

KỸ THUẬT LẬP TRÌNH

Chương 6: Tập tin & luồng

Mục tiêu

► Sau khi học xong chương này, người học có thể:

1 Hiểu được ý nghĩa và cách sử dụng tập tin trong lập trình

2 Biết cách vận dụng đọc và ghi tập tin trong bài toán cụ thể



Nội dung



Tập tin
(file)

1

Giới thiệu

2

Quy trình làm việc với tập tin

3

Ghi tập tin

4

Đọc tập tin

5

Ví dụ minh họa

1. Giới thiệu
2. Quy trình làm việc với tập tin

3. Ghi tập tin
4. Đọc tập tin

5. Ví dụ



1. Giới thiệu
2. Quy trình làm việc với tập tin

3. Ghi tập tin
4. Đọc tập tin

5. Ví dụ

-
1. Tạo đối tượng nhập/xuất tập tin
 2. Mở tập tin
 3. Ghi dữ liệu ra tập tin
 4. Đọc dữ liệu từ tập tin
 5. Đóng tập tin



1. Tạo đối tượng nhập/xuất tập tin

Thêm chỉ thị: `#include <fstream>`

Thư viện `fstream` chứa class `ifstream` (input file stream) và `ofstream` (output file stream)

Cú pháp khai báo đối tượng nhập dữ liệu từ file (đọc file):

```
ifstream fileObject;
```

Cú pháp khai báo đối tượng xuất dữ liệu ra file (ghi file):

```
ofstream fileObject;
```

Trong đó:

- `ifstream`: tên class của đối tượng đọc file
- `ofstream`: tên class của đối tượng ghi file
- `fileObject`: tên đối tượng được tạo ra

1. Tạo đối tượng nhập/xuất tập tin

Thêm chỉ thị: `#include <fstream>`

Thư viện fstream chứa class ifstream (input file stream) và ofstream (output file stream)

Cú pháp khai báo đối tượng nhập dữ liệu từ file (đọc file):

```
ifstream fileObject;
```

Cú pháp khai báo đối tượng xuất dữ liệu ra file (ghi file):

```
ofstream fileObject;
```

Ví dụ:

```
ifstream inNhanVien;
```

```
ofstream outSinhVien;
```

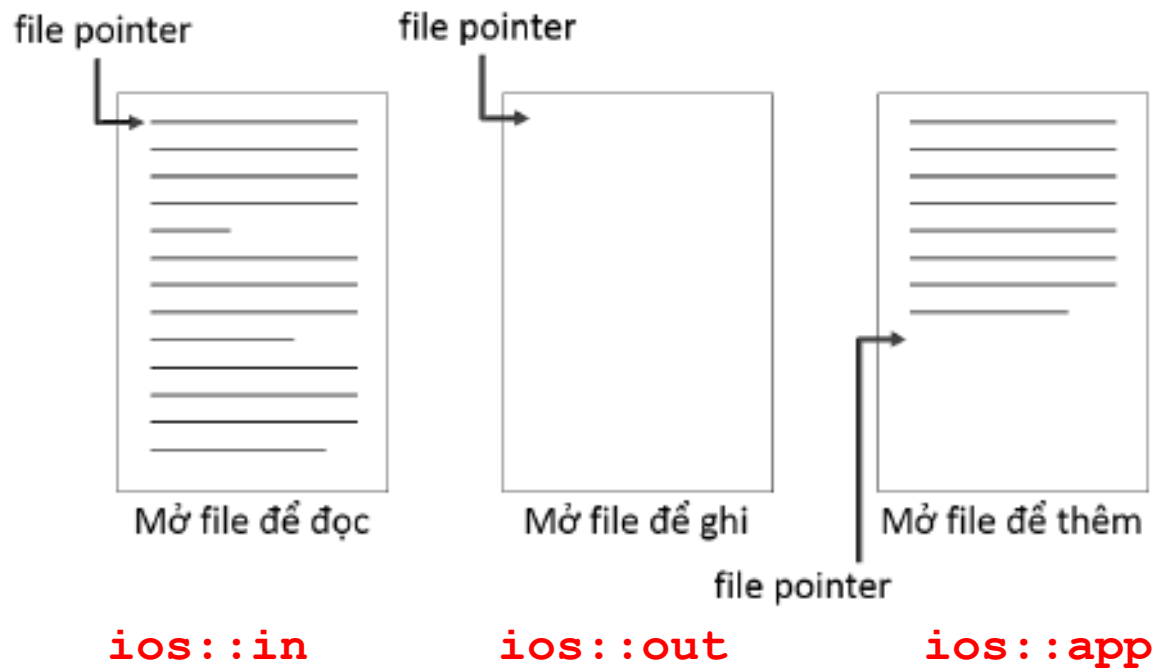
2. Mở tập tin

```
fileObject.open(fileName [, mode]);
```

- ✓ **fileObject**: tên đối tượng đã tạo ở bước 1 (**ifstream** hoặc **ofstream**)
- ✓ **fileName**: tên file cần làm việc, là chuỗi hoặc hằng chuỗi (có thể cung cấp đường dẫn).
- ✓ **mode**:
 - ios::in** : dùng với **ifstream**, mở file để đọc (mặc định của file đọc)
 - ios::out** : dùng với **ofstream**, mở file để ghi, nếu đã có sẽ xóa (mặc định của file xuất)
 - ios::app** : dùng với **ofstream**, mở file để ghi thêm ở cuối file (file chưa có sẽ tạo mới)

2. Mở tập tin

```
fileObject.open(fileName [, mode]);
```



2. Mở tập tin

```
fileObject.open(fileName [, mode]);
```

Ví dụ:

- Mở file để đọc dữ liệu

```
inDoanhThu.open("doanhthu.txt", ios::in);
```

Hoặc `inDoanhThu.open("doanhthu.txt");`

- Mở file để ghi dữ liệu

```
outNhanVien.open("nhanvien.txt", ios::out);
```

Hoặc `outNhanVien.open("nhanvien.txt");`



2. Mở tập tin

```
fileObject.open(fileName [, mode]);
```

Ví dụ:

- Mở file để ghi vào cuối file:

```
outNhanVien.open("D:/taptin/nhanvien.txt", ios::app);
```

3. Ghi dữ liệu ra tập tin

4. Đọc dữ liệu từ tập tin

Trước khi đọc/ghi dữ liệu ra file nên kiểm tra tình trạng làm việc của file:

```
fileObject.is_open() ;
```

Trả về kết quả True/False.

Ví dụ:

```
if ( outFile.is_open() )
{
    cout << "Mo file thanh cong! Co the tiep tục doc/ghi du
lieu" << endl;
    //Tiếp tục đọc/ghi file
}
else
    cout << "Khong mo duoc file de ghi du lieu" << endl;
```

5. Đóng tập tin

```
fileObject.close();
```

fileObject: tên đối tượng file

Ví dụ:

```
inNhanVien.close();
```

//Khung chương trình làm việc với tập tin

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
    ofstream outFile;           //dành cho ghi file
    ifstream inFile;           //dành cho đọc file
    //outFile.open("tênfile+vị trí", mode);
    //hoặc inFile.open("tênfile+vị trí", mode);
    if (outFile.is_open() hoặc inFile.is_open() )
    {
        //Ghi file hoặc đọc file
        outFile.close(); hoặc inFile.close();

    }
    else
        cout << "Khong mo duoc file de ghi du lieu" << endl;
}
```

▶ Cú pháp ghi 1 giá trị ra file

```
fileObject << data;
```

Ví dụ:

```
outNhanVien << "Nguyen Van An" << endl;
```

//sẽ tự động chuyển sang dòng mới để ghi tiếp

```
int b = 2;
```

```
outNhanVien << "Nguyen Van An" << '#' << b << endl;
```

//sẽ ghi dữ liệu kèm dấu # xem như phân cách 2 fields, tự động chuyển sang dòng mới để ghi tiếp

▶ Cú pháp đọc giá trị từ file

```
fileObject >> variableName;
```

Hoặc đọc từ chuỗi:

```
getline(fileObject, strName [, delimCharater] );
```

Trong đó:

- **fileObject**: tên đối tượng ifstream
- **variableName, strName**: tên biến lưu dữ liệu đọc được từ file
- **delimCharater**: ký tự kết thúc, mặc định là newline

► Cú pháp đọc giá trị từ file

```
fileObject >> variableName;
```

Hoặc đọc từ chuỗi:

```
getline(fileObject, strName [, delimCharater] );
```

Ví dụ:

```
int nam, luong;
```

```
inLuong >> nam;
```

```
inLuong.ignore(1); //bỏ qua #
```

```
inLuong >> luong;
```

```
inLuong.ignore(1); //bỏ qua newline
```

số năm làm việc	tiền lương
13	54000
25	83000
5	36000
2	20500

➤ Đọc 1 ký tự đã được ghi xuống file trước đó

```
char kt = ' ';
```

```
inFile >> kt;
```

➤ Đọc 1 chuỗi

```
string diachi = " ";
```

```
int sonha;
```

```
getline(infile, diachi, '#'); //bỏ qua dấu cách giữa 2 fields
```

```
inFile >> sonha;
```

```
inFile.ignore(1); //bỏ qua newline
```



➤ Nên kiểm tra giá trị tại dòng mà con trỏ trỏ đến khi nào là kết thúc?

Sử dụng:

`fileObject.eof()`

Trả về True nếu con trỏ đang trỏ đến cuối file

Trả về False nếu con trỏ chưa trỏ đến cuối file (còn dữ liệu)

Ví dụ:

```
while ( inFile.eof() == false )  
{  
  
    //đọc file  
  
}
```

-
- Viết chương trình nhận vào giá trị 3 số nguyên. Sau đó ghi xuất file giá trị 3 biến này dưới dạng cách nhau dấu `-'.
 - Đọc lại 3 giá trị này vào 3 biến khác nhau, tính và xuất kết quả tích 3 số ra màn hình.

```
#include <iostream>

#include <fstream>

using namespace std;

int main()
{
    ofstream outFile;
    ifstream inFile;
    outFile.open("BTVD.txt", ios::app);

    int a, b, c;
    int kq;
    cout << "Nhap 3 so nguyen: ";
    cin >> a >> b >> c;
```



//Ghi giá trị 3 số nguyên ra file cách nhau bằng dấu '-'

```
if (outFile.is_open())  
{  
    outFile << a << '-' << b << '-' << c << endl;  
    outFile.close();  
    cout << "Ghi file thanh cong!" << endl;  
}  
else  
    cout << "Khong mo duoc file de ghi du lieu" << endl;
```

```
//Đọc giá trị 3 số nguyên từ file cách nhau bằng dấu '-' và tính tích
inFile.open("BTVD.txt");
if (inFile.is_open())
{
    inFile >> a;
    inFile.ignore(1);
    inFile >> b;
    inFile.ignore(1);
    inFile >> c;
    kq = a * b * c;
    cout << "Ket qua tinh la: " << kq << endl;
    inFile.close();
}
else
    cout << "Khong mo duoc file de doc du lieu" << endl;
}
```

Bài tập

1. Viết chương trình nhập vào giá trị họ và tên kèm với năm sinh của 1 người. Lưu trữ 2 thông tin này xuống file làm việc. Đảm bảo mỗi thời điểm file chỉ lưu trữ thông tin 1 người.
2. Đọc dữ liệu từ file đã ghi ở BT1, sau đó xuất lại thông tin họ tên kèm với tính tuổi của người đó.
3. Phát triển BT1 và BT2 thành hàm ghi và đọc dữ liệu.
4. Viết chương trình xây dựng các hàm cho phép:
 - Nhập vào 2 chuỗi, mỗi chuỗi lưu ở 1 file khác nhau.
 - Đọc 2 chuỗi từ 2 file, tiến hành so sánh, nối 2 chuỗi lại và xuất kết quả.
 - Đọc 1 chuỗi từ 1 file bất kỳ, đếm xem chuỗi có bao nhiêu nguyên âm.
 - Hàm main kiểm chứng với các hàm ở cả 2 chuỗi.

Q & A