static void main(String[] args) throws
            IOException, ClassNotFoundException {
        FileOutputStream fos = new FileOutputStream("fileobject.dat");
        ObjectOutputStream oos = new ObjectOutputStream(fos);
        Stock[] stocks = {new Stock(1001, "CD ROM", 100.00, 20),
            new Stock(1002, "DRAM", 75.00, 30),
            new Stock(1003, "P4 Processor", 300.00, 100),
            new Stock(1004, "Canon Jet", 80.00, 10),
            new Stock(1005, "HP Scanner", 75.00, 90)};
        //Ghi mang doi tuong vao file 'fileobject.dat'
        oos.writeObject(stocks);
        oos.close();
    }
}
```

Xử lý luồng với ObjectInputStream/Object OutputStream

```
public static void main(String[] args) {
    FileInputStream fis = null;
    ObjectInputStream ois = null;
    try {
        fis = new FileInputStream("fileobject.dat");
        ois = new ObjectInputStream(fis);
        Stock[] stocks1 = (Stock[]) ois.readObject();
        System.out.println("Doc tu file: ");
        for (Stock s : stocks1) {
            System.out.println(s);
        }
        ois.close();fis.close();

    } catch (Exception e) {
        System.out.println("Co loi: " + e);
    }
}
```

Demo2

Thực hiện code cho 2 slide ở trên

Phương thức đọc file trả về đối tượng, ghi đối tượng vào file

```
public static Object readObject(String path) {
    try {
        ObjectInputStream ois = new ObjectInputStream(new FileInputStream(path));
        Object object = ois.readObject();
        ois.close();
        return object;
    } catch (IOException | ClassNotFoundException e) {
        throw new RuntimeException(e);
    }
}
```

```
public static void writeObject(String path, Object object) {
    try {
        ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream(path));
        oos.writeObject(object);
        oos.close();
    } catch (IOException e) {
        System.out.println("Loi: " + e.getMessage());
    }
}
```

Phương thức đọc file trả về đối tượng, ghi đối tượng vào file

```
public class Student implements Serializable {

    public String name;
    public double marks;
    public String major;
```

```
public static void main(String[] args) throws IOException {

    List<Student> list = new ArrayList<>();
    list.add(new Student("Tuấn", 5, "CNTT"));
    list.add(new Student("Cường", 7.5, "DLLH"));
    list.add(new Student("Hạnh", 8.5, "CNTT"));
    XFile.writeObject("./src/java2/chuong4/students.dat", list);

    List<Student> list1 = new ArrayList<>();
    list1 = (List<Student>) XFile.readObject("./src/java2/chuong4/students.dat");

    for (Student sv: list1) {
        System.out.printf("Name: %s  Mark: %.1f  Major: %s\n", sv.name, sv.marks, sv.major);
    }
}
```

Demo3

Thực hiện code cho 2 slide ở trên

Ứng dụng:

Với giao diện giống bài tập ở chương 2, anh (chị) hãy:
Xây dựng ứng dụng đó theo các yêu cầu về CRUD dữ liệu
Dữ liệu được lưu vào file và lấy từ file ra để sử dụng.

Với giao diện giống bài tập ở chương 3, anh (chị) hãy:
Xây dựng ứng dụng đó theo các yêu cầu về CRUD dữ liệu
Dữ liệu được lưu vào file và lấy từ file ra để sử dụng.

Bài tập