Bài tập

Nguyễn Khắc Huy

Bài tập

Bài tập 4.1:

Xây dựng lớp **Date** có những phương thức sau:

(Nhóm tạo hủy)

- ✓ Khởi tạo mặc định vào ngày hiện tại.
- ✓ Khởi tạo từ ngày, tháng, năm cho trước.
- ✓ Khởi tạo từ năm và ngày tuyệt đối trong năm.
- ✓ Khởi tạo từ một đối tượng Date khác.

(Nhóm truy xuất thông tin)

- ✓ Thông báo ngày, tháng, năm.
- ✓ Thông báo thứ trong tuần.
- ✓ Thông báo tuần trong năm.
- ✓ Thông báo ngày tuyệt đối trong năm.

Bài tập

Bài tập 4.1:

Xây dựng lớp **Date** (tiếp theo):

(Nhóm xử lý nghiệp vụ)

- ✓ Kiểm tra năm nhuận.
- ✓ So sánh thứ tự với một ngày khác.
- ✓ Tính khoảng cách đến một ngày khác (đơn vị ngày).
- ✓ Tìm ngày liền trước, liền sau.
- ✓ Cộng ngày, tháng, năm.

(Nhóm toán tử)

- ✓ Toán tử so sánh: >, <, ==, >=, <=, !=.
- ✓ Toán tử nhập xuất: >>, <<.

Bài tập

Bài tập 4.2:

Xây dựng lớp **Time** có những phương thức sau:

(Nhóm tạo hủy)

- ✓ Khởi tạo mặc định vào thời điểm hiện tại.
- ✓ Khởi tạo từ giờ, phút, giây cho trước.
- ✓ Khởi tạo từ giây tuyệt đối trong ngày.
- ✓ Khởi tạo từ một đối tượng Time khác.

(Nhóm truy xuất thông tin)

- ✓ Thông báo giờ, phút, giây.
- ✓ Thông báo giây tuyệt đối trong ngày.

Bài tập

Bài tập 4.2:

Xây dựng lớp **Time** (tiếp theo):

(Nhóm xử lý nghiệp vụ)

- ✓ So sánh thứ tự với một thời điểm khác.
- ✓ Tính khoảng cách đến thời điểm khác (đơn vị giây).
- ✓ Cộng giờ, phút, giây.

(Nhóm toán tử)

- ✓ Toán tử so sánh: >, <, ==, >=, <=, !=.
- ✓ Toán tử nhập xuất: >>, <<.

Một số vấn đề thảo luận

Nguyễn Khắc Huy

Type cast

- Bài tập 3.5 - IntArray
  - Ép kiểu về int *

Unary Operator - Postfix vs Prefix

```
void main() {
        int x, y;
        x = y = 5;
        x++;
        cout << x;
        ++y;
        cout << y;
}
```

```
void main() {
        int x, y, z;
        x = y = z = 5;
        z = y++;
        cout << z;
        z = ++x;
        cout << z;
}
```

```
void main() {
        PhanSo p1 (1,2), p2;
        p2 = p1++;
        p2 = ++p1;
}
```

Từ khóa friend

Template, Thư viện C++

Nguyễn Khắc Huy

BMCNPM – ĐHKHTN TPHCM
09/2014

Nội dung

- Function Template
- Class Template
- Thư viện C++
- Bài tập

Function Template

- Xét hàm tìm min giữa 2 số:

```
int timMin(int a, int b)
{
        return (a < b) ? a : b;
}

float timMin(float a, float b)
{

}

PhanSo timMin(PhanSo a, PhanSo b)
{
        return (a < b) ? a : b;
}
```

Dùng Function Template!!

Function Template

- Hàm tìm min dùng Function Template:

```
template <class T>
T timMin(T a, T b)
{
    return (a < b) ? a : b;
}
```

```
void main()
{
    int a = 5, b = 3;
    int c = timMin(a, b);

    float d = timMin(1.5, 2.3);

    PhanSo p1(1, 2);
    PhanSo p2(1, 3);
    PhanSo p3 = timMin(p1, p2);
}
```

Function Template

- Đặc điểm của Function Template:
  - ✓Hàm tổng quát cho nhiều kiểu dữ liệu khác nhau.
  - ✓Tham số hóa kiểu dữ liệu.
  - ✓Kiểu cụ thể được quyết định khi gọi hàm.
- Ghi chú:
  - ✓Từ khóa **“class”** có thể thay bằng **“typename”**.
  - ✓Phần khai báo và cài đặt đều có khai báo template.
  - ✓Phần cài đặt hàm phải nằm cùng file:
    - Phần khai báo hàm.
    - Phần gọi sử dụng hàm.

Nội dung

- Bài tập
- Function Template
- Class Template
- Thư viện C++

Class Template

- Xét lớp đối tượng Mang:

```
class MangNguyen
{ private:
        int         m_iKichThuoc;
        int         *m_pDuLieu;
public:
        Mang(int iKichThuoc);

};
cla
{ private:
        int         m_iKichThuoc;
        PhanSo              *m_pDuLieu;
public:
        Mang(int iKichThuoc);
        PhanSo LayPhanTu(int iViTri);
};
```

Class Template

- Lớp Mang dùng Class Template:

```
template <class T>
class Mang
{
private:
        int     m_iKichThuoc;
        T       *m_pDuLieu;
public:
        Mang(int iKichThuoc);
        T layPhanTu(int iViTri);
};
```

```
void main()
{
        Mang<int>       m1(10);
        int   a = m1.layPhanTu(5);

        Mang<PhanSo> m2(5);
        PhanSo   p = m2.layPhanTu(2);
}
```

Class Template

- Đặc điểm của Class Template:
  - ✓Lớp tổng quát cho nhiều kiểu dữ liệu khác nhau.
  - ✓Tham số hóa kiểu dữ liệu.
  - ✓Kiểu cụ thể được truyền vào khi tạo đối tượng.
- Ghi chú:
  - ✓Từ khóa **"class"** có thể thay bằng **"typename"**.
  - ✓Phần cài đặt lớp phải nằm cùng file:
    - Phần khai báo lớp.
    - Phần tạo và sử dụng đối tượng của lớp.

  ➔ **Viết cài đặt bên trong lớp khi dùng Template.**

Nội dung

- Bài tập
- Function Template
- Class Template
- Thư viện C++

Thư viện C++

- Khái niệm thư viện:
  - ✓Tập hợp những lớp, hàm có sẵn giúp giải quyết công việc thường gặp.
  - ✓Bộ công cụng hữu ích của lập trình viên.
  - ✓Một vài thư viện C++:
    - Thư viện chuẩn (C++ Standard Library).
    - Thư viện boost.
    - Thư viện MFC (**M**icrosoft **F**oundation **C**lasses).

Thư viện C++

- Thư viện chuẩn:
  - ✓Thư viện cơ bản nhất của C++.
  - ✓Các lớp và hàm nằm trong namespace std.
  - ✓File Header không .h.
  - ✓Phân nhóm:
    - Nhóm nhập xuất: iostream, iomanip, fstream, …
    - **Nhóm STL.**
    - …
    - Thư viện chuẩn C: file header **c**xxx.

Thư viện C++

- Thư viện STL (**S**tandard **T**emplate **L**ibrary):
  - ✓Một phần của thư viện chuẩn.
  - ✓Các lớp và hàm hỗ trợ lập trình với template.
  - ✓Phân nhóm:
    - Nhóm container: vector, list, deque, set, …
    - Nhóm string: string, …
    - Nhóm iterator.
    - …

Thư viện C++

- Lớp string:
  - ✓File header <string>.
  - ✓Lớp đại diện cho các đối tượng chuỗi.
  - ✓**Giải quyết 3 vấn đề con trỏ.**
  - ✓Các phương thức chính:
    - string(char *): khởi tạo từ một chuỗi ký tự.
    - length(): lấy chiều dài chuỗi.
    - Toán tử [ ]: lấy ký tự tại một vị trí nào đó.
    - Toán tử >, <, ==, >=, <=, !=: so sánh theo thứ tự từ điển.
    - Toán tử +, +=: nối chuỗi.
    - find(char *): tìm chuỗi con.
    - substr(int, int): lấy chuỗi con.

Thư viện C++

- Ví dụ lớp string:

```
void main()
{
        string    s1(“software”);
        string    s2(“SoftWare”);

        if (s1 == s2)
                cout << “equal.” << endl;
        else
                cout << “not equal.” << endl;

        s2 = s1.substr(4, 4);
        cout << s2;

        string    s3 = s1 + s2;
        cout << s3 << endl;
}
```

Lớp vector

- Lớp vector:
  - ✓File header <vector>.
  - ✓Lớp mảng kiểu T.
  - ✓**Giải quyết các vấn đề con trỏ.**
  - ✓Các phương thức chính:
    - vector<T>(): khởi tạo mảng kiểu T.
    - size(): lấy kích thước mảng.
    - push_back(T): thêm phần tử vào cuối mảng.
    - Toán tử [ ]: lấy phần tử tại một vị trí nào đó.

Thư viện C++

- Ví dụ:

```
void main()
{
        vector<int>    v1;
        v1.push_back(1);
        v1.push_back(2);

        for (int i = 0; i < v1.size(); i++)
                cout << v1[i] << " ";

        vector<PhanSo *>   v2;
        v2.push_back(new PhanSo(2, 6));
        v2[0]->rutGon();
}
```

Tóm tắt

- Template:
  - ✓Cách thức tham số hóa kiểu dữ liệu.
  - ✓Cho phép lập trình trên kiểu dữ liệu tổng quát.
  - ✓Function template:
    - Hàm tổng quát cho nhiều kiểu dữ liệu khác nhau.
    - Kiểu cụ thể được quyết định khi gọi hàm.
  - ✓Class template:
    - Lớp tổng quát cho nhiều kiểu dữ liệu khác nhau.
    - Kiểu cụ thể được truyền vào khi tạo đối tượng từ lớp.
  - ✓Phần cài đặt nằm cùng file:
    - Phần khai báo.
    - Phần sử dụng.

Tóm tắt

- Thư viện C++:
  - ✓Bộ công cụ hỗ trợ lập trình.
  - ✓Thư viện chuẩn:
    - Thư viện STL.
  - ✓Thư viện boost.
  - ✓Thư viện MFC.

Bài tập

Bài tập 4.3:

Sử dụng template, chỉnh sửa lại lớp **mảng** cho phép lưu trữ và thao tác trên kiểu dữ liệu bất kỳ.

(Gợi ý)

- ✓ Dùng class template khai báo lớp mảng.
- ✓ Dữ liệu mảng kiểu T.
- ✓ Các phương thức thao tác dữ liệu mảng dùng function template.

Lời cảm ơn

- Nội dung được xây dựng dựa trên slide trình bày của Thầy Đinh Bá Tiến, Thầy Nguyễn Minh Huy.