

# TIN ĐẠI CƯƠNG

## NHẬP - XUẤT FILE, STRUCT

**Trần Thị Ngân**

Bộ môn Công nghệ phần mềm, Khoa CNTT

Trường đại học Thủy Lợi

# Nội dung chính

1. Thao tác với các tập tin

2. Bài tập

## Sự cần thiết sử dụng file trong C++

- Câu lệnh **cin** : nhập dữ liệu từ bàn phím và đưa vào các biến
- Câu lệnh **cout** : hiển thị giá trị các biểu thức ra màn hình
- Khi kết thúc chương trình thì các biến, các biểu thức không còn nữa → dữ liệu bị mất
- Mong muốn của người sử dụng: có thể lưu trữ các biến, các kết quả tính toán ra tập tin (file), nhờ vậy kết quả tính toán sẽ không bị mất và có thể đọc dữ liệu có trong file để xử lý tiếp
- → giải pháp : thao tác với các tập tin (cụ thể là dạng văn bản)

## 1. Thao tác với các tập tin

- C++ cung cấp các lớp (class) sau để đọc - ghi dữ liệu với file

`ofstream` : ghi dữ liệu vào file

`ifstream` : đọc dữ liệu từ file

`fstream` : cả đọc và ghi file

- Các lớp này được định nghĩa trong thư viện `fstream` và đặt trong không gian tên `std`

→ khai báo thư viện `fstream` ở đầu chương trình :

```
#include <fstream>
```

## Ghi ra file bằng ofstream

- ▲ Tạo đối tượng và gắn nó với một file trong thiết bị lưu trữ
- ▲ Khi thay `cout` bởi tên đối tượng, dữ liệu sẽ được ghi vào file đại diện bởi đối tượng đó, thay vì in ra màn hình
- ▲ **Cách 1** : tạo một đối tượng chưa gắn với tập tin cụ thể nào, sau đó dùng hàm `open` để mở một tập tin và gắn nó với đối tượng vừa tạo

Ví dụ :

```
#include <fstream>
using namespace std;
int main()
{
    ofstream ofs; //khai bao ofs la mot doi tuong cua lop ofstream
    ofs.open("thudo.txt"); //mo file thudo.txt de ghi du lieu va gan voi ofs
    ofs << "Madrid" << endl; //ghi dong chu "Madrid" vao thudo.txt
    ofs << "Paris" << endl; //ghi dong chu "Paris" vao thudo.txt
    ofs.close(); //dong file thudo.txt, giai phong doi tuong
    return 0;
}
```

## Ghi ra file bằng ofstream

- **Cách 2** : mở một tập tin đồng thời gắn nó với tên đối tượng

Ví dụ : `ofstream ofs("thudo.txt") ;`

- Chú ý : sau khi thao tác xong, sử dụng hàm `close()` để đóng file và giải phóng đối tượng
- Kiểm tra việc mở tập tin
  - Sử dụng hàm `is_open()` : trả về giá trị true nếu việc mở file thành công
  - Sử dụng hàm `fail()` : trả về giá trị true nếu việc mở file thất bại

## Đọc file bằng ifstream

- tương tự như ghi file, có 2 cách

ifstream ifs;

ifs.open("thudo.txt");

Và

ifstream ifs("thudo.txt");

Ví dụ :

```
#include <fstream>
#include <string>
#include <iostream>
using namespace std;
int main()
{
    ifstream ifs("thudo.txt"); //mo file thudo.txt de doc du lieu va gan voi ifs
    if (ifs.is_open())
    {
        string dong;
        while (!ifs.eof()) //kiem tra xem da den dong cuoi cung cua file chua
        {
            getline(ifs, dong); //doc tung dong, thay cin boi ten doi tuong ifs
            cout << dong << endl; //va in dong do ra man hinh
        }
        ifs.close(); //dong file thudo.txt, giai phong doi tuong
    }
    return 0;
}
```

## Đọc và ghi file bằng fstream

Ghi file : sử dụng fstream tương tự như ofstream, chỉ lưu ý khi mở file để ghi thì thêm **fstream : :out**

Mở file để ghi dùng  
ofstream

```
ofstream ofs;  
ofs.open("thudo.txt");  
ofs << "Madrid" << endl;  
ofs << "Paris" << endl;  
ofs.close();
```

Mở file để ghi dùng  
fstream

```
fstream ofs;  
ofs.open("thudo.txt", fstream::out);  
ofs << "Madrid" << endl;  
ofs << "Paris" << endl;  
ofs.close();
```



## Đọc và ghi file bằng fstream

Đọc file: sử dụng fstream tương tự như ifstream, chỉ lưu ý khi mở file để đọc thì thêm `fstream::in`

### Mở file để đọc

```
ifstream ifs;  
ifs.open("hello.txt");  
string dong;  
while (!ifs.eof())  
{  
    getline(ifs, dong);  
    cout << dong << endl;  
}  
ifs.close();
```

```
fstream ifs;  
ifs.open("hello.txt", fstream::in);  
string dong;  
while (!ifs.eof())  
{  
    getline(ifs, dong);  
    cout << dong << endl;  
}  
ifs.close();
```

## 2. Bài tập

### Bài 1

Viết ra file `so_chan.txt` tất cả các số chẵn từ 0 đến 30 (mỗi số trên 1 dòng). Sau đó đọc từ file `so_chan.txt` trên, thêm 1 vào mỗi giá trị, viết kết quả vào file `so_le.txt`.

### Bài 2

Đọc 2 file `taptin1.txt` và `taptin2.txt`. Tạo ra file `taptin3.txt` có nội dung bắt đầu bằng nội dung của `taptin1.txt` sau đó là nội dung của `taptin2.txt`.

# KIỂU DỮ LIỆU STRUCT

## Sự cần thiết

Ví dụ : Trong kì thi tuyển sinh ĐH, phòng đào tạo phải nhập dữ liệu cho các học sinh, bao gồm:

- Họ đệm, Tên, Địa chỉ (các chuỗi kí tự)
- Điểm 3 môn xét tuyển, Tổng điểm (các số thực)

Sau đó phải sắp xếp Tổng điểm theo thứ tự giảm dần

Tất cả các dữ liệu khác như Tên, Điểm các môn cũng phải thay đổi thứ tự theo thứ tự của Tổng điểm

- Giải pháp: các dữ liệu Họ đệm, Tên, Điểm các môn, Tổng điểm... luôn đi kèm với nhau khi sắp xếp
- Xây dựng một kiểu dữ liệu mới bao gồm tất cả các dữ liệu kể trên

### 3. Kiểu dữ liệu tự tạo

- C++ cung cấp một số kiểu dữ liệu cơ bản (số nguyên, số thực, logic, ...)
- C++ cũng cho phép kết hợp các dữ liệu cơ bản để xây dựng các kiểu dữ liệu phức tạp hơn. Ví dụ:
  - Học sinh : Họ đệm, Tên, Địa chỉ, Điểm 3 môn, Tổng điểm
  - Điểm trên mặt phẳng : tọa độ x, tọa độ y

## Kiểu dữ liệu tự tạo

- Ngoài khai báo dữ liệu, ta còn có thể định nghĩa các phép toán, hàm. . . đi kèm với kiểu dữ liệu đó

Ví dụ : trên kiểu dữ liệu Học sinh, viết hàm tính Tổng điểm, xây dựng toán tử so sánh. . .

- Nhiều kiểu dữ liệu tự tạo được sử dụng rất phổ biến (string, vector. . .)

## 4. Khai báo và sử dụng struct

Cú pháp :

```
struct <tên kiểu> {  
    <các dữ liệu thành phần>  
};
```

Ví dụ : Khai báo kiểu dữ liệu struct

```
struct Point {                //kiểu dữ liệu struct tên là Point  
    double x, y;              //các dữ liệu thành phần  
};
```

Sử dụng:

```
Point diemA;                  //một biến kiểu Point  
diemA.x = 2.5;                //thành phần x = 2.5  
diemA.y = 1.8;                //thành phần y = 1.8
```

## 5. Bài tập

Tự tạo các kiểu dữ liệu mới sau đây.

1. Kiểu dữ liệu **Point** mô tả một điểm trên mặt phẳng toạ độ
2. Kiểu dữ liệu **Line** mô tả một đoạn thẳng trên mặt phẳng toạ độ (gồm hai điểm đầu và cuối)
3. Kiểu dữ liệu **TamGiac** mô tả một tam giác trên mặt phẳng toạ độ (gồm những thành phần gì?)
4. Kiểu dữ liệu **GiáoViên** lưu trữ thông tin về các giáo viên trong trường, gồm : họ tên, địa chỉ, số điện thoại, năm vào trường
5. Kiểu dữ liệu **SinhViên** lưu trữ thông tin về các sinh viên trong trường (gồm những thành phần gì?)