

LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG HƯỚNG DẪN THỰC HÀNH

Luyện tập với file – getter – đa hình

Giáo viên lý thuyết:



ThS Lê Xuân Định

lxding@fit.hcmus.edu.vn

Giáo viên thực hành:



ThS Lê Xuân Định

lxding@fit.hcmus.edu.vn



ThS Trần Huy Quang

thquang@fit.hcmus.edu.vn



ThS Trần Duy Quang

tdquang@fit.hcmus.edu.vn



Trần Minh Khoa

tranminhkhoa1402@gmail.com

1 Mục tiêu

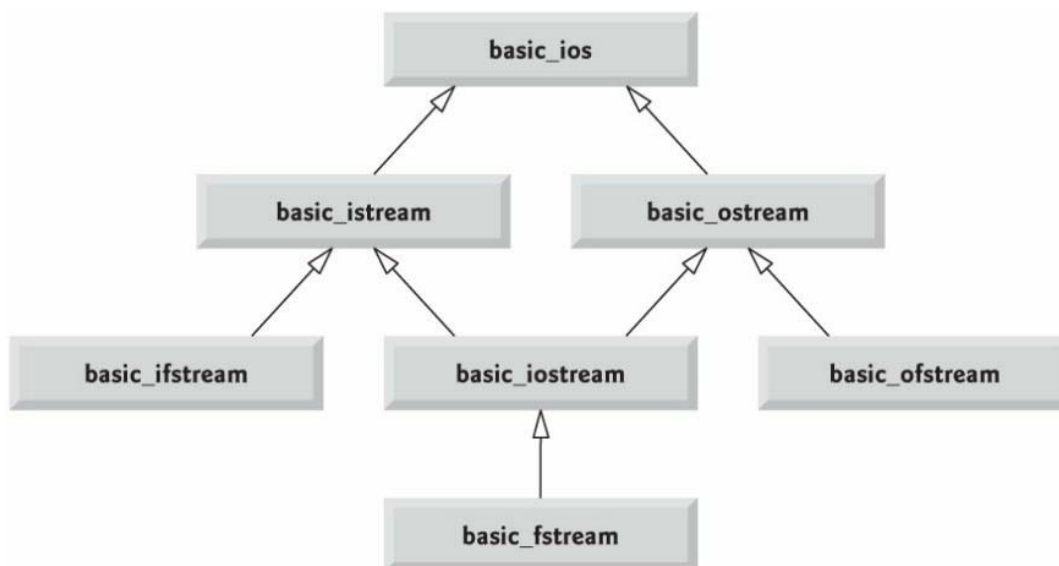
- Làm quen với các luồng nhập xuất tập tin văn bản.
- Viết lớp đơn giản nhất có construct và getter.
- Ứng dụng tính đa hình để quản lý chung các đối tượng thuộc các lớp khác nhau.

2 Tập tin trong C++

C++ xem tập tin là một chuỗi tuần tự các byte (còn được gọi là dòng dữ liệu – stream). Một tập tin sẽ kết thúc với một kí tự đánh dấu end-of-file hoặc tùy thuộc vào cấu trúc dữ liệu của hệ điều hành biểu diễn tập tin.



Hình 1 – Góc nhìn của C++ đối với một tập tin n byte [1]



Hình 2 – Một phần cấu trúc phân cấp của cây kế thừa theo mẫu nhập xuất [1]

Tutorial 1: Làm quen với việc ghi dữ liệu ra tập tin text.

Yêu cầu: Viết chương trình nhập vào thông tin ngân hàng của 5 khách hàng, lưu vào tập tin clients.txt. Thông tin của mỗi khách hàng được lưu trên từng dòng.

Mô tả định dạng của tập tin clients.txt cần tạo ra

5	Có 5 khách hàng trong tập tin này
100 - Đỗ Nguyên Kha - 15000000	Thông tin khách hàng thứ nhất
200 - Đặng Hoàng Hải - 75950000	
300 - Trương Phước Lộc - 90000000	
400 - Bùi Tấn Lộc - 78000000	
500 - Trần Minh Triết - 300000000	

Biết rằng thứ tự lưu trữ là:

- + Mã khách hàng: chuỗi có độ dài cố định 3 kí tự.
- + Tên khách hàng: chuỗi có độ dài tối đa 25 kí tự.
- + Số tiền đang có: số nguyên. Có thể âm, dương hoặc = 0

Chi tiết mã nguồn như sau

Mã nguồn hàm main

```
25  /// Nhập danh sách khách hàng
26  int count;
27  cout << "Ban muon nhap bao nhieu khach hang?";
28  cin >> count;
29
30  vector<Client*> list; // Chứa danh sách khách hàng đã nhập
31  int ID; // Mã số khách hàng
32  string name; // Họ tên khách hàng
33  double balance; // Số tiền trong tài khoản
34
35  for (int i = 0; i < count; i++)
36  {
37      cout << endl << "Nhap khach hang thu :" << i + 1 << endl;
38      cout << "Ma so khach hang:"; cin >> ID;
39      cout << "Ten:"; cin >> name;
40      cout << "So tien trong tai khoan:"; cin >> balance;
41
42      list.push_back(new Client(ID, name, balance));
43  }
44
45  /// Xuất thông tin danh sách khách hàng ra tập tin
46  ofstream writer("clients.txt", ios::out);
47  if (!writer) {
48      cerr << "Khong the mo file!";
49      exit(EXIT_FAILURE);
50  }
51
52  // Dòng đầu tiên lưu số lượng của danh sách
53  writer << list.size();|
54  // Mỗi dòng tiếp theo là thông tin của một khách hàng
55  for (auto& client : list)
56      writer << endl << client->ToString();
```

Lớp Client lúc này có thể cài đặt như sau:

```
1  #pragma once
2  #include <iostream>
3  #include <string>
4  #include <sstream>
5  #include <vector>
6  using namespace std;
7
8  class Client
9  {
10     int mAccountID;
11     string mName;
12     double mBalance;
13
14 public:
15     Client(int id, string name, double balance)
16     {
17         mAccountID = id;
18         mName = name;
19         mBalance = balance;
20     }
21
22     string ToString()
23     {
24         stringstream ss;
25         ss << mAccountID << " - " << mName << " - " << mBalance;
26
27         return ss.str();
28     }
29 };
```

Những điểm cần chú ý của mã nguồn trên:**1. Duyệt qua danh sách đối tượng**

Trong hàm main dòng 55 là kỹ thuật duyệt đối tượng có từ C++11. Sử dụng từ khóa auto để cho trình biên dịch tự động suy ra kiểu của đối tượng, sử dụng & để lấy tham chiếu đến nhằm tránh tạo bản sao khi duyệt qua đối tượng.

2. Chuyển đối tượng sang chuỗi

Sử dụng stringstream có điểm đặc biệt là giống với cout, dùng chung toán tử, nhưng viết biến ra một stream riêng, khi cần xuất ra tập tin thì chuyển sang chuỗi với hàm str() (mã nguồn lớp client dòng 27).

Tutorial 2: Làm quen với việc đọc tập tin text.

Yêu cầu: Viết chương trình đọc thông tin ngân hàng của các khách hàng trong tập tin và hiển thị ra màn hình.

Mô tả cách thức tổ chức của tập tin clients.txt cần đọc

5	Có 5 khách hàng trong tập tin này
100 - Đỗ Nguyên Kha - 15000000	Thông tin khách hàng thứ nhất
200 - Đặng Hoàng Hải - 75950000	Khách hàng này có số tiền âm!!!
300 - Trương Phước Lộc - 90000000	
400 - Bùi Tấn Lộc - 0.0	Khách hàng này ko có tiền
500 - Trần Minh Triết - 300000000	Khách hàng này có số

	tiền âm!!!

Mã nguồn chi tiết của chương trình như sau

+ Mã nguồn hàm main

```
15     ifstream reader("clients.txt", ios::in);
16     if (!reader) // Không thể mở file để đọc
17     {
18         cout << "Khong the mo file de doc!";
19         exit(EXIT_FAILURE);
20     }
21
22     string line;
23     getline(reader, line, '\n');
24
25     int count = stoi(line); // Số lượng khách hàng
26     vector<Client*> list; // Chứa danh sách khách hàng
27
28     // Đọc từng dòng và chuyển thành đối tượng
29     for (int i = 0; i < count; i++)
30     {
31         getline(reader, line, '\n');
32
33         Client* client = Client::Parse(line);
34         list.push_back(client);
35     }
36
37     // Hiển thị danh sách khách hàng
38     for (auto& client: list)
39     {
40         cout << endl << client->ToString();
41     }
```


+ Lớp Client cần bổ sung hàm **Parse** như sau:

```
static Client* Parse(string line)
{
    string SEPERATOR = " - ";
    vector<string> tokens;

    int startpos = 0; // Vị trí bắt đầu tìm
    size_t pos = line.find(SEPERATOR); // Thử tìm vị trí đầu tiên

    while (pos != string::npos) // Tìm thấy
    {
        tokens.push_back(line.substr(startpos, pos-startpos));

        // Cập nhật vị trí bắt đầu tìm lại
        startpos = pos + SEPERATOR.length();

        // Tiếp tục tìm
        pos = line.find(SEPERATOR, startpos);
    }

    // Phần còn sót lại
    if (startpos < line.length())
        tokens.push_back(line.substr(startpos, line.length() - startpos));

    int id = stoi(tokens[0]);
    string name = tokens[1];
    double balance = stod(tokens[2]);
    Client* p = new Client(id, name, balance);

    return p;
}
```

Một số vấn đề cần chú ý từ mã nguồn trên

Cách tách chuỗi

Sử dụng hàm find của std::string để tìm vị trí chuỗi con xuất hiện đầu tiên, sau đó tạo ra chuỗi con từ các thông tin có được và đưa vào một vector tokens.

Tiếp theo, cập nhật vị trí cần tìm ban đầu và lần lượt cho các chuỗi con tìm thấy vào vector tokens.

Cần chú ý là chuỗi con cuối cùng cần phải được xử lý riêng.

Chuyển chuỗi sang số

Hai hàm chuyển cơ bản là stoi – chuyển chuỗi sang số nguyên và stod – chuyển chuỗi sang số thực.

3 Bài tập cá nhân

Năm 2030, loài người đã có những bước tiến đáng kể trong việc chinh phục không gian. Nổi tiếp sau sự kiện đưa tàu thăm dò Curiosity lên sao Hỏa vào năm 2014, con người phái tiếp tàu mẹ Mothership tới nơi này với mục đích xây dựng căn cứ và tạo môi trường thích hợp cho loài người sinh sống ở đây vào năm 2040.

Hàng ngày, sử dụng công nghệ in 3D, tàu mẹ nhận lệnh từ căn cứ ở Trái Đất và tạo ra các robot phục vụ những công việc nhất định tại sao Hỏa. Có 3 loại robot được sử dụng chủ yếu.

- **Alpha:** Robot công phá, sử dụng năng lượng mặt trời để di chuyển, mỗi 1km sử dụng 100 W năng lượng, dùng năng lượng nhiệt hạch để phá đá. Để tạo ra cần 5000 J năng lượng, 4 kg vàng, 2 kg sắt, chế tạo trong 2h.
- **Beta:** Robot vận chuyển, sử dụng năng lượng mặt trời để di chuyển, mỗi 1km sử dụng 200 W năng lượng. Mỗi robot beta có thể vận chuyển tối đa 50kg đất đá. Để tạo ra cần 4000 J năng lượng, 6 kg vàng, không cần sắt, chế tạo trong 3h.
- **Omega:** Robot xây dựng, không có khả năng di chuyển, mỗi khi di chuyển phải nhờ tới robot Beta. Mỗi h làm việc tốn 300 W năng lượng. Để tạo ra cần 6000 J năng lượng, 10 kg sắt, không cần vàng, chế tạo trong 15h.

Cuối mỗi ngày, Mothership sẽ gửi thông tin tiêu tốn các tài nguyên của mình dùng trong chế tạo robot về căn cứ ở Trái Đất để Bộ chỉ huy quyết định phương án tiếp tế tài nguyên.

Sử dụng phương pháp lập trình hướng đối tượng, hãy viết chương trình trợ giúp tàu mẹ Mothership làm các việc như sau:

1. Đọc danh sách các robot cần phải chế tạo từ tập tin (xem bên dưới) và tạo ra các robot tương ứng.
2. Thống kê đã tạo ra tổng cộng bao nhiêu robot, và bao nhiêu con mỗi loại.
3. Thống kê tổng năng lượng đã tiêu thụ, tổng thời gian đã sử dụng để chế tạo toàn bộ các robot.
4. Ứng với mỗi loại tài nguyên vàng, sắt, thống kê tổng khối lượng đã sử dụng tương ứng.
5. Cho tất cả những robot Alpha và robot Beta di chuyển 10 km. Thống kê tổng năng lượng các robot đã tiêu thụ cho việc di chuyển.

Lưu ý:

- Vì lí do hồng hóc của bộ xử lí trung tâm trong quá trình hạ cánh, tàu mẹ Mothership không thể thực hiện được phép tính nhân. (➔ Chỉ sử dụng phép cộng và phép đóng gói đối tượng.)
- Tuy mỗi loại robot có cấu trúc dữ liệu và hành động khác nhau, nhưng tàu mẹ chỉ được quyền tạo ra một danh sách robot mà thôi.
- Để kiểm tra loại robot, C++ dùng toán tử `typeid` để kiểm tra kiểu (lớp), VD “`typeid(*robot)==typeid(Alpha)`”. Để sử dụng 1 robot với giao diện cụ thể, C++ dùng toán tử ép kiểu từ giao diện tổng quát, VD “`Alpha* a = dynamic_cast <Alpha*>(robot)`”

Tập tin input mẫu: Dòng đầu cho biết tổng số robot, các dòng còn lại lần lượt là loại robot phải tạo ra.

```
7
Beta
Omega
Alpha
Beta
Beta
Omega
Alpha
```