

Chương 2

Lớp và đối tượng (phần 2)

Nội dung

- 2.6 Đối tượng làm tham số của hàm
- 2.7 Hàm trả về đối tượng
- 2.8 Kết hợp đối tượng
- 2.9 Nạp chồng hàm thành phần (function overloading)
- 2.10 Hàm toán tử (operator function)

•B/m KTHT, khoa CNTT, ĐHXD

•Bài giảng C++. 2/5/2022

2.6) Đối tượng là tham số của hàm

- Có thể lấy một đối tượng làm tham số truyền cho hàm
- Khi truyền theo tham trị, hàm sẽ
- Khi truyền theo tham chiếu, hàm có thể gọi đến

•Bài giảng C++. 2/5/2022

Chú ý khi truyền tham số là đối tượng

- Sử dụng truyền tham trị có thể mất thêm thao tác xử lý (để tạo bản sao) và bộ nhớ (để lưu trữ bản sao)
- Sử dụng truyền tham chiếu có thể hiệu quả hơn và cho phép hàm có thể
- Tuy nhiên phải cẩn thận để

•7-4

Tiếp

- Để tiết kiệm thao tác và bộ nhớ mà vẫn tránh sửa đổi nhầm dữ liệu của tham số thì cần
- Nếu hàm printMatrix gọi đến hàm thành phần của lớp Matrix, thì hàm thành phần đó cần

•Bài giảng C++. 2/5/2022

Chương trình minh họa matrix4.cpp

- Viết 1 hàm để nhập dữ liệu phần tử cho 1 đối tượng Matrix (từ bàn phím)
void inputMatrix(Matrix& mt)
- Viết 1 hàm để in dữ liệu phần tử cho 1 đối tượng Matrix
void printMatrix(const Matrix& mt)

•Bài giảng C++. 2/5/2022

•Bài giảng C++. 2/5/2022

•Bài giảng C++. 2/5/2022

2.7) Hàm trả về kết quả là đối tượng

- Một hàm có thể trả về đối tượng
- Ví dụ hàm thực hiện việc nhập dữ liệu cho 1 đối tượng Matrix,

•Bài giảng C++. 2/5/2022

Ví dụ hàm khởi tạo 1 ma trận mới và trả về đối tượng Matrix

```
Matrix initMatrix()
{
    int r,c;
    cin >> r;    // nhập hàng và cột
    cin >> c;

}
```

•Bài giảng C++. 2/5/2022

- Ví dụ khác là hàm thực hiện việc nhân 1 ma trận với 1 số nguyên,

```
Matrix mulMatrix(const Matrix& mt, int a)
{
    int r,c;
    r = mt.getRow();
    c = mt.getCol();
    Matrix m(r,c); // tạo đối tượng cục bộ
    ... // tính m(i,j) = a * mt(i,j)
    return m; // trả về đối tượng kết quả
}
```

•Bài giảng C++. 2/5/2022

Chương trình minh họa matrix5.cpp

- Viết 1 hàm trả về một đối tượng Matrix mà dữ liệu của nó được nhập từ bàn phím
Matrix initMatrix()
- Hãy viết 1 hàm trả về một đối tượng Matrix là kết quả của phép nhân giữa 1 ma trận (tham số mt) với 1 số nguyên (tham số a)
Matrix mulMatrix(const Matrix& mt, int a)

•Bài giảng C++. 2/5/2022

2.8) Kết hợp đối tượng

- Trong 1 chương trình, có thể phải kết hợp
- Có nhiều cách kết hợp tùy theo

Các trường hợp kết hợp đối tượng

- Dữ liệu thành phần của 1 đối tượng là
- Hàm thành phần của 1 đối tượng sử dụng
- Trong hàm thành phần của 1 đối tượng, ta gọi
- Hàm thành phần của 1 đối tượng

Một ví dụ về kết hợp đối tượng

- Trong chương trình có class Matrix biểu diễn ma trận và
- Ta có thể tạo hàm thành phần của lớp Matrix để trả về

```
Vecto Matrix::getRowVecto(int i)
```

•Bài giảng C++. 2/5/2022

Chương trình minh họa matrix6.cpp

- Khai báo và định nghĩa các hàm thành phần của **lớp Vecto**
- Viết hàm thành phần của lớp Matrix để trả về vecto hàng i của ma trận

```
Vecto Matrix::getRowVecto(int i)
```

- Viết hàm thành phần của lớp Matrix để trả về vecto cột j của ma trận

```
Vecto Matrix::getColVecto(int j)
```

•Bài giảng C++. 2/5/2022

•Bài giảng C++. 2/5/2022

•Bài giảng C++. 2/5/2022

2.9) Nạp chồng hàm (function overloading)

- Cho phép nhiều hàm trong một lớp có thể
- Khi gọi hàm thành phần trùng tên từ đối tượng,

•Bài giảng C++. 2/5/2022

Ví dụ

```
class Matrix{
    int row,col;
    int a[100][100];
public:
    ...
    //hàm lấy phần tử ma trận
    Matrix add(Matrix mt); //a[i][j] + mt.a[i][j]
    Matrix add(int k); //a[i][i] + k
};
```

```
int main(){
    Matrix m1(2,2), m2(2,2);
    inputMatrix(m1);
    inputMatrix(m2);

    return 0;
}
```

2.10 Hàm toán tử (operator function)

- Hàm toán tử là **hàm thành phần**
- Cú pháp khai báo

```
Matrix operator+(int k); //hàm toán tử +
```

•Bài giảng C++. 2/5/2022

Ví dụ khai báo hàm toán tử

```
class Matrix{
    int row,col;
    int a[100][100];
public:
    ...
    // công ma trận với 1 số
    Matrix operator+(int k);
    // nhân ma trận với 1 số

    // nhân ma trận với ma trận

    ...
};
```

Tác dụng của hàm toán tử

- Hàm toán tử có tác dụng
- **Lời gọi hàm toán tử** sẽ sử dụng

```
Matrix m1(4);  
inputMatrix(m1);
```

•Bài giảng C++. 2/5/2022

Ví dụ lời gọi hàm toán tử

```
class Matrix{  
    int row,col;  
    int a[100][100];  
public:  
    ...  
    // cộng ma trận với 1 số  
    Matrix operator+(int k);  
    // nhân ma trận với 1 số  
    Matrix operator*(int k);  
  
    // nhân ma trận với ma trận  
    Matrix operator*(Matrix& mt);  
    ...  
};
```

```
int main(){  
    Matrix m1(4);  
    inputMatrix(m1);  
    // gọi hàm toán tử  
  
    printMatrix(m1);  
    return 0;  
}
```

Nhận xét

- Cú pháp gọi hàm toán tử giống với
- Toán hạng bên trái phép toán là

```
Matrix operator+(int k); //hàm toán tử +
```

- Hàm trả về 1 đối tượng Matrix, do đó có thể

•Bài giảng C++. 2/5/2022

Tiếp

- Phép toán + là phép toán 2 ngôi, do đó hàm toán tử + cần
- Nếu hàm toán tử lấy theo phép toán 1 ngôi,

VD: hàm toán tử - một ngôi (phép toán đảo dấu)

```
Matrix m1(4);  
inputMatrix(m1);
```

•Bài giảng C++. 2/5/2022

Ví dụ hàm toán tử 1 ngôi và kết hợp nhiều hàm trong 1 biểu thức

```
class Matrix{
    int row,col;
    int a[100][100];
public:
    ...
    // lấy đảo dấu ma trận
    Matrix operator-();

    // nhân ma trận với 1 số
    Matrix operator*(int k);

    // tăng ma trận lên 1
    Matrix operator++();
};
```

```
int main(){
    Matrix m1(4);
    inputMatrix(m1);
    // gọi hàm toán tử

    printMatrix(m4);
    return 0;
}
```

Định nghĩa hàm toán tử

- Cách viết phần lệnh của hàm toán tử giống
- Hàm toán tử giống như 1 phép toán trong biểu thức nên
- Việc định nghĩa hàm toán tử hoàn toàn do lập trình viên

•Bài giảng C++. 2/5/2022

Ví dụ định nghĩa hàm toán tử 2 ngôi

```
// cộng ma trận với 1 số
Matrix Matrix::operator+(int k)
{
    Matrix t(row,col);
    for(int i=0;i<row;i++)
        for(int j=0;j<col;j++)
        {
            t.a[i][j] = a[i][j]+k;
        }
    return t;
}
```

```
// nhân ma trận với 1 số
Matrix Matrix::operator*(int k)
{
    Matrix t(row,col);
    for(int i=0;i<row;i++)
        for(int j=0;j<col;j++)
        {
            t.a[i][j] = a[i][j]*k;
        }
    return t;
}
```

Ví dụ định nghĩa hàm toán tử 1 ngôi

```
// đảo dấu ma trận
Matrix Matrix::operator-()
{
    Matrix t(row,col);
    for(int i=0;i<row;i++)
        for(int j=0;j<col;j++)
        {
            t.a[i][j] = -a[i][j];
        }
    return t;
}
```

```
// tăng ma trận thêm 1 dạng prefix
Matrix Matrix::operator++()
{
    Matrix t(row,col);
    // tăng trước
    for(int i=0;i<row;i++)
        for(int j=0;j<col;j++)
        {
            t.a[i][j] = t.a[i][j]+1;
        }
    return t;
}
```

t là biến Matrix cục bộ của hàm để lưu giá trị trả về

Chương trình minh họa matrix7.cpp

- Viết hàm toán tử 1 ngôi **operator~** để tính ma trận chuyển vị

Matrix operator~();

- Viết hàm toán tử 2 ngôi **operator[]** để trả lại Vecto hàng i của ma trận

Vecto operator[](int i);

•Bài giảng C++. 2/5/2022

- Viết hàm toán tử (2 ngôi) **+=** để mở rộng ma trận bằng cách thêm 1 hàng lấy từ 1 Vecto (thêm vào sau hàng cuối cùng)

Matrix operator+=(Vecto& v)

$$\begin{array}{ccc} 3 & 0 & 1 \\ 0 & 2 & 0 \end{array} += \begin{array}{ccc} 1 & 1 & 0 \end{array} \rightarrow \begin{array}{ccc} 3 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 1 & 0 \end{array}$$

•Bài giảng C++. 2/5/2022

Bài tập minh họa

- Nối tiếp class DoiTac đã có trong bài tập trước
- Bổ sung dữ liệu thành phần sau
 - Số loại mặt hàng đã mua (n mặt hàng)
 - Số lượng từng mặt hàng đã mua (n giá trị)
- Viết các hàm sau
 - void inputDoiTac(DoiTac& dt) để nhập liệu các dữ liệu thành phần
 - DoiTac initDoiTac() để tạo ra 1 đối tượng và nhập liệu

•Bài giảng C++. 2/5/2022

- Viết hàm toán tử lấy số lượng của mặt hàng i
`int& DoiTac::operator[](int i);`
- Viết hàm thành phần tính và trả về tổng số lượng hàng mà đối tác đã mua
`int DoiTac::tongSoluongHang();`
- void printDoiTac(const DoiTac& dt) để in ra màn hình các dữ liệu thành phần trên 1 dòng
Họ tên, Công ty, Chức vụ, Di động, Tổng hàng\n

•Bài giảng C++. 3/5/2022

- Viết hàm main()
 - Tạo 2 đối tượng DoiTac bằng hàm initDoiTac()
 - Thay đổi số lượng hàng đã mua của 2 đối tượng
 - In thông tin 2 đối tượng ra màn hình