

# Ứng dụng MFC (Visual C++) trong mô phỏng Robot và hệ Cơ điện tử

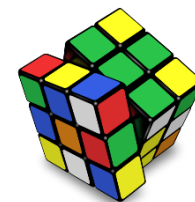


## Bài 2: Lập trình hướng đối tượng và thực hành với MFC

PHẠM MINH QUÂN

[mquan.ph@gmail.com](mailto:mquan.ph@gmail.com)

# Nội dung



## 1. Nhắc lại về lập trình hướng đối tượng

*1.1. Khái niệm về lớp, thuộc tính, phương thức, kế thừa...*

*1.2. Các lớp cơ bản được tạo sẵn khi lập trình MFC*

## 2. Thực hành: Khai báo và sử dụng lớp trong VC++

*2.1. Khai báo và định nghĩa lớp mới*

*2.2. Sử dụng các đối tượng của lớp trong ứng dụng MFC*

## 3. Mở rộng

*3.1. Giới thiệu cấu trúc các mô-đun, lớp của phần mềm e-Robot*

*3.2. Giới thiệu chi tiết về nhóm các mô-đun hỗ trợ bao gồm: các structure, các hàm toàn cục, mô-đun đại số tuyến tính.*

*3.3. Giới thiệu các nhiệm vụ cần thực hiện của một đề tài thiết kế, chế tạo robot*

# 1. Nhắc lại về lập trình hướng đối tượng

## 1.1. Các khái niệm cơ bản:

- Lớp (Class)
- Thuộc tính (Attribute)
- Phương thức (Method)
- Đối tượng (Object)

Ví dụ: File “...Dlg.h”

```
class CMFCApplication2Dlg : public CDialogEx
{
// Construction
public:
    CMFCApplication2Dlg(CWnd* pParent = NULL); // standard constructor

// Dialog Data
    enum { IDD = IDD_MFCAPPLICATION2_DIALOG };

protected:
    virtual void DoDataExchange(CDataExchange* pDX); // DDX/DDV support

// Implementation
protected:
    HICON m_hIcon;

// Generated message map functions
    virtual BOOL OnInitDialog();
    afx_msg void OnSysCommand(UINT nID, LPARAM lParam);
    afx_msg void OnPaint();
    afx_msg HCURSOR OnQueryDragIcon();
    DECLARE_MESSAGE_MAP()
};
```

Annotations:

- //Tên lớp (points to `CMFCApplication2Dlg`)
- //Phương thức (hàm) khởi tạo (points to `CMFCApplication2Dlg(CWnd* pParent = NULL);`)
- //Thuộc tính (biến thành phần của lớp) (points to `HICON m_hIcon;`)
- //Các phương thức (hàm thành phần của lớp) (points to `DECLARE_MESSAGE_MAP()`)

❖ Các phương thức đặc biệt:

- Hàm khởi tạo (constructor): gọi tới khi đối tượng của lớp được khởi tạo
- Hàm hủy (destructor): gọi tới khi đối tượng của lớp bị xóa (thường làm nhiệm vụ xóa bộ nhớ)

# 1. Nhắc lại về lập trình hướng đối tượng

## 1.2. Các lớp cơ bản được tạo sẵn trong ứng dụng MFC dialog-based

### ❑ Lớp ...App: quản lý khung ứng dụng

```
class CMFCApplication2App : public CWinApp
{
public:
    CMFCApplication2App();

    // Overrides
public:
    virtual BOOL InitInstance();
    // Implementation

    DECLARE_MESSAGE_MAP()
};
```

```
extern CMFCApplication2App theApp;
```

### ❑ Lớp CAboutDlg: quản lý hộp thoại About

```
class CAboutDlg : public CDialogEx
{
public:
    CAboutDlg();

    // Dialog Data
    enum { IDD = IDD_ABOUTBOX };

    protected:
    virtual void DoDataExchange(CDataExchange* pDX);

    // Implementation
protected:
    DECLARE_MESSAGE_MAP()
};
```

### ❑ Lớp ...Dlg: quản lý hộp thoại (dialog) chính

```
class CMFCApplication2Dlg : public CDialogEx
{
    // Construction
public:
    CMFCApplication2Dlg(CWnd* pParent = NULL); // standard constructor

    // Dialog Data
    enum { IDD = IDD_MFCAPPLICATION2_DIALOG };

    protected:
    virtual void DoDataExchange(CDataExchange* pDX); // DDX/DDV support

    // Implementation
protected:
    HICON m_hIcon;

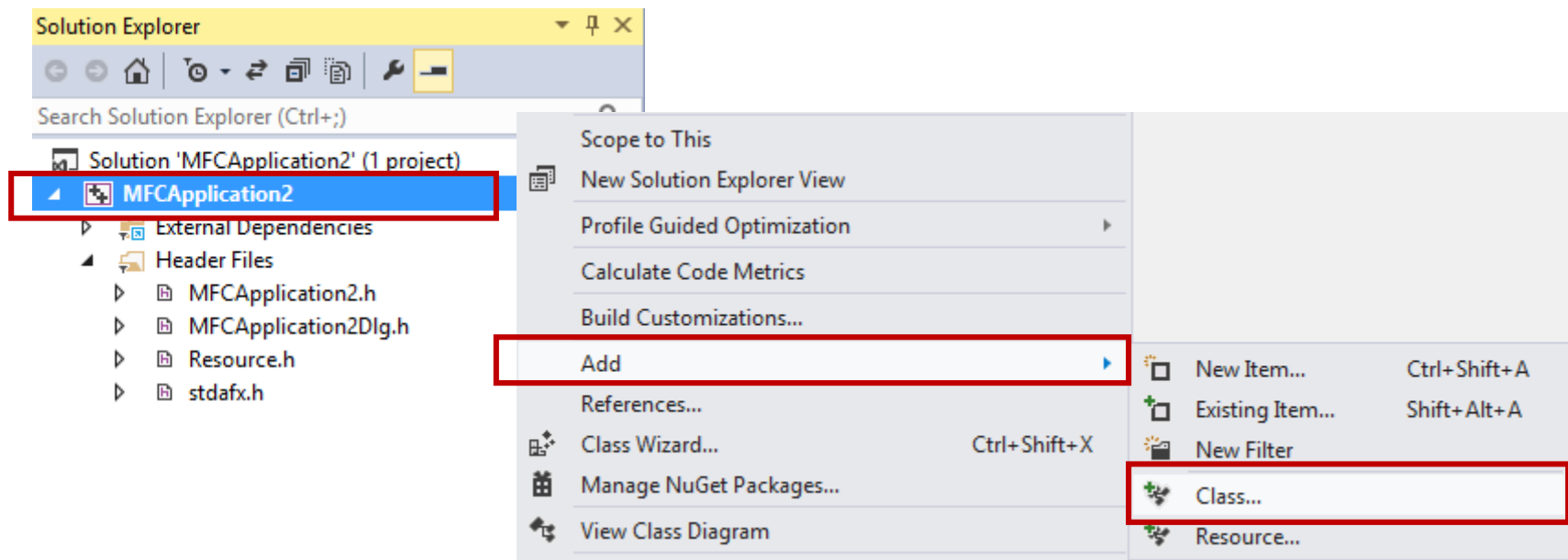
    // Generated message map functions
    virtual BOOL OnInitDialog();
    afx_msg void OnSysCommand(UINT nID, LPARAM lParam);
    afx_msg void OnPaint();
    afx_msg HCURSOR OnQueryDragIcon();
    DECLARE_MESSAGE_MAP()
};
```

## 2. Thực hành: Khai báo và sử dụng lớp (class) trong VC++

### 2.1. Khai báo và định nghĩa lớp mới

#### ❑ Tạo một lớp mới

- Click chuột phải vào tên project trong cửa sổ Solution Explorer. Chọn Add -> Class...

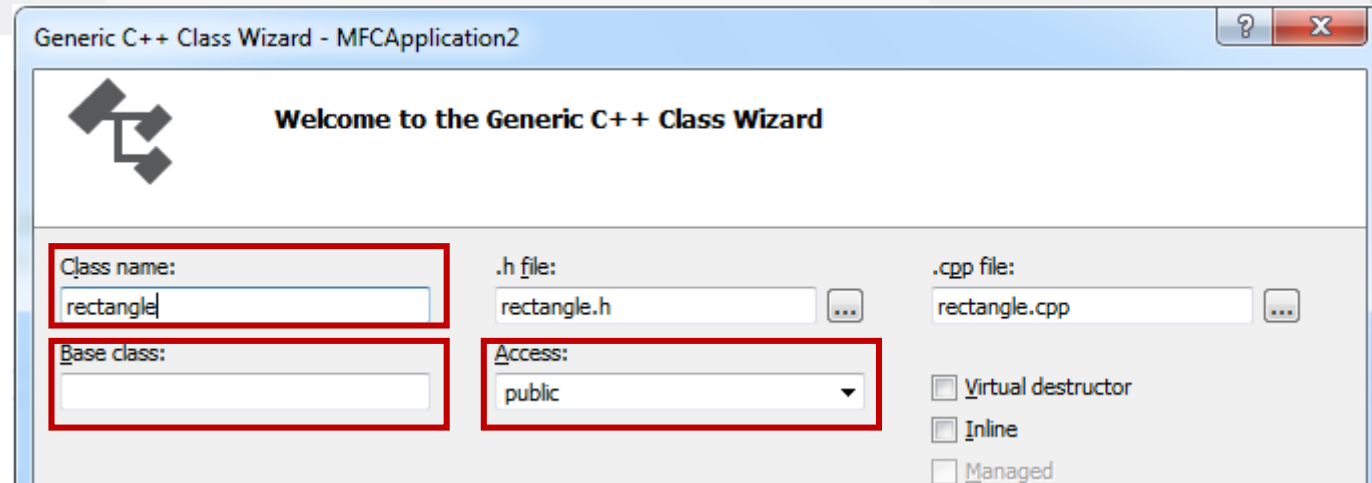
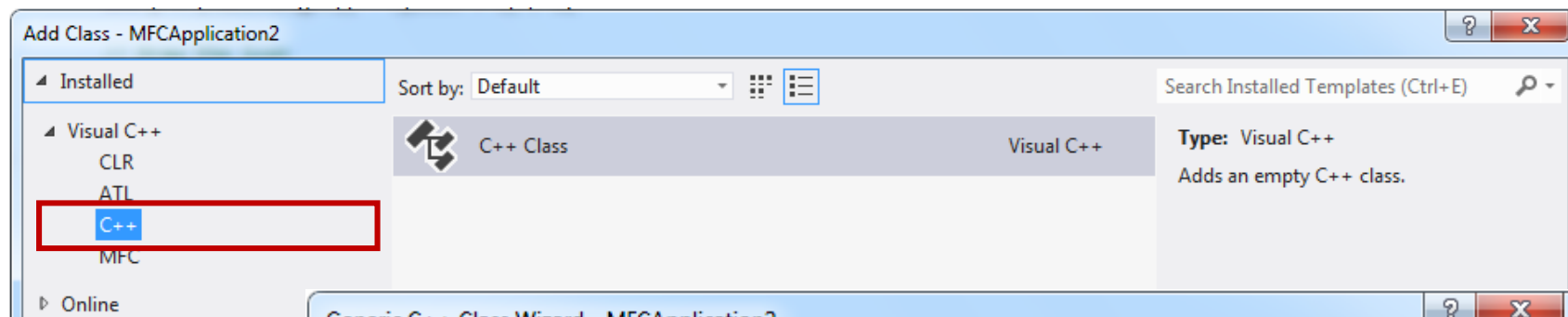


## 2. Thực hành: Khai báo và sử dụng lớp (class) trong VC++

### 2.1. Khai báo và định nghĩa lớp mới

❑ Tạo một lớp mới

➤ Chọn kiểu lớp là C++ Class và điền các thông tin của lớp cần tạo



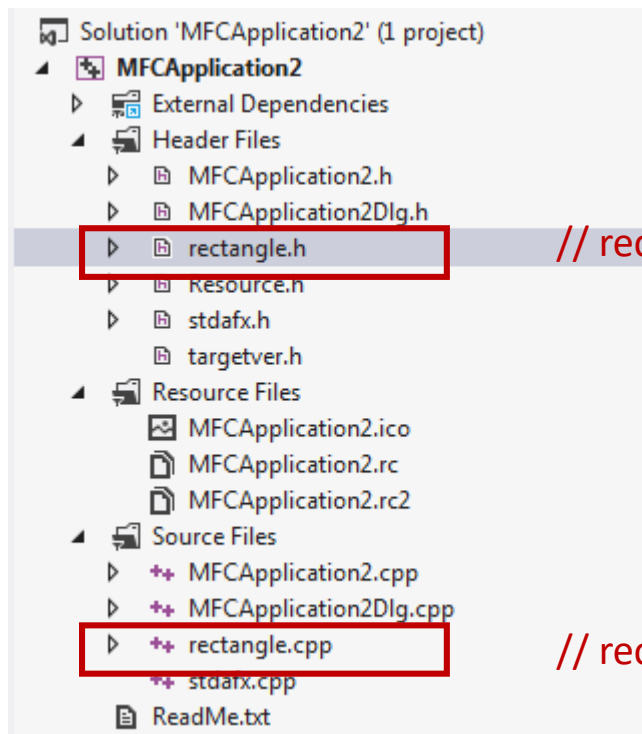
// Đặt tên lớp là rectangle  
(quản lý đối tượng là hình chữ nhật)

// Không có base class

## 2. Thực hành: Khai báo và sử dụng lớp (class) trong VC++

### 2.1. Khai báo và định nghĩa lớp mới

❑ Chương trình tự động tạo 2 file mới:



// rectangle.h : là file tiêu đề, dùng để khai báo lớp (declaration)

// rectangle.cpp : là file nội dung, dùng để định nghĩa lớp (definition)

## 2. Thực hành: Khai báo và sử dụng lớp (class) trong VC++

### 2.1. Khai báo và định nghĩa lớp mới

❑ Khai báo và định nghĩa lớp rectangle như sau:

File rectangle.h

```
class rectangle
{
public:
    rectangle(void);
    ~rectangle(void);

protected:
    double length;
    double width;

public:
    void SetSize(double a1, double a2);
    double GetLength();
    double GetWidth();
    double CalculateArea();
    double CalculatePerimeter();
};
```

File rectangle.cpp

```
#include "stdafx.h"
#include "rectangle.h"

rectangle::rectangle(void)
{
    length = 0;
    width = 0;
}

rectangle::~rectangle(void)
{
}

void rectangle::SetSize(double a1, double a2)
{
    if (a1>=a2) {length=a1; width=a2;}
    else {length=a2; width=a1;}
}

double rectangle::GetLength() { return length; }

double rectangle::GetWidth() { return width; }

double rectangle::CalculateArea() { return length*width; }

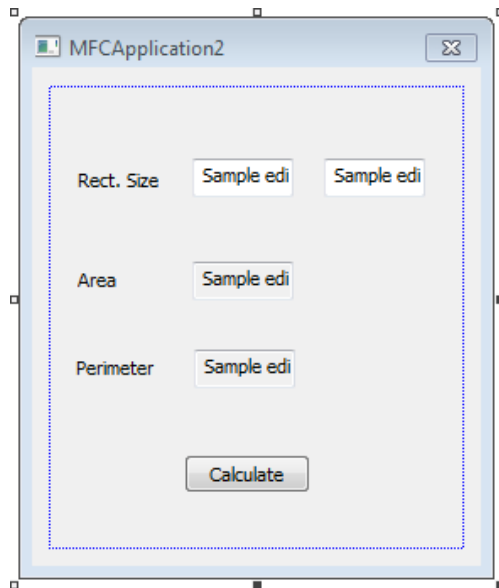
double rectangle::CalculatePerimeter() { return 2*(length+width); }
```



## 2. Thực hành: Khai báo và sử dụng lớp (class) trong VC++

### 2.2. Sử dụng các đối tượng của lớp mới trong ứng dụng MFC

❑ Thêm các control vào dialog và thêm các biến điều khiển control như sau:



File ...Dlg.h

```
public:
    double edit_a1;
    double edit_a2;
    double edit_area;
    double edit_peri;

    afx_msg void OnBnClickedButtonCalc();
```

File ...Dlg.cpp

```
CMFCApplication2Dlg::CMFCApplication2Dlg(CWnd* pParent /*=NULL*/)
    : CDialogEx(CMFCApplication2Dlg::IDD, pParent)
    , edit_a1(0)
    , edit_a2(0)
    , edit_area(0)
    , edit_peri(0)
{
    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
}

void CMFCApplication2Dlg::DoDataExchange(CDataExchange* pDX)
{
    CDialogEx::DoDataExchange(pDX);
    DDX_Text(pDX, IDC_EDIT_A1, edit_a1);
    DDX_Text(pDX, IDC_EDIT_A2, edit_a2);
    DDX_Text(pDX, IDC_EDIT_AREA, edit_area);
    DDX_Text(pDX, IDC_EDIT_PERI, edit_peri);
}

BEGIN_MESSAGE_MAP(CMFCApplication2Dlg, CDialogEx)
    ON_WM_SYSCOMMAND()
    ON_WM_PAINT()
    ON_WM_QUERYDRAGICON()
    ON_BN_CLICKED(IDC_BUTTON_CALC, &CMFCApplication2Dlg::OnBnClickedButtonCalc)
END_MESSAGE_MAP()
```

## 2. Thực hành: Khai báo và sử dụng lớp (class) trong VC++

### 2.2. Sử dụng các đối tượng của lớp mới trong ứng dụng MFC

#### ❑ Khai báo và sử dụng đối tượng của lớp mới

- Khai báo đối tượng của lớp mới trong file ...Dlg.h

```
// MFCApplication2Dlg.h : header file
//
#include "rectangle.h"

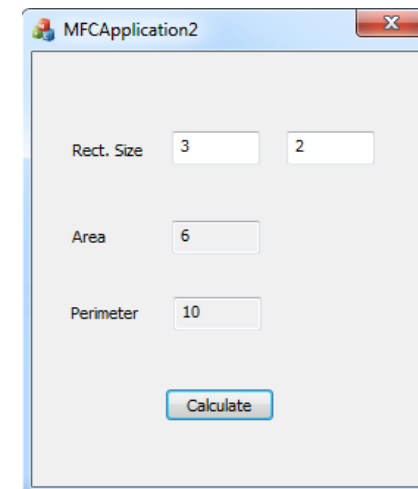
...

rectangle rect1;
```

- Chỉnh sửa hàm OnBnClicked...() trong file ...Dlg.cpp

```
void CMFCApplication2Dlg::OnBnClickedButtonCalc()
{
    // TODO: Add your control notification handler code here
    UpdateData(TRUE);
    rect1.SetSize(edit_a1, edit_a2);
    edit_area = rect1.CalculateArea();
    edit_peri = rect1.CalculatePerimeter();
    UpdateData(FALSE);
}
```

➡ Kết quả chạy:



## 3. Mở rộng

---

*3.1. Giới thiệu cấu trúc các mô-đun, các lớp của phần mềm e-Robot*

*3.2. Giới thiệu chi tiết về nhóm các mô-đun hỗ trợ bao gồm: các structure, các hàm toàn cục, mô-đun đại số tuyến tính.*

*3.3. Giới thiệu các nhiệm vụ cần thực hiện của một đề tài thiết kế, chế tạo robot*

(Xem các tài liệu kèm theo)

# Hết Bài 2

---

