
Phần 2: Ngôn ngữ lập trình C++

Chương 5: Các lớp nhập/xuất trong C++

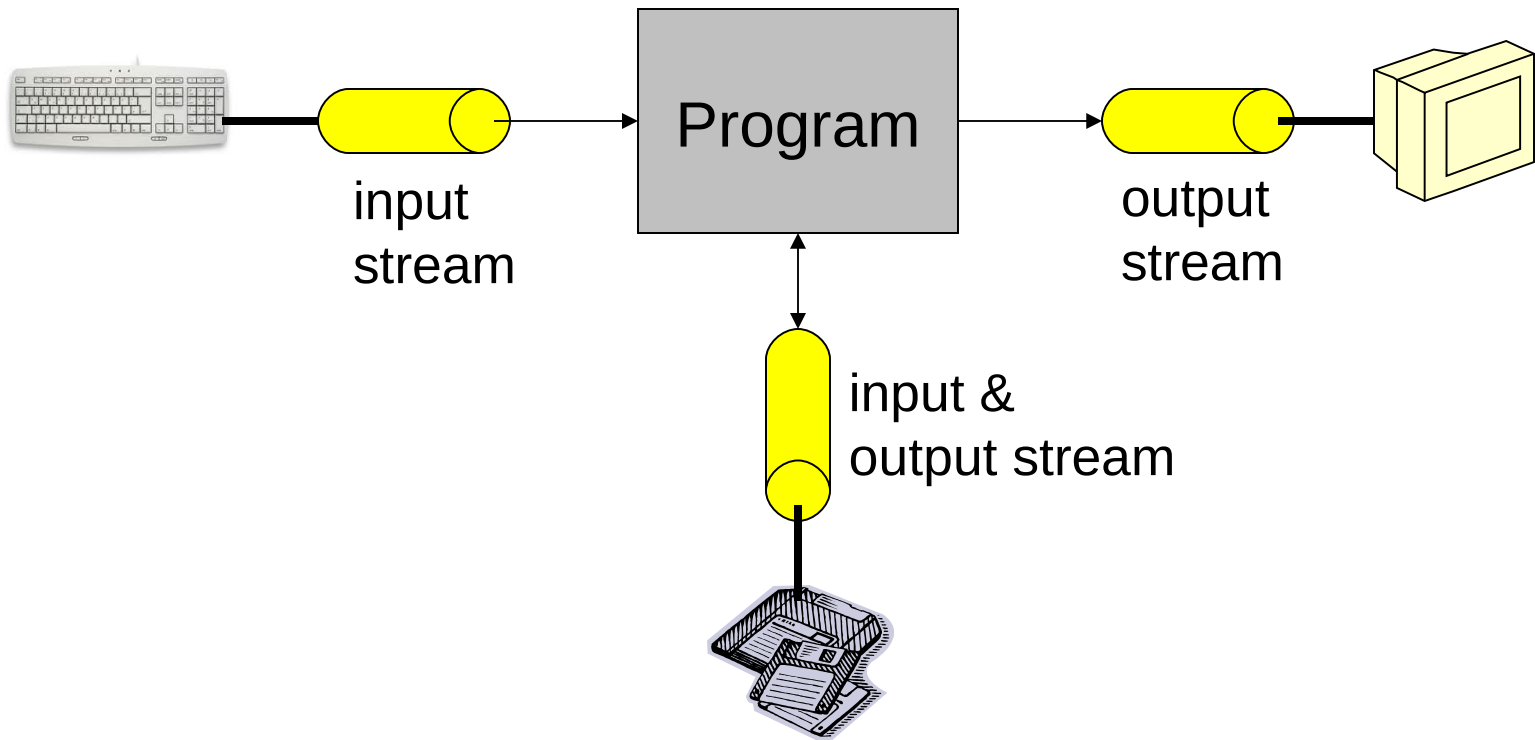
Nội dung chính

1. Giới thiệu chung
2. Các thao tác nhập/xuất cơ bản
3. Các thao tác trên tệp

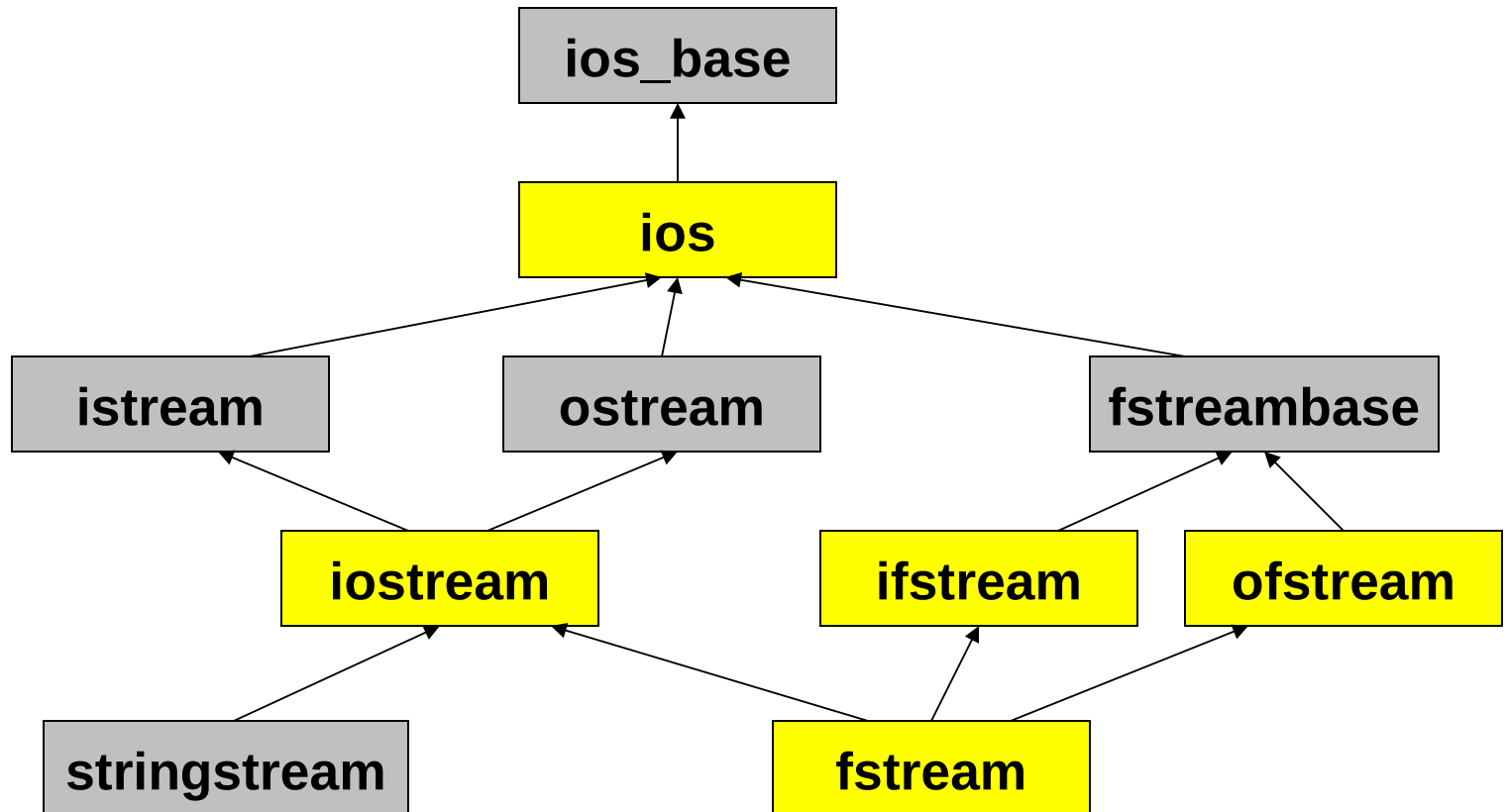
1. Giới thiệu chung

- Khái niệm **dòng** (kênh, **stream**):
 - Dòng là một sự trừu tượng hóa mà đại diện cho một thiết bị mà trên đó các thao tác nhập và xuất được thực hiện. Về cơ bản, một dòng có thể được biểu diễn như một nguồn (source) hay đích (destination) của các kí tự có độ dài vô hạn
 - Nói chung, các dòng thường kết hợp với các nguồn hay đích vật lý của các kí tự, như file, bàn phím, màn hình, v.v, do đó các kí tự được lấy ra hay được ghi vào/ra một dòng sẽ được nhập/xuất một cách vật lý đối với thiết bị vật lý.

Mô hình mô tả dòng



Giới thiệu các lớp chính thực hiện nhập/xuất



Lớp ios_base

- Lớp ios_base được thiết kế là lớp cơ sở nhất của tất cả các lớp stream trong cấu trúc phân cấp các lớp. Nó mô tả các phần cơ bản nhất của một stream, mà chung cho tất cả các đối tượng stream, nên nó không được thiết kế như một lớp có khởi tạo đối tượng, mà là một lớp trừu tượng.
- Cả lớp ios_base và lớp dẫn xuất ios của nó đều định nghĩa các thành phần của mọi stream mà độc lập với việc đối tượng stream đó là nhập hay xuất.

Lớp `ios_base`

- Cụ thể hơn, lớp `ios_base` chịu trách nhiệm duy trì bên trong nó các thông tin của một stream như sau:

Các thông tin định dạng:

- **format flags:** các cờ định dạng dùng để mô tả các thao tác nhập/xuất sẽ được thực hiện thế nào.
- **field width:** mô tả độ rộng của phần tử tiếp theo sẽ được xuất ra.
- **display precision:** mô tả độ chính xác sau dấu phẩy của dữ liệu kiểu số thực khi xuất ra.

Các thông tin trạng thái

- **error state:** phản ánh tình trạng lỗi hiện tại.
- **exception mask:** chỉ số trạng thái exception.

Lớp `ios_base`

- Đồng thời, nó cũng chứa các hàm thành viên phục vụ cho việc nhận hay thay đổi các thông tin trên như:
 - Trạng thái của những cờ **format flags** có thể có được hay bị thay đổi bằng cách gọi các hàm thành viên ***flags***, ***setf*** và ***unsetf***.
 - Giá trị **field width** có thể nhận được hay bị thay đổi nhờ việc gọi hàm thành viên ***width***.
 - Giá trị **display precision** có thể nhận được hay bị thay đổi bởi gọi hàm thành viên ***precision***.
 - Đối tượng **error state** hiện tại có thể nhận được bằng cách gọi hàm ***ios::rdstate*** và có thể thay đổi bằng cách gọi các hàm ***ios::clear*** and ***ios::setstate***. Các giá trị riêng có thể nhận được bằng cách gọi các hàm ***ios::good***, ***ios::eof***, ***ios::fail*** và ***ios::bad***.
 - Giá trị của **exception mask** có thể nhận được hoặc bị thay đổi bằng cách gọi hàm thành viên ***ios::exceptions***.

Lớp ios

- Được thiết kế là lớp cơ sở cho tất cả các lớp stream khác. Trong khi các hàm thành viên của lớp này là phụ thuộc vào kiểu dữ liệu của các tham số khuôn mẫu (type-dependent of template parameters), thì các thành viên của lớp ios_base lại là độc lập.
- Lớp này bổ sung thêm một số thông tin sau:
 - **fill character**: kí tự được dùng để điền thêm vào các khoảng trống trong *field width*. Giá trị này có thể nhận được hay bị thay đổi nhờ hàm *fill*.
 - **pointer to tied stream**: con trỏ đến đối tượng stream mà được gắn (tied) với đối tượng stream này. Giá trị này có thể nhận được hay bị thay đổi nhờ hàm thành viên *tie*.
 - **pointer to stream buffer**: con trỏ đến đối tượng *streambuf*. Giá trị này có thể nhận được hay bị thay đổi nhờ hàm thành viên *rdbuf*.

Lớp istream

- Các đối tượng của lớp istream được dùng để đọc và diễn dịch đầu vào (interpret input) từ các dãy kí tự (sequences of characters). Lớp này cung cấp các hàm thành viên để thực hiện các thao tác này, và có thể được chia thành 2 nhóm chính:

Nhập có định dạng

Các hàm này trích dữ liệu từ các dãy kí tự mà có thể được diễn dịch và định dạng. Các thao tác này được thực hiện nhờ các hàm thành viên hay các hàm toàn cục mà định nghĩa lại phép toán trích dữ liệu (***operator>>***).

Nhập không có định dạng

Đa số các hàm thành viên khác của lớp istream được dùng để thực hiện việc nhập không có định dạng. Các hàm này có thể lấy ra một số lượng xác định các kí tự từ dãy kí tự vào (***get, getline, peek, read, readsome***), rồi điều chỉnh (manipulate) việc lấy ra (***ignore, seekg, tellg, unget***) hoặc lấy thông tin về thao tác nhập không định dạng đã được gọi lần gần đây nhất (***gcount***).

- Đối tượng **cin** là một thể hiện của lớp này

Lớp ostream

- Các đối tượng của lớp **ostream** được dùng để ghi và định dạng đầu ra như các dãy kí tự. Lớp này cung cấp các hàm để thực hiện các thao tác này và chúng có thể được chia thành 2 nhóm chính:

Xuất có định dạng

Các hàm này diễn dịch và định dạng dữ liệu được ghi ra như các dãy kí tự. Các hàm này được thực hiện nhờ việc sử dụng hàm thành viên và các hàm toàn cục mà định nghĩa lại phép toán chèn (insertion operator) (**operator<<**).

Xuất không có định dạng

Đa số các hàm thành viên khác của lớp ostream được dùng để thực hiện các thao tác không định dạng. Các hàm này ghi ra một số xác định các kí tự vào dãy kí tự ra (**put**, **write**) và điều chỉnh con trỏ **put** (**seekp**, **tellp**).

Đối tượng **cout** là một thể hiện của lớp này

Lớp `fstream`

- Lớp `fstream` cung cấp một giao diện để đọc và ghi dữ liệu từ các file mà được coi như các luồng nhập/xuất.
- Các đối tượng của lớp này duy trì bên trong một con trỏ trỏ đến đối tượng **`filebuf`** mà có thể nhận được nhờ gọi hàm thành viên **`rdbuf`**.
Việc kết hợp một tệp với một đối tượng của lớp này có thể được thực hiện thông qua **`hàm tạo`** hoặc nhờ gọi hàm **`open`**.
- Sau khi tất cả các thao tác cần thiết trên file đã được thực hiện, thì nó có thể được đóng lại nhờ hàm **`close`**.

2. Các thao tác nhập/xuất cơ bản

- Các thao tác nhập dữ liệu
 - Nhập kí tự: g/s nhập kí tự đã khai báo “**char ch;**”
 - **cin>>ch;**
 - **ch = cin.get();**
 - **cin.get(ch);**
 - Nhập chuỗi kí tự: g/s nhập cho mảng kí tự đã khai báo “**char st[MAX_LEN];**”
 - **cin>>st;** Nhập chuỗi không chứa khoảng trắng
 - **cin.get(st,MAX);** Nhập chuỗi có chứa khoảng trắng
 - Nhập các kiểu dữ liệu khác:
Sử dụng **cin>>**

2. Các thao tác nhập/xuất cơ bản

■ Các thao tác xuất dữ liệu

□ Xuất kí tự

- `cout<<ch`; Xuất kí tự có định dạng

- `cout.put(ch)`; Xuất kí tự không có định dạng

□ Xuất chuỗi kí tự

- `cout<<st`; Xuất chuỗi có định dạng

- `cout.write(st,N)`; Xuất N kí tự trong st, không định dạng

□ Xuất các kiểu dữ liệu khác:

Sử dụng `cout<<`

3. Các thao tác trên tệp

- Sử dụng đối tượng thuộc lớp `fstream`

Các thao tác cơ bản gồm:

- Mở tệp
 - Dùng hàm tạo:
`fstream (const char * filename, ios_base::openmode mode = ios_base::in | ios_base::out);`
 - Dùng hàm `open`: các tham số tương tự hàm tạo
- Đọc tệp:
 - Dùng toán tử `>>` tương tự như đọc từ **`cin`**
 - Sử dụng hàm **`read`**
- Ghi lên tệp:
 - Sử dụng toán tử `<<` tương tự như ghi lên **`cout`**
 - Sử dụng hàm **`write`**

3. Các thao tác trên tệp

- Kiểm tra trạng thái mở tệp: sd hàm **fail**, nó trả về TRUE nếu việc mở tệp gặp lỗi
- Kiểm tra trạng thái khi đọc/ghi: sd hàm **bad**, nó trả về TRUE nếu đọc tệp gặp lỗi
- Kiểm tra tình trạng đọc đến cuối tệp, sd hàm **eof**, nó trả về TRUE nếu đọc đến cuối tệp

Bài tập

- **Bài 1:** Viết chương trình đọc nội dung một tệp text rồi in ra màn hình. Chương trình yêu cầu nhập tên tệp từ bàn phím, có kiểm tra sự hợp lệ của tên tệp. Nếu có lỗi không đọc được tệp thì yêu cầu nhập lại cho đến khi đọc được. Khi đó sẽ in nội dung tệp ra màn hình rồi kết thúc.
- **Bài 2:** Viết chương trình cho phép tìm và thay thế một chuỗi kí tự có trong một tệp văn bản bằng một chuỗi kí tự khác.
 - Đầu vào: tên tệp fname, chuỗi cần tìm, chuỗi cần thay thế.
 - Đầu ra: tệp fname với nội dung đã được thay thế, và số lần thay thế.

Bài tập

- Bài 3: Viết chương trình copy một tệp sang một thư mục mới. Chương trình cho phép đổi tên tệp cũ khi copy.
 - Đầu vào: tên tệp muốn copy, thư mục mới (có thể trùng với thư mục của tệp muốn copy, nhưng khi đó thì phải có tên tệp mới), tên tệp mới (không bắt buộc, tên mặc định là tên tệp muốn copy)
 - Đầu ra: tệp mới sau khi đã copy
- Bài 4: Xây dựng một lớp MergeSplitFile, cho phép thực hiện việc trộn hai hay nhiều tệp thành một tệp, hoặc ngược lại tách một tệp đã bị trộn về các tệp ban đầu. Cụ thể lớp sẽ có các thao tác sau:
 - Nhập vào một danh sách các tệp hoặc thư mục chứa các tệp cần trộn
 - Trộn các tệp trên thành một tệp mới
 - Đọc nội dung một tệp đã trộn để lấy ra các thông tin về các tệp đã được trộn trong đó
 - Tách tệp đã trộn về các tệp ban đầu.

Thank you!