GIÁO TRÌNH THỰC HÀNH
PHƯƠNG PHÁP LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Giáo viên: Nguyễn Bá Phúc
Phạm Thị Hồng
Nguyễn Vũ Dzũng

Mục lục

BÀI 1: LẬP TRÌNH TRÊN MÔI TRƯỜNG VISUAL STUDIO

1.1. Giới thiệu

- Visual studio là công cụ hỗ trợ mạnh trong việc xây dựng các ứng dụng Windows, các ứng dụng Web, điện thoại di động, ...
- Đặc điểm:
    + Phổ biến, thông dụng
    + Có tính chuyên nghiệp cao

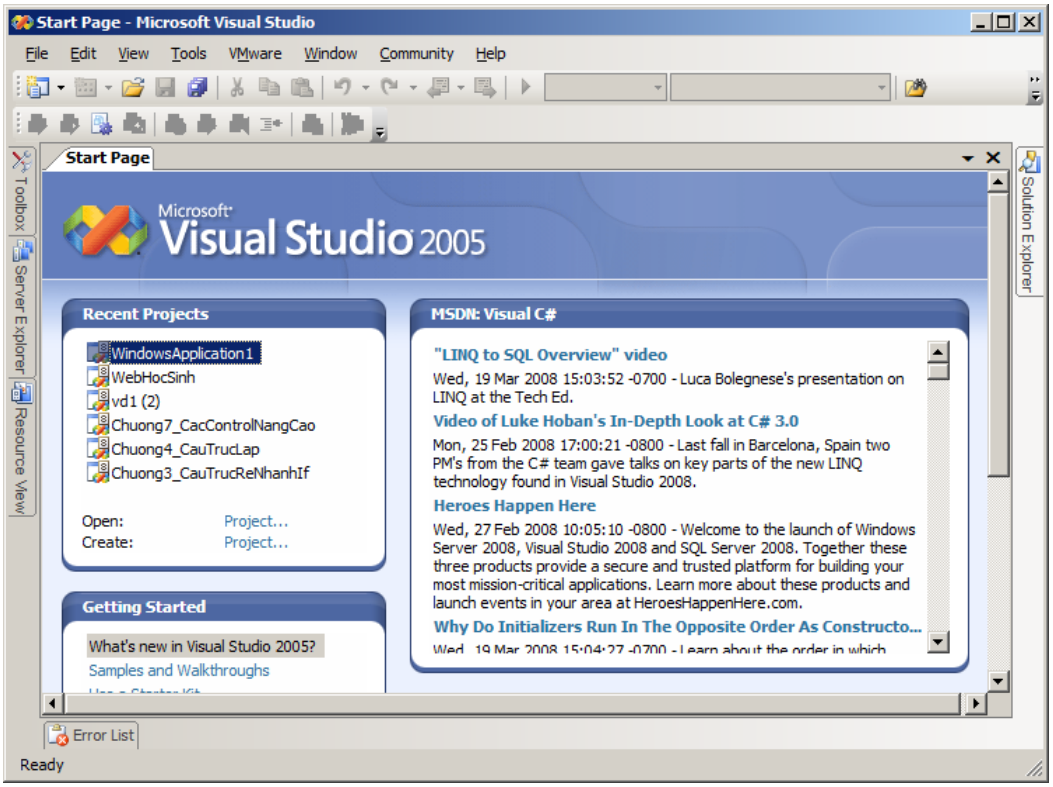
1.2. Làm quen với Visual Studio 2005

Trong giới hạn môn học, chúng ta chỉ làm việc trên môi trường Console của Visual C++ thuộc bộ công cụ Visual Studio 2005.

- Solution Explorer cho biết tất cả các project đang được mở và các tập tin trong Project. Tự Solution Explorer ta có thể dễ dàng truy xuất đến bất cứ thành phần trong Project.
- Trong Solution Explorer chỉ chứa duy nhất 1 Solution. Solution là một tập đầy đủ chứa các tập tin mã nguồn (code files) và các tài nguyên (resources), một solution có thể chứa nhiều project. Mỗi project chứa các tập tin mã nguồn và các tài nguyên của một ứng dụng cụ thể liên quan.

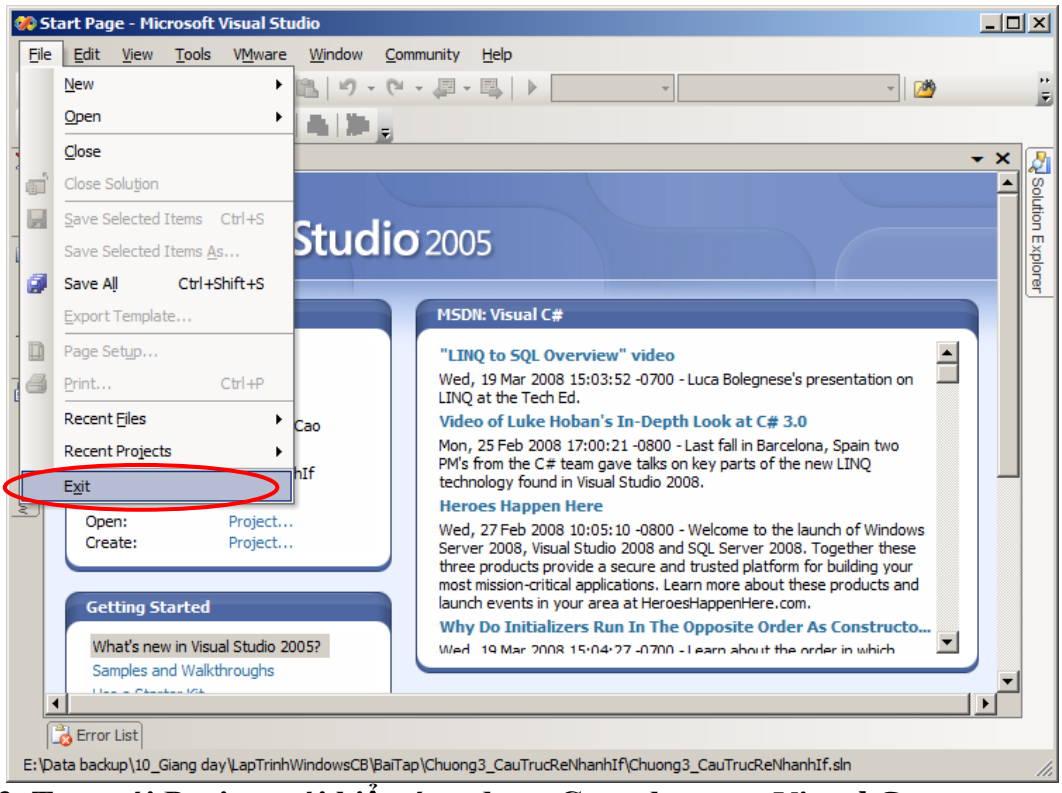
1.2.1. Khởi động Visual Studio 2005

- Vào menu **Start -> Programs -> Microsoft Visual Studio 2005 -> Microsoft Visual Studio 2005**
- Giao diện Visual Studio 2005 có dạng như sau:



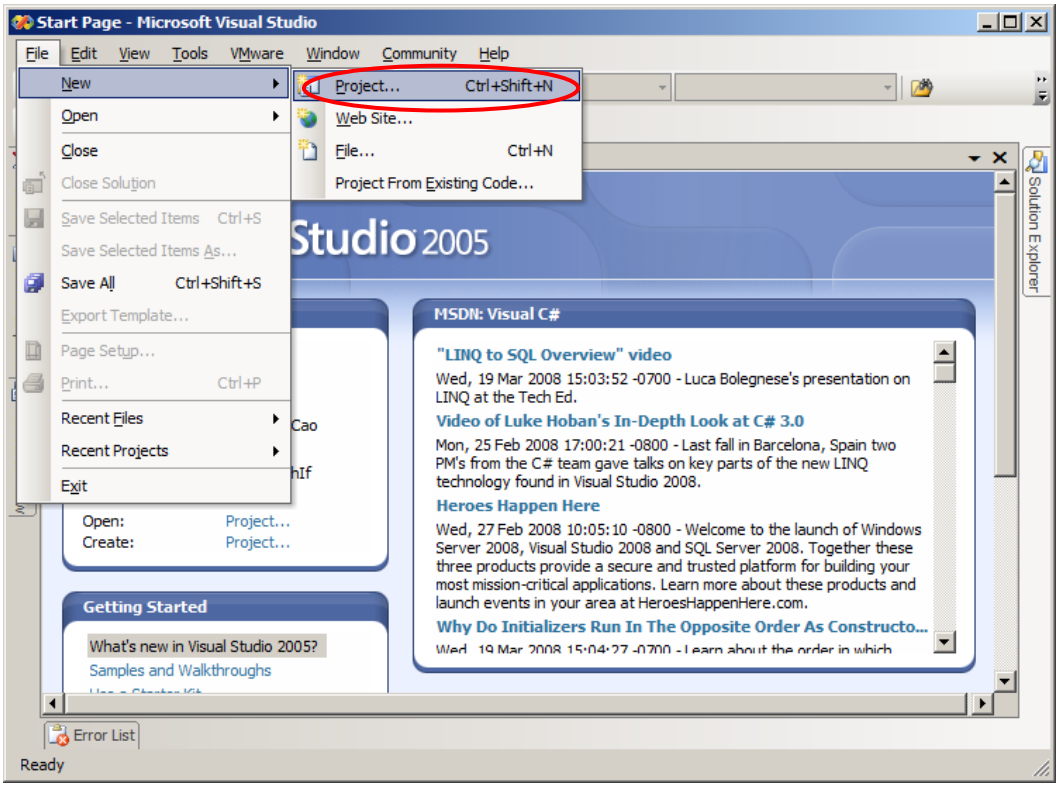
1.2.2. Thoát khỏi Visual Studio 2005

Trên menu bar, Click chuột chọn **File -> Exit**. Hoặc nhấn tổ hợp phím **Alt+F4**

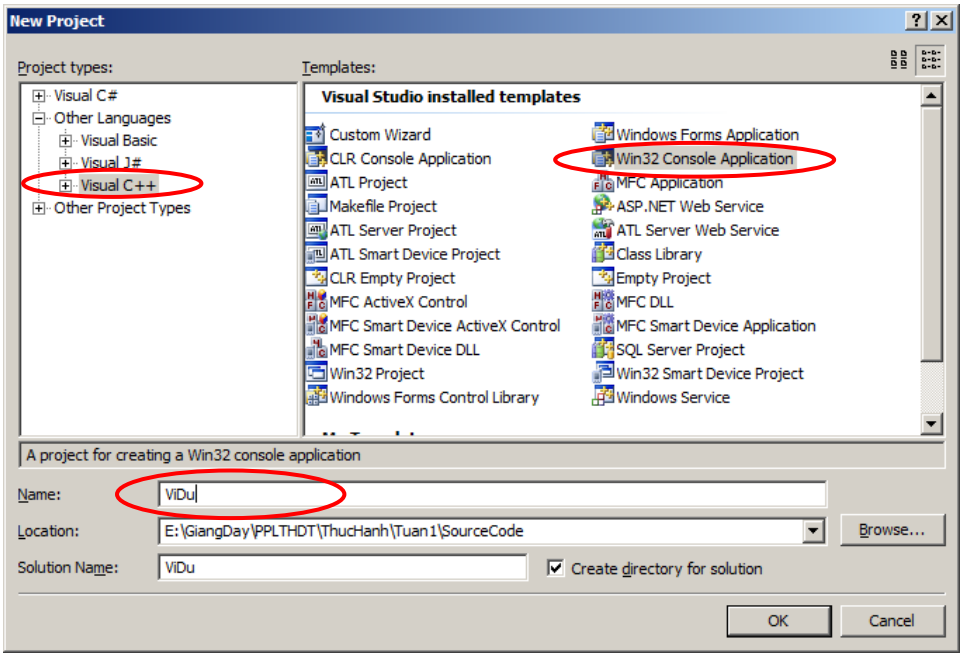


1.2.3. Tạo mới Project với kiểu ứng dụng Console trong Visual C++

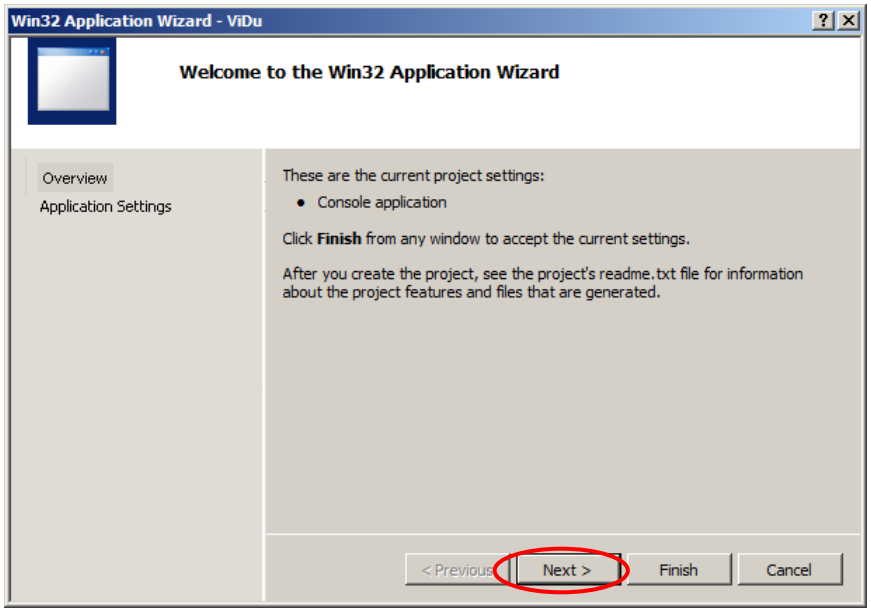
Bước 1: Trên menu bar, click chuột chọn **File** -> **New->Project** hoặc nhấn tổ hợp phím **Ctrl+Shift+N**



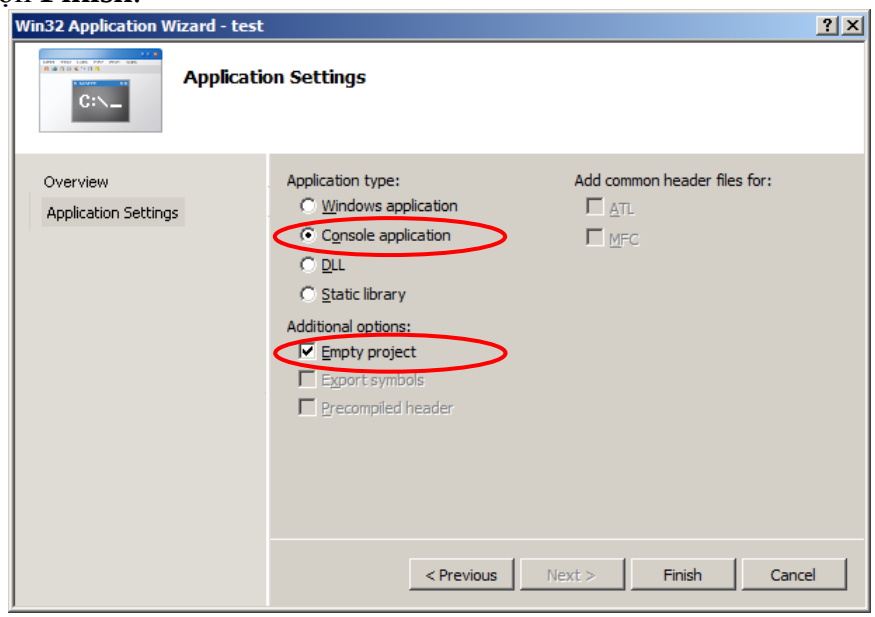
Bước 2: Trong **Project types**, chọn **Other Languages -> Visual C++**. Sau đó, trong **Templates**, chọn **Win32 Console Application**.



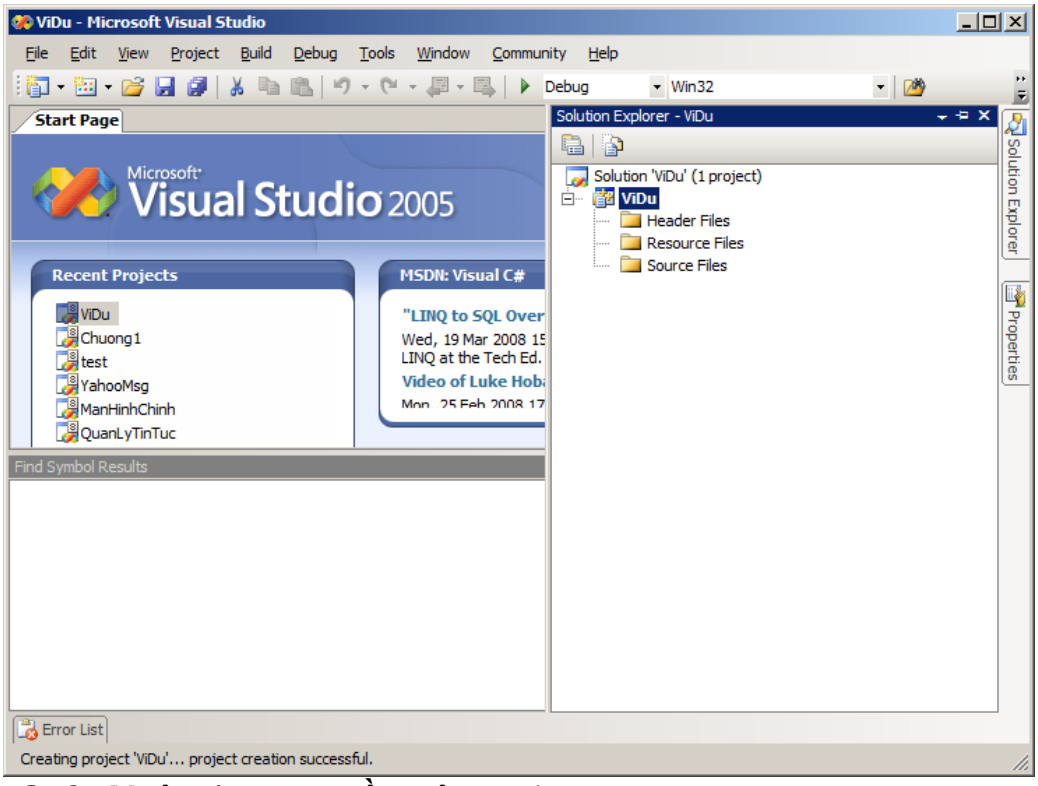
Bước 3: Click chuột vào **Next** để tiếp tục



Bước 4: Chọn **Application type** là **Console application** và chọn **Empty project**. Sau đó, chọn **Finish**.



Sau khi tạo xong, giao diện sẽ có dạng như sau:

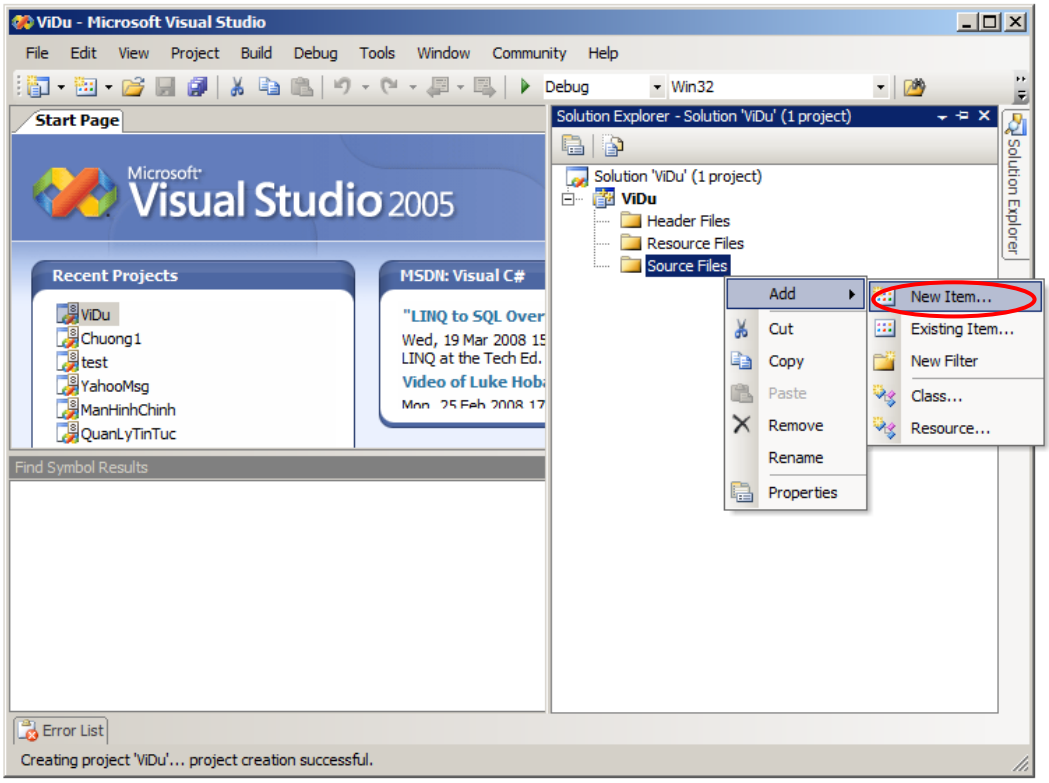


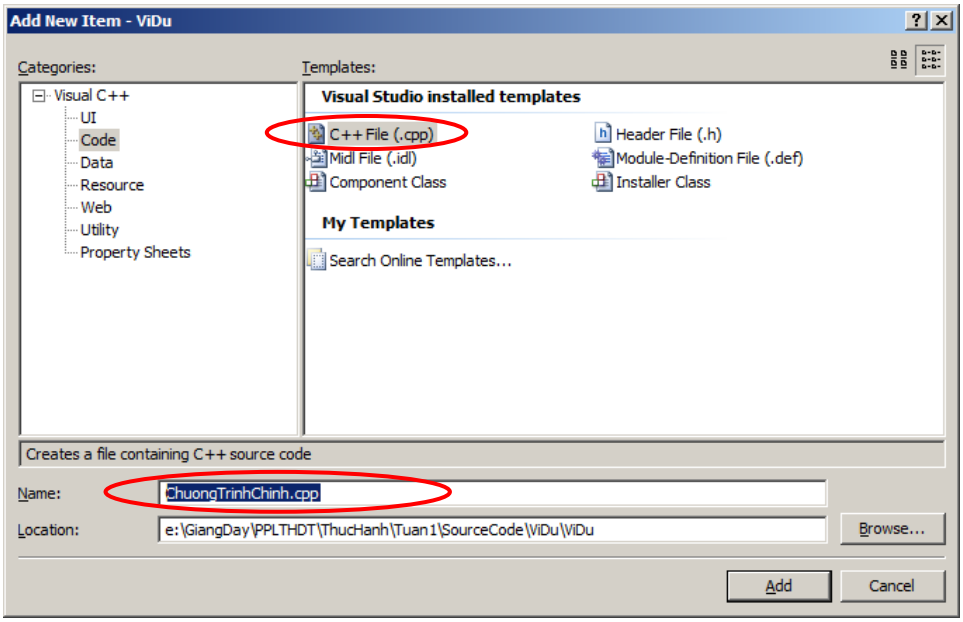
1.2.4. Quản lý tập tin mã nguồn trên Project

1.2.4.1. Thêm tập tin

* Thêm tập tin .cpp

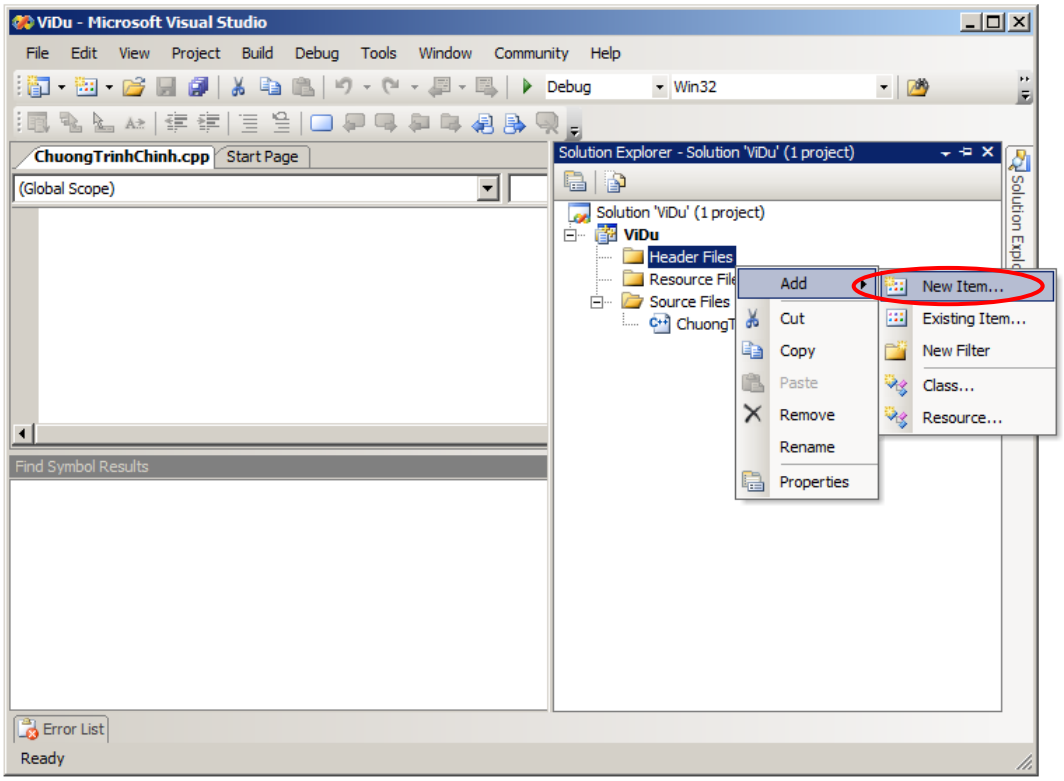
Trong project hiện hành, click phải chuột vào **Source Files**, chọn **Add -> New Item**

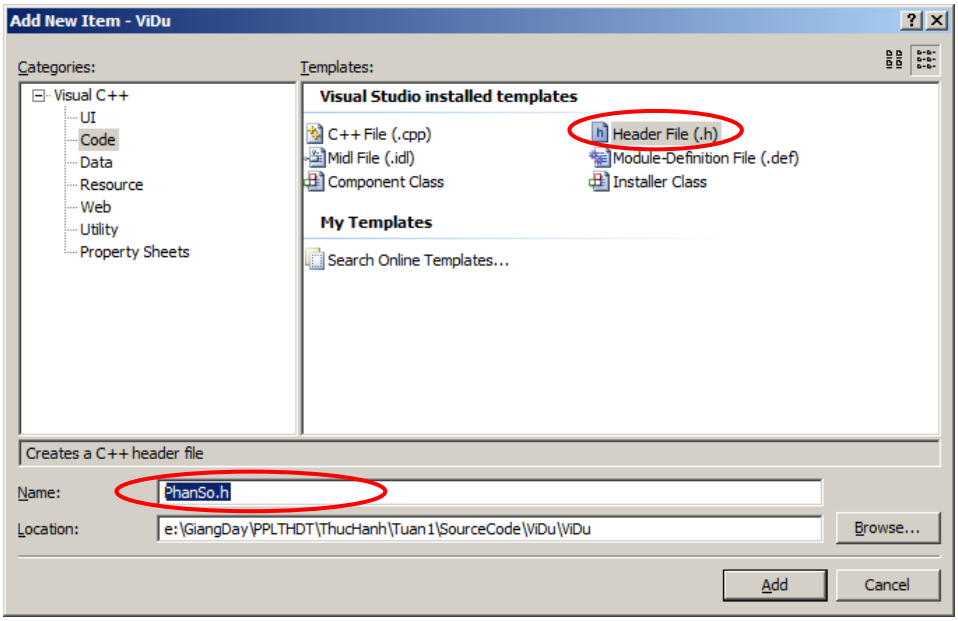




* Thêm tập tin .h

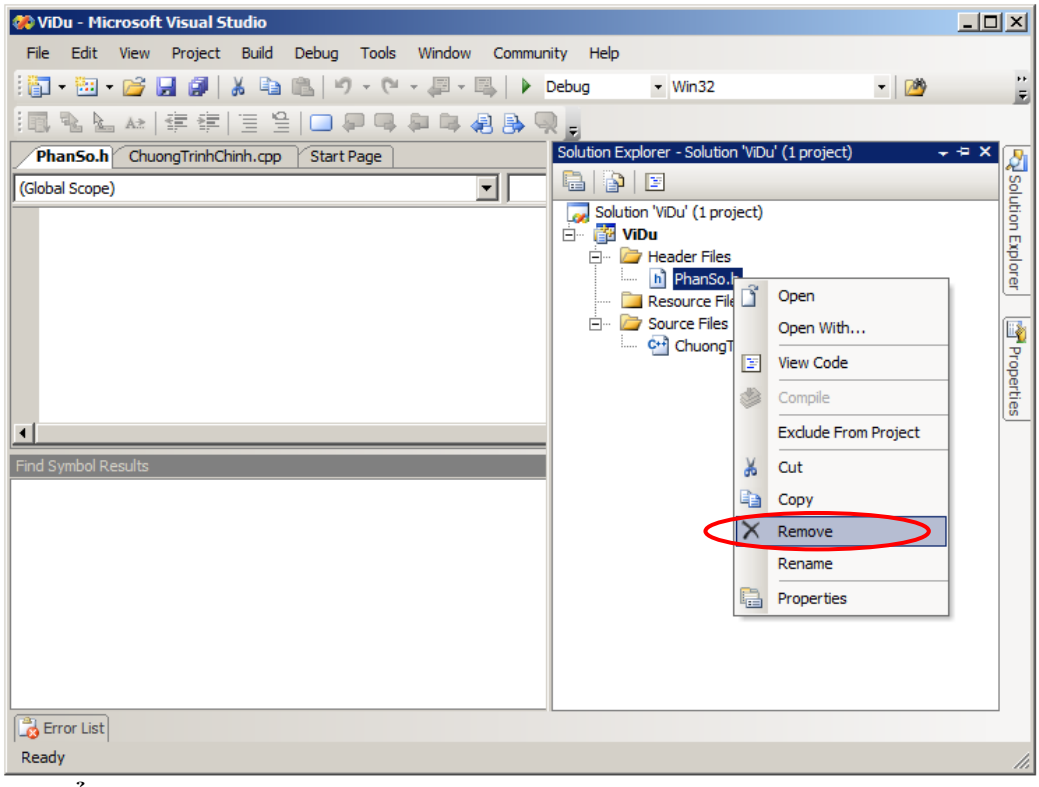
Trong project hiện hành, click phải chuột vào **Header Files**, chọn **Add -> New Item**





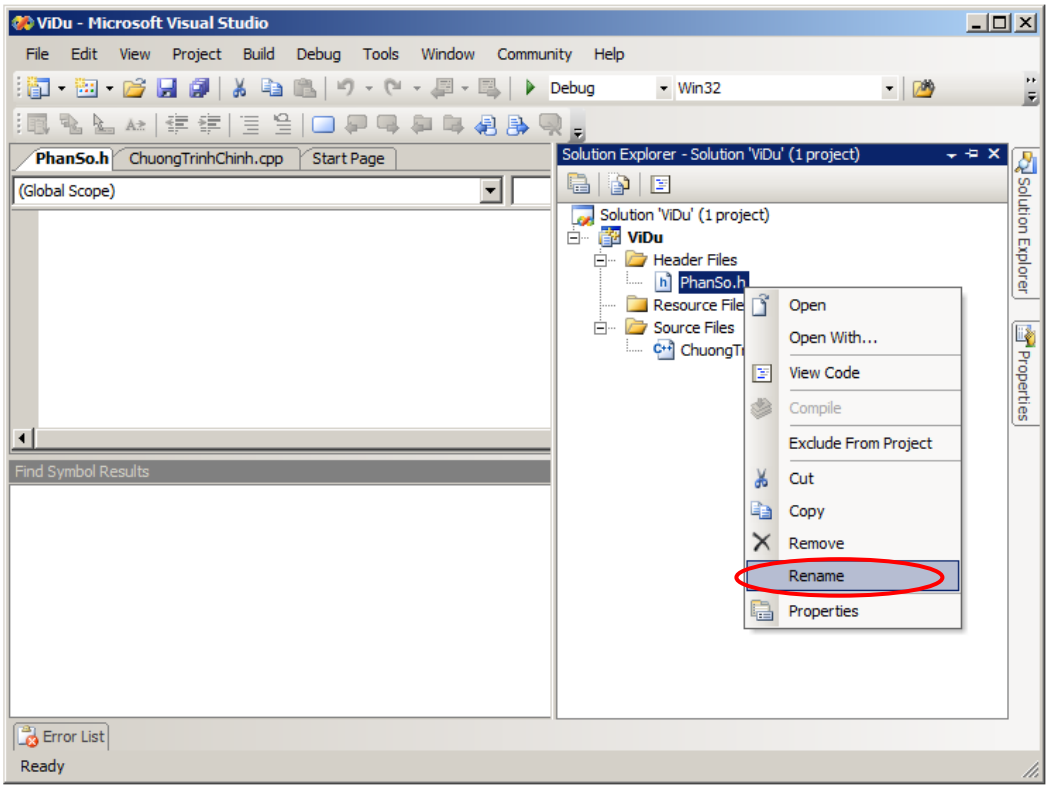
1.2.4.2. Xóa tập tin

Trong project hiện hành, click phải chuột vào tập tin cần xóa, chọn **Remove.**



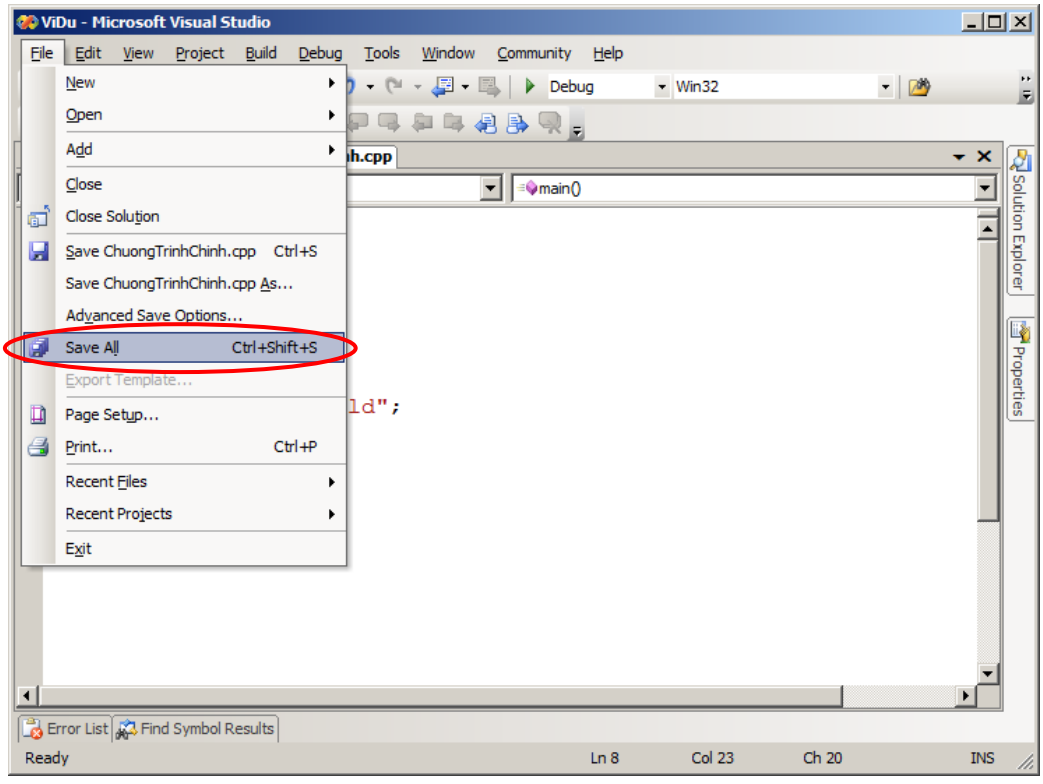
1.2.4.3. Đổi tên tập tin

Trong project hiện hành, click phải chuột vào tập tin cần xóa, chọn **Rename.**



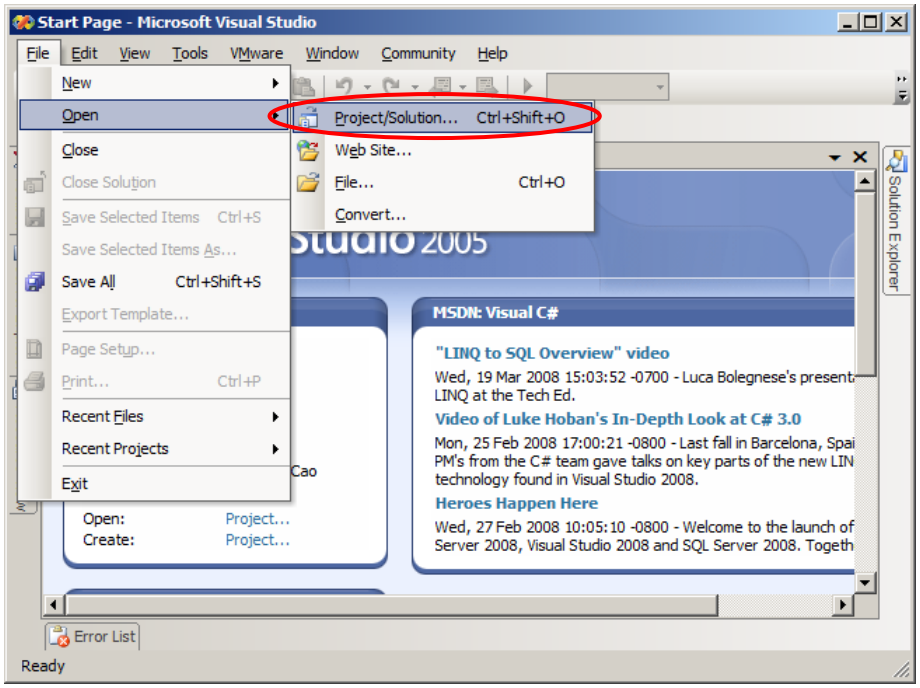
1.2.5. Lưu Project/Solution

Trên menu bar, click chuột chọn **File** -> **Save All**. Hoặc nhấn tổ hợp phím Ctrl+Shift+S**.**

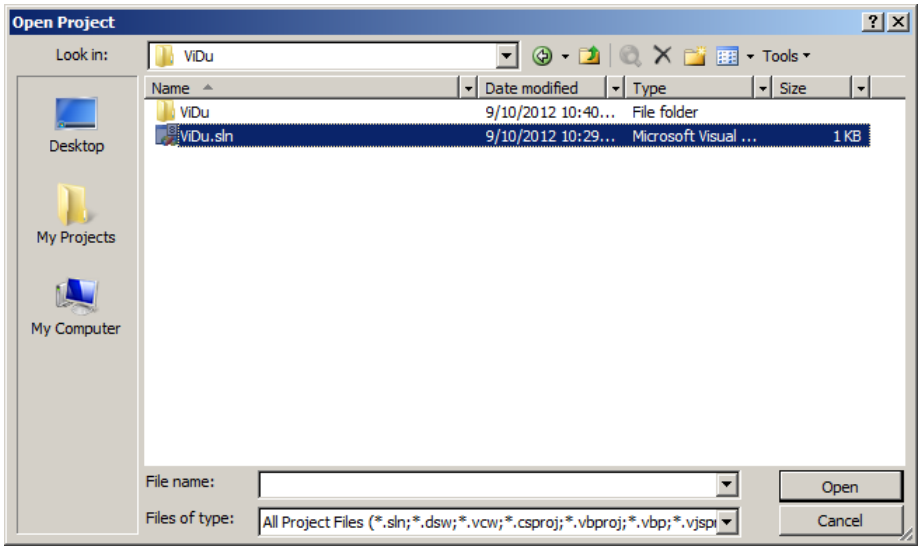


1.2.6. Mở Project/Solution

Trên menu bar, click chuột chọn **File** -> **Open** -> **Project/Solution**. Hoặc nhấn tổ hợp phím **Ctrl+Shift+O.**



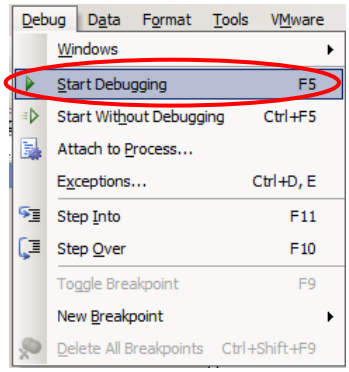
Hộp thoại (dialog) Open Project xuất hiện. Chọn tên tập tin Project cần mở. Sau đó click chuột vào nút **Open** để mở Project.



1.2.7. Biên dịch và chạy chương trình

Để biên dịch và thực thi ứng dụng của project hiện hành ta thực hiện như sau:

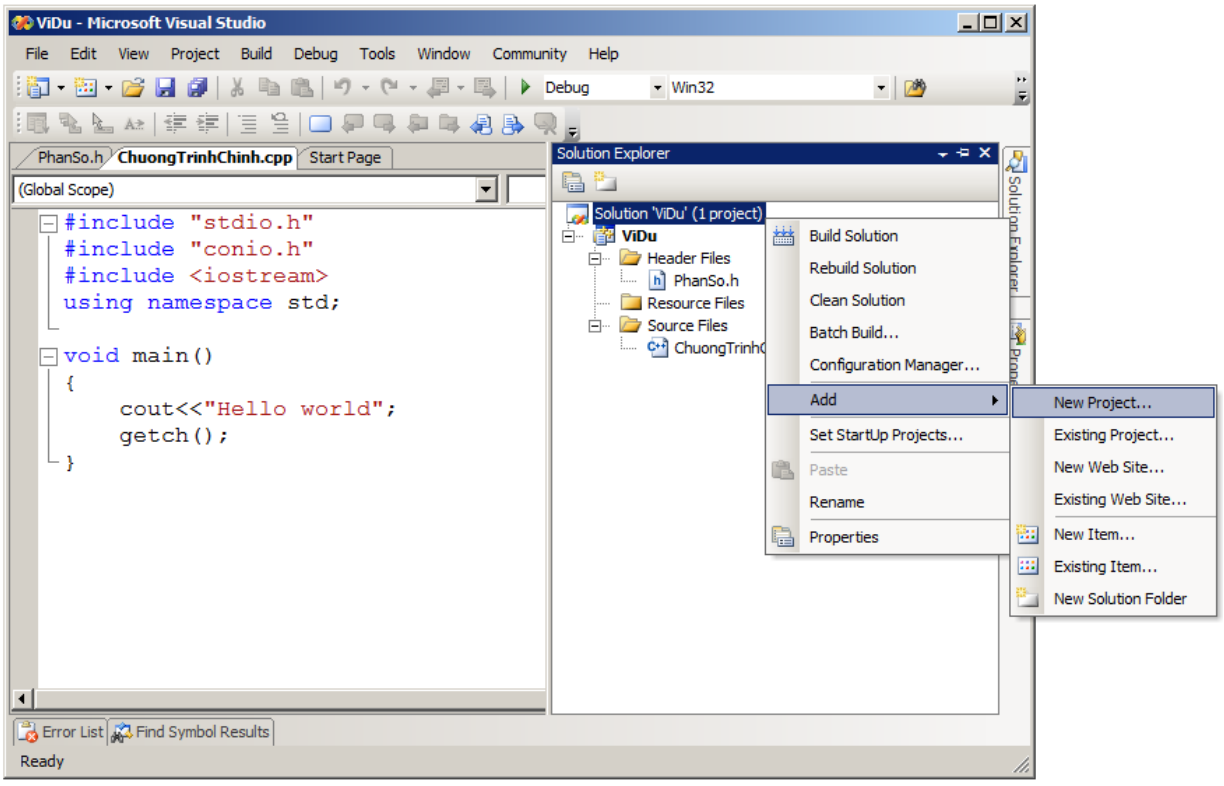
Trên menu bar, click chuột chọn **Debug** -> **Start Debugging.** Hoặc nhấn phím **F5**



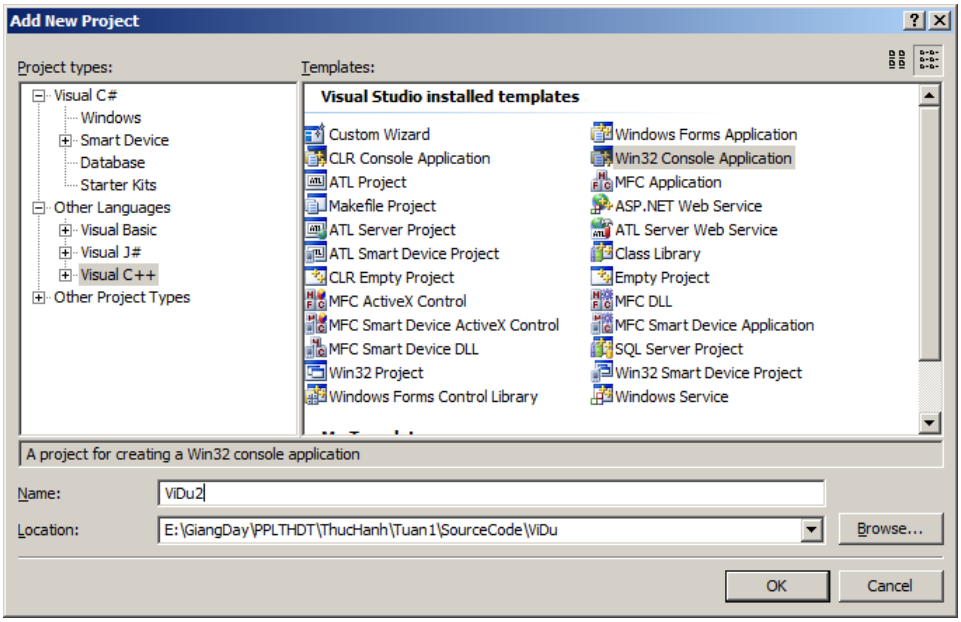
Nếu project có lỗi thì Visual Studio 2005 sẽ xuất hiện thông báo về lỗi
Ngược lại, project sẽ được biên dịch và thực thi.

1.2.8. Thêm Project trong Solution

Trong Solution Explorer, click phải chuột vào Solution, chọn Add -> New Project

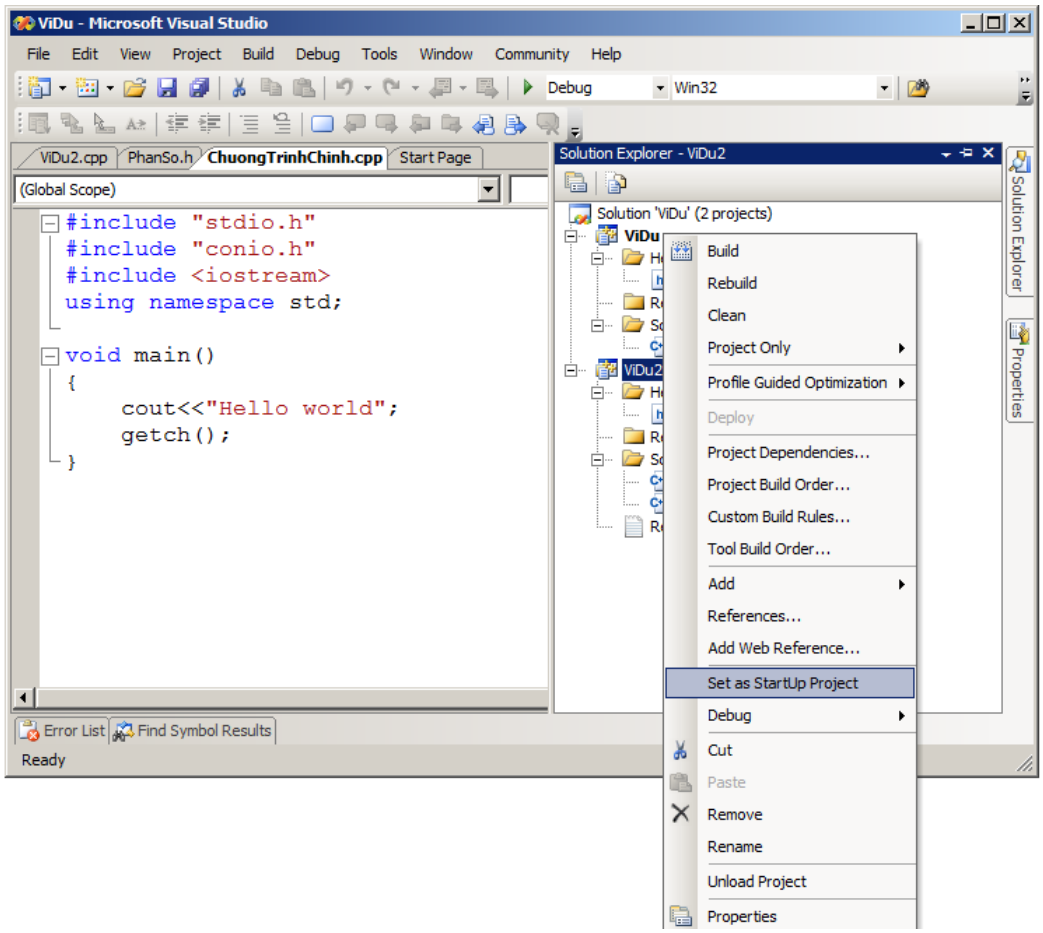


Màn hình thêm project xuất hiện



Thực hiện các bước tương tự như tạo mới.

Lưu ý: Để thiết lập project chạy lúc khởi động ứng dụng, trong Solution chọn Project cần thiết lập, click phải chuột chọn **Set as Startup Project**.



1.3. Các ví dụ

Ví dụ 1: Viết chương trình nhập n và tính tổng S(n) = 1 + 2 + 3 + ... + n

```
#include <stdio.h>
#include <conio.h>
#include <iostream>
using namespace std;

void Nhap (int &n);
long Tong (int n);
void Xuat (long tong);

void main ( )
{
	int n;

	Nhap (n);
	long kq = Tong (n);
	Xuat (kq);

	getch ( );
}



void Nhap (int &n)
{
	cout<<"Nhap n: ";
	cin>>n;
}

long Tong (int n)
{
	long s = 0;
	int i = 1;
	while (i <= n)
	{
		s = s + i;
		i = i + 1;
	}
	return s;
}

void Xuat (long tong)
{
	cout<<"\nTong: "<<tong;
}
```

Ví dụ 2: Tìm giá trị lớn nhất trong mảng một chiều các số nguyên a.

```
#include "stdio.h"
#include "conio.h"
```

```
#include <iostream>
using namespace std;

void NhapMang(int a[], int &n);
void XuatMang(int a[], int n);
int TimGiaTriLonNhat(int a[], int n);

void main()
{
	int a[100];
	int n;
	int max;
	int vtnn;

	NhapMang(a, n);
	XuatMang(a, n);

	max = TimGiaTriLonNhat(a, n);
	cout<<"\nGTLN: "<<max;

	getch();
}
void NhapMang(int a[], int &n)
{
	cout<<"Nhap n: ";
	cin>>n;

	for (int i = 0; i < n; i++)
	{
		cout<<"Nhap a["<<i<<"]:";
		cin>>a[i];
	}
}

void XuatMang(int a[], int n)
{
	for (int i = 0; i < n; i++)
	{
		cout<<a[i]<<" ";
	}
}

int TimGiaTriLonNhat(int a[], int n)
{
	int max;
	max = a[0];

	for (int i = 0; i < n; i++)
	{
		if (a[i] > max)
		{
			max = a[i];
		}
	}
```

```
        return max;
}
```

1.4. Bài tập

Hãy thực hiện các bài tập dưới đây trong cùng 1 Solution với tên <MSSV>_<HoTen>_BTTuan1 với mỗi bài là 1 Project với tên Bai_<xx>
Ví dụ:
Tên Solution: 306111999_LeVanQuang_BTTuan1
Tên các Project: Bai_01, Bai_02, Bai_03, ..., Bai_24

Đề bài:

Bài 01: Viết chương trình tìm số nhỏ nhất trong 3 số nguyên a, b, c.
Bài 02: Viết chương trình nhập vào số nguyên n có 4 chữ số. Hãy tìm chữ số nhỏ nhì.
Bài 03: Viết chương trình giải phương trình bậc 2: $ax^2 + bx + c = 0$.
tháng, năm).
Bài 04: Viết chương nhập vào một ngày (ngày, tháng, năm). Hãy cho biết ngày tiếp theo (ngày,
Ghi chú: Năm có thể là năm thường hoặc năm nhuận.

Bài 05: Nhập số nguyên dương n. Tính $S = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}$

Bài 06: Nhập số nguyên dương n. Tính $S = 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} \ldots + \frac{1}{2n+1} - \frac{1}{2n+2}$

Bài 07: Nhập số nguyên dương n. Tính $P = \sqrt{1 + \sqrt{2 + \cdots + \sqrt{(n-1) + \sqrt{n}}}}$

Bài 08: Nhập số nguyên dương n. Tính $P = \sqrt{n + \sqrt{(n-1) + \cdots + \sqrt{2 + \sqrt{1}}}}$

Bài 09: Viết chương trình nhập vào tuổi cha và tuổi con hiện tại. (tuổi cha > 2 lần tuổi con). Hỏi

sau bao nhiêu năm nữa tuổi cha bằng 2 lần tuổi con.

Bài 10: Cho dãy Fibonaci như sau:

$F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2}$

Hãy viết chương trình nhập vào số nguyên dương n. Hãy tính F(n)

Bài 11: Hãy tìm "vị trí giá trị dương nhỏ nhất" trong mảng một chiều các số thực. Nếu mảng không có giá trị dương thì trả về một giá trị ngoài đoạn [0, n-1] là -1 nhằm mô tả không có vị trí nào thỏa điều kiện.

Bài 12: Cho mảng một chiều các số nguyên. Hãy viết hàm tìm số chẵn lớn nhất nhỏ hơn mọi giá trị lẻ có trong mảng.

Bài 13: Cho mảng một chiều các số nguyên. Hãy viết hàm tìm số xuất hiện ít nhất trong mảng.

Bài 14: Hãy liệt kê các vị trí mà giá trị tại đó là số nguyên tố trong mảng một chiều các số nguyên.

Bài 15: Hãy liệt kê các vị trí mà giá trị tại các vị trí đó bằng giá trị dương nhỏ nhất trong mảng một chiều các số thực.

Bài 16: Tính tổng các giá trị là số nguyên tố trong mảng một chiều các số nguyên.

Bài 17: Tính trung bình cộng các số nguyên dương trong mảng một chiều các số nguyên.

Bài 18: Đếm số lượng giá trị có chữ số tận cùng bằng 5 trong mảng một chiều các số nguyên.

Bài 19: Đếm số lượng các giá trị phân biệt có trong mảng.

Bài 20: Hãy liệt kê các giá trị có số lần xuất hiện nhiều nhất trong mảng các số nguyên.

Bài 21: Hãy kiểm tra mảng có tăng dần hay không?

Bài 22: Hãy cho biết các phần tử trong mảng có bằng nhau không?

Bài 23: Hãy sắp xếp các giá trị trong mảng các số thực tăng dần.

Bài 24: Hãy sắp xếp các giá trị trong mảng các số nguyên giảm dần.

BÀI 2: LẬP TRÌNH ĐƠN THỂ VÀ TỔ CHỨC MÃ NGUỒN

2.1. Hàm trong C/C++

Một trong những vấn đề rất quan trọng khi viết các chương trình lớn là làm sao để phân chia và tổ chức mã nguồn chương trình một cách hợp lý để có thể dễ dàng bắt lỗi, chỉnh sửa, bảo trì, mở rộng và tái sử dụng sau này. Để giải quyết được vấn đề này, ngôn ngữ lập trình C/C++ cung cấp một loại chương trình con gọi là hàm. Khái niệm đơn giản nhất của hàm là: "hàm là một sự chia nhỏ của chương trình".

Lưu ý:

- Các thông số đầu vào của một hàm được gọi là tham số hàm.
- Phân loại tham số: có 2 loại tham số:

+ **Tham trị**: *không thay đổi giá trị sau lời gọi hàm.*

+ **Tham biến**: *giá trị của tham số thay đổi sau lời gọi hàm*.

2.2. Một số vấn đề nâng cao về hàm

2.2.1. Hàm trùng tên

Ngôn ngữ C/C++ cho phép viết các hàm trùng tên trong cùng một tập tin mã nguồn chương trình, khái niệm này gọi là *chồng hàm* (function overloading) hay là *đa xạ hàm* (functional polymorphism). Các hàm này phân biệt với nhau nhờ kiểu dữ liệu của tham số và số lượng các tham số.

Ví dụ 1: Viết hàm làm tròn số thực round (tương tự như hàm round trong excel)

```
#include <conio.h>
#include <iostream>
#include <cmath>
using namespace std;

double round(double x);
double round(double x, int n);

void main()
{
	double x;
	double y, z;
	int n;

	cout<<"Nhap so thuc x: ";
```

```
	cin>>x;
	cout<<"Nhap so nguyen duong n: ";
	cin>>n;

	y = round(x);
	cout<<"round(x) = "<<y<<endl;

	z = round(x, n);
	cout<<"round(x, n) = "<<z<<endl;

	getch();
}

double round(double x)
{
	double kq;
	if (x >= 0)
	{
		kq = floor(x+0.5);
	}
	else
	{
		kq = -floor(-x+0.5);
	}
	return kq;
}

//n: so le thap phan
double round(double x, int n)
{
	double kq;
	double lt;

	lt = pow((double)10,n); //luy thua 10^n
	x = x * lt;

	if (x >= 0)
	{
		kq = floor(x+0.5)/lt;
	}
	else
	{
		kq = -floor(-x+0.5)/lt;
	}
	return kq;
}
```

Lưu ý: hàm `floor` là hàm làm tròn nguyên cận dưới. Ví dụ: `floor(3.82134) = 3`, `floor(-3.82134)= - 4`

2.2.2. Hàm với giá trị mặc định của tham số

Ngôn ngữ C/C++ cho phép bỏ bớt số hàm trùng tên mà vẫn đảm bảo nhiều cách gọi hàm khác nhau bằng cách sử dụng giá trị mặc định cho các tham số vắng

mặt. Kỹ thuật này thường dùng khi viết một hàm tổng quát với nhiều tham số và khi một vài tham số nhận các giá trị đặc biệt (giá trị mặc định) thì hàm vẫn đúng giống như không có tham số đó.

Ví dụ 2: Cài đặt lại Ví dụ 1 với hàm làm tròn số gồm 2 tham số x và n.

Nhận xét: khi n = 0 thì có ý nghĩa giống như hàm `double round(double x)`

```
#include <conio.h>
#include <iostream>
#include <cmath>
using namespace std;

double round(double x, int n=0);

void main()
{
	double x;
	double y, z;
	int n;

	cout<<"Nhap so thuc x: ";
	cin>>x;
	cout<<"Nhap so nguyen duong n: ";
	cin>>n;

	y = round(x);
	cout<<"round(x) = "<<y<<endl;

	z = round(x, n);
	cout<<"round(x, n) = "<<z<<endl;

	getch();
}

//n: so le thap phan
double round(double x, int n)
{
	double kq;
	double lt;

	lt = pow((double)10,n); //luy thua 10^n
	x = x * lt;

	if (x >= 0)
	{
		kq = floor(x+0.5)/lt;
	}
	else
	{
		kq = -floor(-x+0.5)/lt;
	}
```

```
	return kq;
}
```

2.2.3. Hàm có tham số là kiểu dữ liệu

Xét ví dụ sau:

Ví dụ 3: Tìm giá trị lớn nhất giữa a và b. Với a và b có kiểu dữ liệu lần lượt là số nguyên (int), số thực (float), ký tự (char).

```
#include <conio.h>
#include <iostream>
using namespace std;

int Max(int a, int b)
{
	int kq;
	if (a >= b)
		kq = a;
	else
		kq = b;

	return kq;
}
float Max(float a, float b)
{
	float kq;
	if (a >= b)
		kq = a;
	else
		kq = b;

	return kq;
}
char Max(char a, char b)
{
	char kq;
	if (a >= b)
		kq = a;
	else
		kq = b;

	return kq;
}


void main()
{
	int a, b;
	float c, d;
	char x, y;

	a = 5;
	b = 7;
```

```
	cout<<"Gia tri lon nhat: "<<Max(a,b);

	c = 7.3;
	d = 8.1;
	cout<<"\nGia tri lon nhat: "<<Max(c,d);

	x = 'k';
	y = 'l';
	cout<<"\nGia tri lon nhat: "<<Max(x,y);
	getch();
}
```

Nhận xét: Mỗi khi có nhu cầu tìm max (giá trị lớn nhất) giữa 2 biến có kiểu dữ liệu mới thì ta phải viết một hàm cho kiểu dữ liệu đó. Hơn nữa, những hàm này có cách cài đặt mã nguồn đều giống nhau, chỉ khác nhau ở kiểu dữ liệu. Điều này gây nên sự trùng lặp mã nguồn trong chương trình và làm cho kích thước chương trình lớn. Một giải pháp hiệu quả để giải quyết việc trùng lặp mã nguồn là áp dụng các hàm được tham số hóa bởi kiểu dữ liệu. Trong ngôn ngữ C/C++ hỗ trợ khả năng dùng các ***khuôn hàm*** (functional template), trong đó, kiểu dữ liệu có thể được xem là một tham số.

Ta viết lại Ví dụ 3 như sau:

```
#include<conio.h>
#include <iostream>
using namespace std;

template <class T>
T Max(T a, T b)
{
	T kq;
	if (a >= b)
		kq = a;
	else
		kq = b;

	return kq;
}
void main()
{
	int a, b;
	float c, d;
	char x, y;

	a = 5;
	b = 7;
	cout<<"Gia tri lon nhat: "<<Max(a, b);

	c = 7.3;
```

```
	d = 8.1;
	cout<<"\nGia tri lon nhat: "<<Max(c,d);

	x = 'k';
	y = 'l';
	cout<<"\nGia tri lon nhat: "<<Max(x,y);
	getch();
}
```

Lưu ý: Kiểu dữ liệu T được xem như là một kiểu dữ liệu cơ bản (float, int, char, long, double, ...)

2.3. Tổ chức hàm với nhiều tập tin mã nguồn

Một chương trình ứng dụng thực tế viết bằng ngôn ngữ C/C++ thường bao gồm một tập hợp nhiều tập tin mã nguồn. Mỗi tập tin mã nguồn là mô tả hay cài đặt của một số hàm liên quan đến nhau.

Thông thường các chương trình C/C++ gồm 2 loại tập tin mã nguồn:

- ***Tập tin dạng khai báo *.h***: là phần mô tả giao tiếp lập trình (programming interface) chứa các lệnh khai báo (prototype) và các hằng số.
- ***Tập tin cài đặt mã nguồn *.cpp***: là phần cài đặt mã nguồn (implementation), chứa mã nguồn chi tiết của mỗi hàm được khai báo trong phần giao tiếp lập trình. Ở phần đầu tập tin có một hay nhiều chỉ thị `#include` để tham chiếu đến các tập tin ***.h**.

Ví dụ 4: Viết chương trình tính tổng 2 phân số.

***Tập tin PhanSo.h**

```
#ifndef _PHANSO_
#define _PHANSO_

struct PhanSo
{
	int TuSo;
	int MauSo;
};

void NhapPhanSo(PhanSo &ps);
void XuatPhanSo(PhanSo ps);
PhanSo TinhTong(PhanSo A, PhanSo B);

#endif
```

***Tập tin PhanSo.cpp**

```
#include "PhanSo.h"
```

```
#include <iostream>
using namespace std;

void NhapPhanSo(PhanSo &ps)
{
	cout<<"Nhap tu so: ";
	cin>>ps.TuSo;
	cout<<"Nhap mau so: ";
	cin>>ps.MauSo;
}
void XuatPhanSo(PhanSo ps)
{
	cout<<ps.TuSo<<"/"<<ps.MauSo;
}

PhanSo TinhTong(PhanSo A, PhanSo B)
{
	PhanSo kq;
	kq.TuSo = A.TuSo*B.MauSo + B.TuSo*A.MauSo;
	kq.MauSo = A.MauSo * B.MauSo;

	return kq;
}
```

***Tập tin ChuongTrinhChinh.cpp**

```
#include "PhanSo.h"
#include "conio.h"
#include <iostream>
using namespace std;

void main()
{
	PhanSo A;
	PhanSo B;
	PhanSo C;

	cout<<"Nhap phan so A: "<<endl;
	NhapPhanSo(A);

	cout<<"Nhap phan so B: "<<endl;
	NhapPhanSo(B);

	C = TinhTong(A, B);

	cout<<"Tong 2 phan so: ";
	XuatPhanSo(C);

	getch();
}
```

2.4. Bài tập

Hãy thực hiện các bài tập dưới đây trong cùng 1 Solution với tên **<MSSV>_<HoTen>_BTTuan2** với mỗi bài là 1 Project với tên **Bai_<xx>**

Ví dụ:

- Tên Solution: 306111999_LeVanQuang_BTTuan2
- Tên các Project: Bai_01, Bai_02, Bai_03, ..., Bai_10

Đề bài:

Bài 01: Tính độ lớn của một vector sau:

- Vector một chiều V=(x): $|V| = x$
- Vector hai chiều V=(x, y): $|V| = \sqrt{x^2 + y^2}$
- Vector ba chiều V=(x, y,z): $|V| = \sqrt{x^2 + y^2 + z^2}$

Yêu cầu: cài đặt hàm trùng tên (tương tự Ví dụ 1) với các khai báo như sau:

```
float TinhDoLonVector(float x);
float TinhDoLonVector(float x, float y);
float TinhDoLonVector(float x, float y, float z);
```

Bài 02: Viết hàm hoán vị 2 số a và b. Với a và b có kiểu dữ liệu cơ bản: int, long, float, double.

```
void HoanVi(int &a, int &b);
void HoanVi(long &a, long &b);
...
```

Bài 03: Viết hàm hoán vị 2 phần tử a và b. Với a và b là kiểu dữ liệu bất kỳ: char, int, long, float, double, PhanSo, ...

Gợi ý: sử dụng template function.

Bài 04: Viết hàm tìm giá trị lớn nhất giữa 3 phần tử a, b và c. Với a, b và c có kiểu dữ liệu cơ bản: char, int, long, float, double, ...

Bài 05: Viết hàm tìm giá trị nhỏ nhất giữa 4 phần tử a, b, c và d. Với a, b, c và d có kiểu dữ liệu cơ bản: char, int, long, float, double, ...

Bài 06: Cho mảng một chiều a có kiểu dữ liệu T có n phần tử. Với T là kiểu dữ liệu cơ bản.

Hãy viết các hàm sau:

a. Nhập mảng a

b. Xuất mảng a

c. Tính tổng các phần tử của mảng a.

d. Tìm giá trị lớn nhất trong mảng a.

e. Đếm số lượng các phần tử có giá trị dương.

f. Sắp xếp mảng tăng dần.

Bài 07: Dựa trên Ví dụ 4 - Phân số, hãy xây dựng tiếp các hàm sau:

a. Tính hiệu 2 phân số.

b. Tính tích 2 phân số.

c. Tính thương 2 phân số.

d. Rút gọn phân số

e. So sánh 2 phân số.

Gợi ý: tổ chức mã nguồn gồm các tập tin *.h, *.cpp.

Bài 08: Cho đơn thức f(x) = ax + b. Hãy thực hiện các yêu cầu sau:

a. Khai báo kiểu cấu trúc DonThuc

b. Viết hàm nhập đơn thức

c. Viết hàm xuất đơn thức

d. Viết hàm tính tổng 2 đơn thức A và B

e. Viết hàm tính hiệu 2 đơn thức A và B

f. Viết hàm giải phương trình khi f(x) = 0

Gợi ý: tổ chức mã nguồn gồm các tập tin DonThuc.h, DonThuc.cpp, ChuongTrinhChinh.cpp

Bài 09: Cho số phức z = a + b*i với a là phần thực, b là phần ảo, i là đơn vị ảo và $i^2 = -1$. Hãy thực hiện các yêu cầu sau:

a. Khai báo kiểu cấu trúc SoPhuc

b. Viết hàm nhập số phức

c. Viết hàm xuất số phức

d. Viết hàm tính tổng 2 số phức A và B

e. Viết hàm tính hiệu 2 số phức A và B

f. Viết hàm tính tích 2 số phức A và B

g. Viết hàm tính thương 2 số phức A và B.

h. Viết hàm so sánh 2 số phức A và B.

Gợi ý: tổ chức mã nguồn gồm các tập tin SoPhuc.h, SoPhuc.cpp, ChuongTrinhChinh.cpp

```
struct SoPhuc
{
	float a;
	float b;
};
```

Bài 10: Cho toạ độ 3 đỉnh của một tam giác trong mặt phẳng Oxy. Hãy thực hiện các yêu cầu sau:

a. Khai báo kiểu cấu trúc Diem

b. Khai báo kiểu cấu trúc TamGiac

c. Viết hàm nhập điểm (đỉnh)

d. Viết hàm xuất điểm

e. Viết hàm tính khoảng cách giữa 2 điểm

f. Viết hàm nhập tam giác

g. Viết hàm xuất tam giác

h. Viết hàm tính diện tích tam giác

i. Viết hàm tính trọng tâm của tam giác

Gợi ý: tổ chức mã nguồn gồm các tập tin Diem.h, Diem.cpp, TamGiac.h, TamGiac.cpp, ChuongTrinhChinh.cpp

```
struct Diem
{
	float x;
	float y;
};
```

```
struct TamGiac
{
	Diem A;
	Diem B;
	Diem C;
};
```

BÀI 3: GIỚI THIỆU VỀ ĐỐI TƯỢNG VÀ LỚP ĐỐI TƯỢNG TRONG C++

3.1. Khái niệm

Đối tượng (object) là một thực thể lưu trữ các thành phần dữ liệu và thành phần xử lý trong thời gian chạy chương trình (runtime). Thành phần dữ liệu được gọi là thuộc tính (attribute), thành phần xử lý được gọi là phương thức (method).

Thuộc tính nằm bên trong đối tượng, nó mô tả tính chất và lưu trữ thông tin đối tượng. Phương thức là phương tiện để sử dụng một đối tượng. Để đơn giản, ta có thể hiểu: thuộc tính của đối tượng là các biến còn phương thức của đối tượng là các hàm liên quan.

Tập hợp các đối tượng có cùng kiểu thuộc tính và phương thức tạo thành lớp đối tượng.

Lập trình hướng đối tượng có 4 tính chất quan trọng sau:

+ Tính trừu tượng (abstraction)

+ Tính đóng gói (encapsulation)

+ Tính kế thừa (inheritance)

+ Tính đa hình (polymorphism)

Lớp đối tượng trong C++ được khai báo theo cấu trúc sau:

```
class <Tên lớp>
{
private:
<Khai báo thành phần riêng tư>
protected:
<Khai báo thành phần bảo tồn>
public:
<Khai báo thành phần công cộng>
};
```

Ví dụ:

```
class PhanSo
{
private:
      int m_iTuSo;
      int m_iMauSo;
```

```
public:
      void NhapPhanSo();
      . . .
}
```

3.2. Phạm vi (scope)

Phạm vi (scope) của các thành phần bên trong lớp đối tượng cho biết phạm vi hoạt động của những thành phần đó. Có 3 loại tầm vực dành cho các thành phần trong lớp đối tượng: public, private, và protected. Trong phạm vi bài học này, ta chỉ đề cập 2 loại phạm vi là: public, private.

Ý nghĩa về cách sử dụng các phạm vi:

	Bên trong lớp	Bên ngoài lớp
private	X	
public	X	X

3.3. Con trỏ this

Con trỏ this được sử dụng trong một phương thức và nó đại diện cho con trỏ đang trỏ đến đối tượng gọi thực hiện phương thức đó.

3.4. Toán tử ::

Toán tử :: hay còn gọi là toán tử phân giải miền (scope resolution operator) được sử dụng để chỉ ra một cách tường minh một thuộc tính hoặc phương thức là thuộc một lớp nào đó.

Khi các phương thức của một lớp được định nghĩa bên ngoài lớp, chúng ta phải sử dụng toán tử phân giải miền để chỉ ra tường minh phương thức của lớp.

3.5. Ví dụ

Một lớp đối tượng trong C++ thường được lưu trữ trong 2 tập tin: *.h và *.cpp. Tập tin *.h chứa phần khai báo của lớp đối tượng, trong khi tập tin *.cpp chứa phần định nghĩa.

Ví dụ: Viết chương trình tính tổng 2 phân số.

***Tập tin PhanSo.h**

```
#ifndef _PHANSO_
#define _PHANSO_
```

```
class PhanSo
{
private:
	int m_iTuSo;
	int m_iMauSo;
public:
	void SetTuSo(int iTuSo);
	void SetMauSo(int iMauSo);
	int  GetTuSo();
	int  GetMauSo();
	void NhapPhanSo();
	void XuatPhanSo();
	PhanSo TinhTong(PhanSo A, PhanSo B);
};
#endif
```

***Tập tin PhanSo.cpp**

```
#include "PhanSo.h"
#include <iostream>
using namespace std;

void PhanSo::SetTuSo(int iTuSo)
{
	this->m_iTuSo = iTuSo;
}
void PhanSo::SetMauSo(int iMauSo)
{
	this->m_iMauSo = iMauSo;
}
int PhanSo::GetTuSo()
{
	return this->m_iTuSo;
}
int PhanSo::GetMauSo()
{
	return this->m_iMauSo;
}
void PhanSo::NhapPhanSo()
{
	cout<<"Nhap tu so: ";
	cin>>this->m_iTuSo;
	cout<<"Nhap mau so: ";
	cin>>this->m_iMauSo;
```

```
}
void PhanSo::XuatPhanSo()
{
	cout<<this->m_iTuSo<<"/"<<this->m_iMauSo;
}
PhanSo PhanSo::TinhTong(PhanSo A, PhanSo B)
{
	PhanSo ps;
	int iTuSo;
	int iMauSo;

	iTuSo = A.GetTuSo()*B.GetMauSo() + B.GetTuSo()*A.GetMauSo();
	iMauSo = A.GetMauSo() * B.GetMauSo();

	ps.SetTuSo(iTuSo);
	ps.SetMauSo(iMauSo);

	return ps;
}
```

***Tập tin ChuongTrinhChinh.cpp**

```
#include "PhanSo.h"
#include "conio.h"
#include <iostream>
using namespace std;

void main()
{
	PhanSo A, B, C;

	cout<<"Nhap phan so A:\n";
	A.NhapPhanSo();

	cout<<"Nhap phan so B:\n";
	B.NhapPhanSo ();

	C = A.TinhTong(A, B);

	cout<<"Tong 2 phan so: ";
	C.XuatPhanSo();

	getch();
```

```
}
```

3.6. Bài tập

Hãy thực hiện các bài tập dưới đây trong cùng 1 Solution với tên **<MSSV>_<HoTen>_BTTuan3** với mỗi bài là 1 Project với tên **Bai_<xx>**

Ví dụ:

- Tên Solution: 306111999_LeVanQuang_BTTuan3
- Tên các Project: Bai_01, Bai_02, Bai_03

Đề bài:

Bài 01: Cho lớp đối tượng Phân số, hãy thực hiện các yêu cầu sau:

a. Khai báo lớp đối tượng PhanSo

b. Viết phương thức nhập phân số

c. Viết phương thức xuất phân số

d. Viết phương thức tính tổng 2 phân số

e. Viết phương thức tính hiệu 2 phân số.

f. Viết phương thức tính tích 2 phân số.

g. Viết phương thức tính thương 2 phân số.

h. Viết phương thức rút gọn phân số

i. Viết phương thức so sánh 2 phân số.

Gợi ý: tổ chức mã nguồn gồm các tập tin *.h, *.cpp.

Bài 02: Cho đơn thức f(x) = ax + b. Hãy thực hiện các yêu cầu sau:

a. Khai báo lớp đối tượng DonThuc

b. Viết phương thức nhập đơn thức

c. Viết phương thức xuất đơn thức

d. Viết phương thức tính tổng 2 đơn thức A và B

e. Viết phương thức tính hiệu 2 đơn thức A và B

f. Viết phương thức giải phương trình khi f(x) = 0

Gợi ý: tổ chức mã nguồn gồm các tập tin DonThuc.h, DonThuc.cpp, ChuongTrinhChinh.cpp

Bài 03: Cho số phức z = a + b*i với a là phần thực, b là phần ảo, i là đơn vị ảo và $i^2 = -1$. Hãy thực hiện các yêu cầu sau:

a. Khai báo lớp đối tượng SoPhuc

b. Viết phương thức nhập số phức

c. Viết phương thức xuất số phức

d. Viết phương thức tính tổng 2 số phức A và B

e. Viết phương thức tính hiệu 2 số phức A và B

f. Viết phương thức tính tích 2 số phức A và B

g. Viết phương thức tính thương 2 số phức A và B.

h. Viết phương thức so sánh 2 số phức A và B.

Gợi ý: tổ chức mã nguồn gồm các tập tin SoPhuc.h, SoPhuc.cpp, ChuongTrinhChinh.cpp

BÀI 4: KHỞI TẠO, HỦY VÀ CÁC TOÁN TỬ TRONG LỚP HƯỚNG ĐỐI TƯỢNG

4.1. Khởi tạo và hủy đối tượng

4.1.1. Toán tử `new`

Toán tử new dùng để khởi tạo một đối tượng mới, nếu khởi tạo thành công thì biến đối tượng sẽ nhận một giá trị khác NULL.

Ví dụ:

```
PhanSo* ps = new PhanSo();
```

4.1.2. Toán tử `delete`

Toán tử delete dùng để hủy những đối tượng đã khởi tạo trước đó khi không còn được sử dụng.

Ví dụ:

```
...
delete ps;
ps = NULL;
```

4.1.3. Ví dụ

Sử dụng lại ví dụ về phân số của hướng dẫn thực hành 3 để minh họa 2 toán tử new, delete.

***Tập tin PhanSo.h**

```
#ifndef _PHANSO_
#define _PHANSO_

class PhanSo
{
private:
	int m_iTuSo;
	int m_iMauSo;
public:
	void SetTuSo(int iTuSo);
	void SetMauSo(int iMauSo);
	int  GetTuSo();
	int  GetMauSo();
	void NhapPhanSo();
	void XuatPhanSo();
};
#endif
```

***Tập tin PhanSo.cpp**

```
#include "PhanSo.h"
#include <iostream>
using namespace std;

void PhanSo::SetTuSo(int iTuSo)
{
	this->m_iTuSo = iTuSo;
}
void PhanSo::SetMauSo(int iMauSo)
{
	this->m_iMauSo = iMauSo;
}
int PhanSo::GetTuSo()
{
	return this->m_iTuSo;
}
int PhanSo::GetMauSo()
{
	return this->m_iMauSo;
}
void PhanSo::NhapPhanSo()
{
	cout<<"Nhap tu so: ";
	cin>>this->m_iTuSo;
	cout<<"Nhap mau so: ";
	cin>>this->m_iMauSo;
}
void PhanSo::XuatPhanSo()
{
	cout<<this->m_iTuSo<<"/"<<this->m_iMauSo;
}
```

***Tập tin ChuongTrinhChinh.cpp**

```
#include "conio.h"
#include <iostream>
using namespace std;

void main()
{
	PhanSo *A = new PhanSo();

	if (A != NULL)
```

```
	{
		A->NhapPhanSo();
		A->XuatPhanSo();

		delete A;
		A = NULL;
	}

	getch();
}
```

Ghi chú: Liên quan đến khởi tạo các lớp đối tượng bằng toán tử new sẽ được đề cập chi tiết hơn ở bài sau (Các phương thức khởi tạo, phương thức hủy).

4.2. Các toán tử trong lớp hướng đối tượng

Trong hướng đối tượng cho phép chúng ta định nghĩa các toán tử cho lớp đối tượng, chẳng hạn như các toán tử: +, -, *, /, >, >=, <, <=, ==

Ngoài ra, để mở rộng việc nhập xuất cho loại đối tượng mới thì cho phép ta sử dụng toán tử nhập >>, toán tử xuất <<.

```
PhanSo ps;
cin>>ps;
cout<<ps;
```

Ví dụ 2: Cho lớp PhanSo, hãy định nghĩa các toán tử

+ Toán tử tính toán: +, -

+ Toán tử so sánh: >

+ Toán tử nhập: >>

+ Toán tử xuất <<

***Tập tin PhanSo.h**

```
#ifndef _PHANSO_
#define _PHANSO_

#include <iostream>
using namespace std;

class PhanSo
{
private:
	int m_iTuSo;
	int m_iMauSo;
public:
	void SetTuSo(int iTuSo);
	void SetMauSo(int iMauSo);
```

```
	int GetTuSo();
	int GetMauSo();

	// cong
	PhanSo operator+(PhanSo ps);
	// tru
	PhanSo operator-(PhanSo ps);

	//so sanh
	bool operator>(PhanSo ps);

	// toan tu nhap
	friend istream& operator>>(istream& inDevice, PhanSo& ps);

	// toan tu xuat
	friend ostream& operator<<(ostream& outDevice, PhanSo &ps);

};


#endif
```

***Tập tin PhanSo.cpp**

```
#include "PhanSo.h"
#include <iostream>
using namespace std;

void PhanSo::SetTuSo(int iTuSo)
{
	this->m_iTuSo = iTuSo;
}
void PhanSo::SetMauSo(int iMauSo)
{
	this->m_iMauSo = iMauSo;
}
int PhanSo::GetTuSo()
{
	return this->m_iTuSo;
}
int PhanSo::GetMauSo()
{
	return this->m_iMauSo;
}
PhanSo PhanSo::operator+(PhanSo ps)
{
	PhanSo kq;

	kq.m_iTuSo = this->m_iTuSo*ps.m_iMauSo + ps.m_iTuSo*this-
>m_iMauSo;
	kq.m_iMauSo = this->m_iMauSo * ps.m_iMauSo;

	return kq;
```

```
}

PhanSo PhanSo::operator-(PhanSo ps)
{
	PhanSo kq;

	kq.m_iTuSo = this->m_iTuSo*ps.m_iMauSo - ps.m_iTuSo*this-
>m_iMauSo;
	kq.m_iMauSo = this->m_iMauSo * ps.m_iMauSo;

	return kq;
}

bool PhanSo::operator>(PhanSo ps)
{
	bool kq;
	if (this->m_iTuSo*ps.m_iMauSo > ps.m_iTuSo*this->m_iMauSo)
		kq = true;
	else
		kq = false;

	return kq;
}

istream& operator>>(istream& inDevice, PhanSo& ps)
{
	cout<<"Nhap tu so: ";
	inDevice>>ps.m_iTuSo;
	cout<<"Nhap mau so: ";
	inDevice>>ps.m_iMauSo;
	return inDevice;
}

ostream& operator<<(ostream& outDevice, PhanSo &ps)
{
	outDevice<<ps.m_iTuSo<<"/"<<ps.m_iMauSo;
	return outDevice;
}
```

***Tập tin ChuongTrinhChinh.cpp**

```
#include "PhanSo.h"
#include "conio.h"
#include <iostream>
using namespace std;

void main()
{

	PhanSo A, B, C;
	cout<<"\nNhap phan so A: \n";
	cin>>A;

	cout<<"\nNhap phan so B: \n";
```

```
	cin>>B;

	C = A + B;

	cout<<"Tong 2 phan so: ";
	cout<<C;

	if (A > B)
		cout<<"Phan so A lon hon phan so B";
	else
		cout<<"Phan so A nho hon hoac bang phan so B";

	getch();
}
```

4.3. Bài tập

Hãy thực hiện các bài tập dưới đây trong cùng 1 Solution với tên **<MSSV>_<HoTen>_BTTuan4** với mỗi bài là 1 Project với tên **Bai_<xx>**

Ví dụ:

- Tên Solution: 306111999_LeVanQuang_BTTuan3

- Tên các Project: Bai_01, Bai_02, Bai_03

Đề bài:

Bài 01: Cho lớp đối tượng Phân số, hãy thực hiện các yêu cầu sau:

a. Khai báo lớp đối tượng PhanSo

b. Viết các toán tử +, -, *, /

c. Viết các toán tử so sánh >, >=, <, <=, ==

d. Viết các toán tử nhập xuất >>, <<

Bài 02: Cho đơn thức f(x) = ax + b. Hãy thực hiện các yêu cầu sau:

a. Khai báo lớp đối tượng DonThuc

b. Viết các toán tử +, -

c. Viết các toán tử nhập xuất >>, <<

Bài 03: Cho số phức z = a + b*i với a là phần thực, b là phần ảo, i là đơn vị ảo và $i^2 = -1$. Hãy thực hiện các yêu cầu sau:

a. Khai báo lớp đối tượng SoPhuc

b. Viết các toán tử +, -, *, /

c. Viết các toán tử so sánh >, >=, <, <=, ==

d. Viết các toán tử nhập xuất >>, <<

BÀI 5: PHƯƠNG THỰC TẠO LẬP VÀ PHƯƠNG THỨC HỦY

5.1. Phương thức tạo lập (Constructor)

Việc khởi tạo đối tượng thông qua hàm (phương thức) dễ dẫn đến quên gọi hàm và không khởi tạo đối tượng cần dùng chẳng hạn như lớp PhanSo có phương thức NhapPhanSo(). Khi đó các thuộc tính của lớp đối tượng có thể nhận các giá trị rác hay các giá trị ngẫu nhiên. Vì vậy, ngôn ngữ hướng đối tượng cung cấp *phương thức tạo lập* (constructor) để tránh việc quên khởi tạo trước khi sử dụng. Lưu ý: một số thuật ngữ về constructor: tạo lập, khởi tạo.

* **Đặc điểm của phương thức tạo lập**:

+ Có tên trùng với tên lớp và không có giá trị trả về.

+ Được gọi khi đối tượng được tạo.

+ Có thể có nhiều phương thức tạo lập chồng nhau (overloading) tức là các phương thức này trùng tên và chỉ khác nhau ở danh sách tham số truyền vào.

* **Có 3 loại phương thức tạo lập**:

+ Phương thức tạo lập mặc định (default constructor): không có tham số đầu vào

+ Phương thức tạo lập sao chép (copy constructor): dùng để tạo ra một đối tượng từ một đối tượng có sẵn.

+ Phương thức tạo lập nhận tham số đầu vào: do người dùng tự định nghĩa với các tham số đầu vào khác nhau để khởi tạo các thuộc tính của đối tượng.

5.2. Phương thức hủy (Destructor)

Phương thức hủy (destructor) của một lớp đối tượng là một phương thức đặc biệt được tự động gọi thực hiện khi đối tượng bị hủy (không còn sử dụng nữa).

* **Đặc điểm của phương thức hủy**:

+ Được tự động gọi thực hiện khi đối tượng bị hủy.

+ Không có giá trị trả về và không có tham số.

+ Một lớp đối tượng có duy nhất một phương thức hủy.

+ Trong C++, phương thức hủy có tên trung với lớp và có dấu ~ đặt ở ngay phía trước.

5.3. Ví dụ

Sử dụng lại ví dụ về phân số. Hãy viết hàm tạo lập và hàm hủy cho lớp đối tượng PhanSo.

***Tập tin PhanSo.h**

```
#ifndef _PHANSO_
#define _PHANSO_

#include <iostream>
using namespace std;

class PhanSo
{
private:
	int m_iTuSo;
	int m_iMauSo;
public:
	//phuong thuc tao lap mac dinh
	PhanSo();

	//Phuong thuc tao lap nhan tham so dau vao
	PhanSo(int iTuSo, int iMauSo);

	//Phuong thuc tao lap sao chep
	PhanSo(const PhanSo &ps);

	//Phuong thuc tao lap sao chep
	PhanSo(PhanSo *ps);

	//Phuong thuc huy
	~PhanSo();

	void SetTuSo(int iTuSo);
	void SetMauSo(int iMauSo);

	int GetTuSo();
	int GetMauSo();

	// toan tu nhap
	friend istream& operator>>(istream& inDevice, PhanSo* ps);

	// toan tu xuat
	friend ostream& operator<<(ostream& outDevice, PhanSo* ps);
};

#endif
```

***Tập tin PhanSo.cpp**

```
#include "PhanSo.h"
#include <iostream>
using namespace std;
```

```
PhanSo::PhanSo()
{
	this->m_iTuSo = 0;
	this->m_iMauSo = 1;
}

PhanSo::PhanSo(int iTuSo, int iMauSo)
{
	this->m_iTuSo = iTuSo;
	this->m_iMauSo = iMauSo;
}

PhanSo::PhanSo(const PhanSo &ps)
{
	this->m_iTuSo = ps.m_iTuSo;
	this->m_iMauSo = ps.m_iMauSo;
}

PhanSo::PhanSo(PhanSo *ps)
{
	this->m_iTuSo = ps->m_iTuSo;
	this->m_iMauSo = ps->m_iMauSo;
}

PhanSo::~PhanSo()
{
	this->m_iTuSo = 0;
	this->m_iMauSo = 1;
	cout<<"\nDa huy doi tuong!";
}


void PhanSo::SetTuSo(int iTuSo)
{
	this->m_iTuSo = iTuSo;
}

void PhanSo::SetMauSo(int iMauSo)
{
	this->m_iMauSo = iMauSo;
}

int PhanSo::GetTuSo()
{
	return this->m_iTuSo;
}

int PhanSo::GetMauSo()
{
	return this->m_iMauSo;
}
```

```
istream& operator>>(istream& inDevice, PhanSo* ps)
{
	cout<<"Nhap tu so: ";
	inDevice>>ps->m_iTuSo;
	cout<<"Nhap mau so: ";
	inDevice>>ps->m_iMauSo;
	return inDevice;
}

ostream& operator<<(ostream& outDevice, PhanSo *ps)
{
	outDevice<<ps->m_iTuSo<<"/"<<ps->m_iMauSo;
	return outDevice;
}
```

***Tập tin ChuongTrinhChinh.cpp**

```
#include "PhanSo.h"
#include "conio.h"
#include <iostream>
using namespace std;

void main()
{
	PhanSo *A;
	PhanSo *B;
	PhanSo *C;

	cout<<"* Phuong thuc tao lap mac dinh: ";
	A = new PhanSo();
	cout<<A;
	cout<<"\nNhap phan so: \n";
	cin>>A;


	cout<<"\n\n* Phuong thuc tao lap nhan tham so: ";
	int iTuSo, iMauSo;
	iTuSo = A->GetTuSo();
	iMauSo = A->GetMauSo();
	B = new PhanSo(iTuSo,iMauSo);
	cout<<B;

	cout<<"\n\n* Phuong thuc tao lap sao chep: ";
	C = new PhanSo(B);
	cout<<C;


	cout<<"\n\n* Huy doi tuong: ";
	//huy doi tuong
	delete A;
	delete B;
	delete C;

	getch();
}
```

5.4. Bài tập

Hãy thực hiện các bài tập dưới đây trong cùng 1 Solution với tên **<MSSV>_<HoTen>_BTTuan5** với mỗi bài là 1 Project với tên **Bai_<xx>**

Ví dụ:

- Tên Solution: 306111999_LeVanQuang_BTTuan5
- Tên các Project: Bai_01, Bai_02, Bai_03

Đề bài:

Bài 01: Cho lớp đối tượng Phân số, hãy thực hiện các yêu cầu sau:

a. Khai báo lớp đối tượng PhanSo

b. Viết hàm tạo lập và hàm hủy

Bài 02: Cho đơn thức f(x) = ax + b. Hãy thực hiện các yêu cầu sau:

a. Khai báo lớp đối tượng DonThuc

b. Viết hàm tạo lập và hàm hủy

Bài 03: Cho số phức z = a + b*i với a là phần thực, b là phần ảo, i là đơn vị ảo và $i^2 = -1$. Hãy thực hiện các yêu cầu sau:

a. Khai báo lớp đối tượng SoPhuc

b. Viết hàm tạo lập và hàm hủy

BÀI 6: BÀI TẬP VỀ MẢNG ĐỘNG CÁC SỐ NGUYÊN

6.1. Đề bài

Viết lớp mảng động các số nguyên MyIntArray, gồm:

* Các thuộc tính:

    int *m_pArr;

    int m_iSize;

* Các phương thức:

    a. Các hàm tạo.

    b. Hàm hủy.

    c. Toán tử >>: nhập mảng

    d. Toán tử <<: xuất mảng

    e. Toán tử []: truy xuất và gán giá trị tại phần tử thứ idx của mảng.

    f. Hàm GetLength: Cho biết số phần tử trong mảng.

    g. Hàm FindFirst (n): Cho biết vị trí của phần tử đầu tiên trong mảng bằng số nguyên n.

    h. Hàm FindMax: tìm giá trị lớn nhất trong mảng.

    i. Hàm FindMin: tìm giá trị nhỏ nhất trong mảng

    j. Hàm Add: Thêm vào cuối mảng một phần tử.

    k. Hàm Insert: Thêm vào mảng một phần tử tại vị trí idx.

    l. Hàm Remove: Xóa phần tử thứ i trong mảng.

    m. Hàm RemoveAll: Xóa tất cả các phần tử trong mảng.

    n. Hàm Sort: Sắp xếp các phần tử trong mảng theo thứ tự tăng dần.

6.2. Hướng dẫn giải

***Tập tin MyIntArray.h**

```
#ifndef _MYINTARRAY_
#define _MYINTARRAY_

#include <iostream>
using namespace std;

class MyIntArray
{
private:
	int *m_pArr;
	int m_iSize;

public:
	// ham tao lap
	MyIntArray();
	MyIntArray(int iSize);
	MyIntArray(int *p, int iSize);
	MyIntArray(MyIntArray *A);

	// ham huy
	~MyIntArray();

	//toan tu nhap
	friend istream& operator>>(istream& inDevice, MyIntArray *A);
	//toan tu xuat
	friend ostream& operator<<(ostream& outDevice, MyIntArray
*A);

	//toan tu []: lay va gan gia tri thong qua chi so idx
	int& operator[](const int idx);

	//do dai chuoi
	int GetLength();

	//vi tri xuat hien dau tien
	int FindFirst(int n);

	//gia tri lon nhat
	int FindMax();

	//gia tri nho nhat
	int FindMin();

	//them cuoi mang
	bool Add(int n);

	//chen vao mang
	bool Insert(int n, int idx);

	//xoa 1 phan tu
	bool Remove(int idx);
```

```
	//xoa tat ca phan tu
	bool RemoveAll();

	//sap xep tang
	void Sort();

};

#endif
```

***Tập tin MyIntArray.cpp**

```
#include "MyIntArray.h"
#include <iostream>
using namespace std;

MyIntArray::MyIntArray()
{
	this->m_iSize = 0;
	this->m_pArr = NULL;
}

MyIntArray::MyIntArray(int iSize)
{
	this->m_iSize = iSize;

	//cap phat bo nho
	this->m_pArr = new int[iSize];
	for (int i = 0; i < this->m_iSize; i++)
	{
		this->m_pArr[i] = 0;
	}
}

MyIntArray::MyIntArray(int *p, int iSize)
{
	this->m_iSize = iSize;
	this->m_pArr = new int[iSize];

	//sao chep cac phan tu
	for (int i = 0; i < this->m_iSize; i++)
	{
		this->m_pArr[i] = p[i];
	}

}

MyIntArray::MyIntArray(MyIntArray *A)
{
	this->m_iSize = A->m_iSize;
	this->m_pArr = new int[this->m_iSize];

	//sao chep cac phan tu
```

```
	for (int i = 0; i < this->m_iSize; i++)
	{
		this->m_pArr[i] = A->m_pArr[i];
	}
}


// ham huy
MyIntArray::~MyIntArray()
{
	if (this->m_iSize > 0)
	{
		this->m_iSize = 0;
		delete []this->m_pArr;
		this->m_pArr = NULL;
	}
}

istream& operator>>(istream& inDevice, MyIntArray *A)
{
	cout<<"Nhap so phan tu: ";
	inDevice>>A->m_iSize;
	A->m_pArr = new int[A->m_iSize];
	for (int i = 0; i < A->m_iSize; i++)
	{
		cout<<"Nhap phan tu "<<i<<": ";
		inDevice>>A->m_pArr[i];
	}

	return inDevice;
}

ostream& operator<<(ostream& outDevice, MyIntArray *A)
{
	outDevice<<"\nSo phan tu: "<<A->m_iSize<<endl;
	for (int i = 0; i < A->m_iSize; i++)
	{
		outDevice<<A->m_pArr[i]<<" ";
	}
	return outDevice;
}


int& MyIntArray::operator[](const int idx)
{
	return this->m_pArr[idx];
}

int MyIntArray::GetLength()
{
	return this->m_iSize;
}
```

```
int MyIntArray::FindFirst(int n)
{
	int kq;
	kq = -1;

	for (int i = 0; i < this->m_iSize; i++)
	{
		if (this->m_pArr[i] == n)
		{
			kq = i;
			break;
		}
	}

	return kq;
}


int MyIntArray::FindMax()
{
	int max;
	max = this->m_pArr[0];

	for (int i = 0; i < this->m_iSize; i++)
	{
		if (this->m_pArr[i] > max)
		{
			max = this->m_pArr[i];
		}
	}
	return max;
}

int MyIntArray::FindMin()
{
	int min;
	min = this->m_pArr[0];

	for (int i = 0; i < this->m_iSize; i++)
	{
		if (this->m_pArr[i] < min)
		{
			min = this->m_pArr[i];
		}
	}
	return min;
}

//Them cuoi mang
bool MyIntArray::Add(int n)
{
	bool kq;

	//cap phat mang p nhieu hon mang m_pArr 1 phan tu
```

```
	int *p = new int[this->m_iSize+1];

	kq = false;

	if (p != NULL)
	{
		//sao chep tu mang m_pArr sang mang p
		for (int i = 0; i < this->m_iSize; i++)
		{
			p[i] = this->m_pArr[i];
		}

		//gan gia tri n vao phan tu cuoi cua mang p
		p[this->m_iSize] = n;

		//xoa mang m_pArr
		delete []this->m_pArr;
		this->m_pArr = NULL;

	//khoi tao lai mang m_pArr moi nhieu hon mang cu 1 phan tu
		this->m_iSize++;
		this->m_pArr = new int[this->m_iSize];

		if (this->m_pArr != NULL)
		{
			//sao chep tu mang p sang mang m_pArr
			for (int i = 0; i < this->m_iSize; i++)
			{
				this->m_pArr[i] = p[i];
			}
			kq = true;
		}
	}

	return kq;
}

bool MyIntArray::Insert(int n, int idx)
{
	bool kq;

	//cap phat mang p nhieu hon mang m_pArr 1 phan tu
	int *p = new int[this->m_iSize+1];

	kq = false;

	if (p != NULL && idx <= this->m_iSize)
	{
	  //sao chep idx phan tu dau tien tu mang m_pArr sang mang p
		for (int i = 0; i < idx; i++)
		{
			p[i] = this->m_pArr[i];
		}
```

```
            //chen mang p tai vi tri idx gia tri n
            p[idx] = n;

          //sao chep lai du lieu tu mang m_pArr bat dau tu vi tri
            //idx vao mang p
            for (int i = idx+1; i < this->m_iSize+1; i++)
            {
                  p[i] = this->m_pArr[i-1];
            }

            //xoa mang m_pArr
            delete []this->m_pArr;
            this->m_pArr = NULL;

          //khoi tao lai mang m_pArr moi nhieu hon mang cu 1 phan tu
            this->m_iSize++;
            this->m_pArr = new int[this->m_iSize];

            if (this->m_pArr != NULL)
            {
                  //sao chep tu mang p sang mang m_pArr
                  for (int i = 0; i < this->m_iSize; i++)
                  {
                        this->m_pArr[i] = p[i];
                  }
                  kq = true;
            }
      }

      return kq;
}

bool MyIntArray::Remove(int idx)
{
      bool kq;

      //cap phat mang p it hon mang m_pArr 1 phan tu
      int *p = new int[this->m_iSize-1];

      kq = false;

      if (p != NULL && idx <= this->m_iSize)
      {
        //sao chep idx phan tu dau tien tu mang m_pArr sang mang p
            for (int i = 0; i < idx; i++)
            {
                  p[i] = this->m_pArr[i];
            }

            // don mang bat dau tai vi tri thu idx
            for (int i = idx; i < this->m_iSize-1; i++)
            {
                  p[i] = this->m_pArr[i+1];
            }
```

```
			//xoa mang m_pArr
			delete []this->m_pArr;
			this->m_pArr = NULL;

			//khoi tao lai mang m_pArr moi it hon mang cu 1 phan tu
			this->m_iSize--;
			this->m_pArr = new int[this->m_iSize];

			if (this->m_pArr != NULL)
			{
				//sao chep tu mang p sang mang m_pArr
				for (int i = 0; i < this->m_iSize; i++)
				{
					this->m_pArr[i] = p[i];
				}
				kq = true;
			}
		}

		return kq;
}

bool MyIntArray::RemoveAll()
{
		bool kq;
		kq = false;
		if (this->m_iSize > 0)
		{
			this->m_iSize = 0;
			delete []this->m_pArr;
			this->m_pArr = NULL;
			kq = true;
		}

		return kq;
}

void MyIntArray::Sort()
{
		int iSize;
		iSize = this->m_iSize;

		for (int i = 0; i < iSize-1; i++)
		{
			for (int j = i + 1; j < iSize; j++)
			{
				if (this->m_pArr[i] > this->m_pArr[j])
				{
					int temp;
					temp = this->m_pArr[i];
					this->m_pArr[i] = this->m_pArr[j];
					this->m_pArr[j] = temp;
				}
```

```
            }
      }
}
```

***Tập tin ChuongTrinhChinh.cpp**

```
#include "MyIntArray.h"
#include "conio.h"
#include <iostream>
using namespace std;

void main()
{
      MyIntArray *a1 = new MyIntArray();
      cin>>a1;
      cout<<a1;

      /*
      MyIntArray *a2 = new MyIntArray(3);
      cout<<a2;

      MyIntArray *a3 = new MyIntArray(a1);
      cout<<a3;
      */
      cout<<"\n\n*Truy xuat phan tu thu idx (toan tu []):";
      int idx;
      cout<<"\nNhap idx: ";
      cin>>idx;
      int kq = (*a1)[idx];
      cout<<"Phan tu thu "<<idx<<": "<<kq<<endl;
      (*a1)[idx] = 5; //gan phan tu thu idx co gia tri la 5

      cout<<"\n\n*Vi tri xuat hien dau tien (FindFist):";
      int pos;
      int n;
      cout<<"\nNhap n: ";
      cin>>n;
      pos = a1->FindFirst(n);
      cout<<"Vi tri dau tien: "<<pos;

      cout<<"\n\n*Gia tri lon nhat (FindMax):";
      int max;
      max = a1->FindMax();
      cout<<"\nGTLN: "<<max;

      cout<<"\n\n*Gia tri nho nhat (FindMin):";
      int min;
      min = a1->FindMin();
      cout<<"\nGTNN: "<<min;

      cout<<"\n\n*Them vao cuoi mang (Add):";
      cout<<"Nhap n: ";
      cin>>n;
      a1->Add(n);
```

```
        cout<<a1;

        cout<<"\n\n*Them vao mang 1 phan tu tai vi tri idx(Insert):";
        cout<<"Nhap n: ";
        cin>>n;
        cout<<"Nhap idx: ";
        cin>>idx;
        a1->Insert(n,idx);
        cout<<a1;

        cout<<"\n\n*Xoa phan tu thu idx (Remove):";
        cout<<"Nhap idx: ";
        cin>>idx;
        a1->Remove(idx);
        cout<<a1;

        cout<<"\n\n*Sap xep tang dan (Sort):";
        a1->Sort();
        cout<<a1;

        cout<<"\n\n*Xoa tat ca phan tu (RemoveAll):";
        a1->RemoveAll();
        cout<<a1;

        //huy doi tuong
        delete a1;
        a1 = NULL;

        getch();
}
```

BÀI 7: BÀI TẬP VỀ MẢNG ĐỘNG CÁC PHÂN SỐ

7.1. Đề bài

Viết lớp mảng động các Phân số MyPhanSoArray, gồm:

* Các thuộc tính:

```
PhanSo  **m_pArr;
int   m_iSize;
```

* Các phương thức:

a. Các hàm tạo.

b. Hàm hủy.

c. Toán tử >>: nhập mảng phân số

d. Toán tử <<: xuất mảng phân số

e. Toán tử []: truy xuất và gán giá trị tại phần tử thứ idx của mảng phân số.

f. Hàm GetLength: Cho biết số phần tử trong mảng phân số.

g. Hàm FindFirst: Cho biết vị trí của phần tử đầu tiên trong mảng bằng phân số ps.

h. Hàm FindMax: tìm phân số lớn nhất trong mảng.

i. Hàm FindMin: tìm phân số nhỏ nhất trong mảng

j. Hàm Add: Thêm vào cuối mảng một phân số.

k. Hàm Insert: Thêm vào mảng một phân số tại vị trí idx.

l. Hàm Remove: Xóa phần tử thứ i trong mảng.

m. Hàm RemoveAll: Xóa tất cả các phần tử trong mảng.

n. Hàm Sort: Sắp xếp các phần tử phân số trong mảng theo thứ tự tăng dần.

7.2. Hướng dẫn giải

1. Phần Khai báo:

*** Tập tin PhanSo.h**

```
#ifndef _PHANSO_
#define _PHANSO_

#include <iostream>
using namespace std;

class PhanSo
{
private:
	int m_iTuSo;
	int m_iMauSo;
public:
	PhanSo();
	PhanSo(int iTuSo, int iMauSo);
	PhanSo(PhanSo *ps);
	~PhanSo();

	void SetTuSo(int iTuSo);
	void SetMauSo(int iMauSo);

	int GetTuSo();
	int GetMauSo();

	//tinh toan
	PhanSo* operator+(PhanSo ps);
	PhanSo* operator-(PhanSo ps);
	PhanSo* operator*(PhanSo ps);
	PhanSo* operator/(PhanSo ps);

	//so sanh
	bool operator>(PhanSo ps);
	bool operator<(PhanSo ps);
	bool operator>=(PhanSo ps);
	bool operator<=(PhanSo ps);
	bool operator==(PhanSo ps);
	bool operator!=(PhanSo ps);

	//toan tu nhap
	friend istream& operator>>(istream& inDevice, PhanSo *ps);
	//toan tu xuat
	friend ostream& operator<<(ostream& outDevice, PhanSo *ps);
};

#endif
```

*** Tập tin MyPhanSoArray.h**

```
#ifndef _MYPHANSOARRAY_
#define _MYPHANSOARRAY_

#include "PhanSo.h"

class MyPhanSoArray
{
private:
	PhanSo	**m_pArr;
	int		m_iSize;
public:
		// ham tao lap
	MyPhanSoArray();
	MyPhanSoArray(int iSize);
	MyPhanSoArray(PhanSo **p, int iSize);
	MyPhanSoArray(MyPhanSoArray *A);
	// ham huy
	~MyPhanSoArray();

	void SetLength(int iSize);
	void SetArray(PhanSo **p, int iSize);


	//toan tu nhap
	friend istream& operator>>(istream& inDevice, MyPhanSoArray
*A);
	//toan tu xuat
	friend ostream& operator<<(ostream& outDevice, MyPhanSoArray
*A);

	//toan tu []: lay va gan gia tri thong qua chi so idx
	PhanSo* operator[](const int idx);

	//do dai chuoi
	int GetLength();
	PhanSo** GetArray();

	//gia tri lon nhat
	PhanSo* FindMax();

	//gia tri nho nhat
	PhanSo* FindMin();

	//them cuoi mang
	bool Add(PhanSo* ps);

	//chen vao mang
	bool Insert(PhanSo* ps, int idx);

	//xoa 1 phan tu
	bool Remove(int idx);
```

```
	//xoa tat ca phan tu
	bool RemoveAll();

	//sap xep tang
	void Sort();
};

#endif
```

2. Phần Cài đặt cụ thể:

*** Tập tin PhanSo.cpp**

```
#include "PhanSo.h"
#include <iostream>
using namespace std;

PhanSo::PhanSo()
{
	this->m_iTuSo = 0;
	this->m_iMauSo = 1;
}
PhanSo::PhanSo(int iTuSo, int iMauSo)
{
	this->m_iTuSo = iTuSo;
	this->m_iMauSo = iMauSo;
}
PhanSo::PhanSo(PhanSo *ps)
{
	this->m_iTuSo = ps->m_iTuSo;
	this->m_iMauSo = ps->m_iMauSo;
}
PhanSo::~PhanSo()
{
	cout<<"\nDa huy doi tuong phan so!";
	this->m_iTuSo = 0;
	this->m_iMauSo = 1;
}

void PhanSo::SetTuSo(int iTuSo)
{
	this->m_iTuSo = iTuSo;
}

void PhanSo::SetMauSo(int iMauSo)
{
	this->m_iMauSo = iMauSo;
}

int PhanSo::GetTuSo()
{
	return this->m_iTuSo;
}
```

```
int PhanSo::GetMauSo()
{
	return this->m_iMauSo;
}
PhanSo* PhanSo::operator+(PhanSo ps)
{
	PhanSo* kq = new PhanSo();

	kq->m_iTuSo = m_iTuSo*ps.m_iMauSo + ps.m_iTuSo*m_iMauSo;
	kq->m_iMauSo = m_iMauSo * ps.m_iMauSo;

	return kq;
}
PhanSo* PhanSo::operator-(PhanSo ps)
{
	PhanSo *kq = new PhanSo();

	kq->m_iTuSo = m_iTuSo*ps.m_iMauSo - ps.m_iTuSo*m_iMauSo;
	kq->m_iMauSo = m_iMauSo * ps.m_iMauSo;

	return kq;
}
bool PhanSo::operator>(PhanSo ps)
{
	bool kq;
	if (this->m_iTuSo*ps.m_iMauSo > ps.m_iTuSo*this->m_iMauSo)
		kq = true;
	else
		kq = false;

	return kq;
}

istream& operator>>(istream& inDevice, PhanSo *ps)
{
	cout<<"Nhap tu so: ";
	inDevice>>ps->m_iTuSo;
	cout<<"Nhap mau so: ";
	inDevice>>ps->m_iMauSo;
	return inDevice;
}


ostream& operator<<(ostream& outDevice, PhanSo *ps)
{
	outDevice<<ps->m_iTuSo<<"/"<<ps->m_iMauSo;
	return outDevice;
}
```

*** Tập tin MyPhanSoArray.cpp**

```
#include "MyPhanSoArray.h"
#include "PhanSo.h"
```

```
#include <iostream>
using namespace std;
MyPhanSoArray::MyPhanSoArray()
{
	this->m_iSize = 0;
	this->m_pArr = NULL;
}
MyPhanSoArray::MyPhanSoArray(int iSize)
{
	this->m_iSize = iSize;
	this->m_pArr = new PhanSo*[iSize];
	for (int i = 0; i < iSize; i++)
	{
		this->m_pArr[i] = new PhanSo();
	}
}
MyPhanSoArray::MyPhanSoArray(PhanSo **p, int iSize)
{
	this->m_iSize = iSize;
	this->m_pArr = new PhanSo*[iSize];
	for (int i = 0; i < iSize; i++)
	{
		this->m_pArr[i] = new PhanSo(p[i]);
	}
}
MyPhanSoArray::MyPhanSoArray(MyPhanSoArray *A)
{
	this->m_iSize = A->m_iSize;

	this->m_pArr = new PhanSo*[this->m_iSize];
	for (int i = 0; i < this->m_iSize; i++)
	{
		this->m_pArr[i] = new PhanSo(A->m_pArr[i]);
	}
}
// ham huy
MyPhanSoArray::~MyPhanSoArray()
{
	cout<<"\n\nDa huy doi tuong mang!!"<<endl;
	if (this->m_iSize > 0)
	{
		for (int i = 0; i < this->m_iSize; i++)
		{
			delete this->m_pArr[i];
			this->m_pArr[i] = NULL;
		}

		this->m_iSize = 0;
		delete this->m_pArr;
		this->m_pArr = NULL;
	}
}

//toan tu nhap
```

```
istream& operator>>(istream& inDevice, MyPhanSoArray *A)
{
	cout<<"Nhap so phan tu: ";
	inDevice>>A->m_iSize;

	A->m_pArr = new PhanSo*[A->m_iSize];
	for (int i = 0; i < A->m_iSize; i++)
	{
		A->m_pArr[i] = new PhanSo();
		cout<<"\nNhap phan tu thu "<<i<<": "<<endl;
		inDevice>>A->m_pArr[i];
	}
	return inDevice;
}
	//toan tu xuat
ostream& operator<<(ostream& outDevice, MyPhanSoArray *A)
{
	outDevice<<"\nMang phan so co so phan tu la: "<<A-
>m_iSize<<endl;
	for (int i = 0; i < A->m_iSize; i++)
	{
		outDevice<<A->m_pArr[i]<<endl;
	}
	return outDevice;
}

PhanSo* MyPhanSoArray::operator[](const int idx)
{
	return this->m_pArr[idx];
}

void MyPhanSoArray::SetLength(int iSize)
{
	this->m_iSize = iSize;
	this->m_pArr = new PhanSo*[iSize];
	for (int i = 0; i < iSize; i++)
	{
		this->m_pArr[i] = new PhanSo();
	}
}
void MyPhanSoArray::SetArray(PhanSo **p, int iSize)
{
	this->m_iSize = iSize;
	this->m_pArr = new PhanSo*[iSize];
	for (int i = 0; i < iSize; i++)
	{
		this->m_pArr[i] = new PhanSo(p[i]);
	}
}
PhanSo** MyPhanSoArray::GetArray()
{
	return this->m_pArr;
}
```

```
int MyPhanSoArray::GetLength()
{
	return this->m_iSize;
}

PhanSo* MyPhanSoArray::FindMax()
{
	PhanSo *max = new PhanSo();

	max = this->m_pArr[0];
	for (int i = 0; i < this->m_iSize; i++)
	{
		//toan tu so sanh > cua lop PhanSo
		if (*this->m_pArr[i] > *max)
		{
			max = this->m_pArr[i];
		}
	}

	return max;
}
```

*** Tập tin ChuongTrinhChinh.cpp**

```
#include "MyPhanSoArray.h"
#include "conio.h"
#include <iostream>
using namespace std;

void main()
{

	MyPhanSoArray *arrPhanSo = new MyPhanSoArray();
	cin>>arrPhanSo;
	cout<<arrPhanSo;

	PhanSo *max = arrPhanSo->FindMax();
	cout<<"\nPhan so lon nhat: ";
	cout<<max;


	delete arrPhanSo;
	getch();
}
```

BÀI 8: LỚP KẾ THỪA

8.1. Giới thiệu về kế thừa

8.1.1. Khái niệm

Kế thừa (inheritance) trong hướng đối tượng là cách tạo lớp mới từ các lớp đã được định nghĩa trước đó. Trong đó, lớp mới gọi là *lớp kế thừa*, *lớp con* hay *lớp dẫn xuất* (*derive class*), còn các lớp đã được định nghĩa trước đó gọi là *lớp cơ sở* hay *lớp cha* (*base class*).

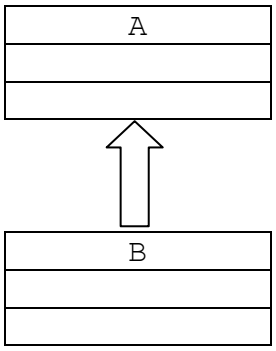
Lớp B kế thừa từ lớp A, tức là lớp B là *đặc biệt hóa* của lớp A (hay nói cách khác: lớp A là *tổng quát hóa* của lớp B). Chúng ta có một số tính chất liên quan đến tính kế thừa:

+ Lớp kế thừa (lớp B) có thể sử dụng lại các thuộc tính và phương thức của lớp cơ sở (lớp A).

+ Lớp kế thừa (lớp B) có thể định nghĩa thêm các thuộc tính và phương thức mới cho riêng nó.

+ Lớp kế thừa (lớp B) có thể chỉnh sửa lại phương thức (có sẵn trong lớp A) cho phù hợp với đặc trưng của nó. Phương thức được định nghĩa gọi là phương thức nạp chồng (*method overriding*).

Hình minh họa bằng UML:



Kế thừa trong C++ được khai báo theo cấu trúc sau:

```
class <TênLớpKếThừa> : <PhạmVi> <TênLớpCơSở>
{
private:
      <Khai báo các thành phần riêng tư>
protected:
      <Khai báo cáo thành phần bảo tồn>
public:
      <Khai báo các thành phần công cộng>
};
```

8.1.2. Phạm vi (scope)

	Bên trong lớp	Lớp kế thừa	Bên ngoài lớp
`private`	X		
`protected`	X	X	
`public`	X	X	X

Lớp kế thừa (Lớp B) không thể truy xuất các thuộc tính và phương thức của lớp cơ sở (lớp A) nếu phạm vi là `private`.

Trong thực tế, phạm vi của lớp B kế thừa từ lớp A thông thường là public.

8.1.3. Phương thức tạo lập (construction) trong kế thừa

Khi một đối tượng thuộc lớp kế thừa được khởi tạo, phương thức tạo lập của lớp cơ sở (lớp A) phải được gọi thực hiện trước, sau đó đó mới đến phương thức tạo lập của lớp kế thừa (lớp B).

Trong phương thức tạo lập của lớp kế thừa (lớp B) chúng ta cần chỉ ra sẽ gọi phương thức nào của lớp cơ sở (lớp A) một cách tường minh. Nếu không chỉ ra, phương thức tạo lập mặc định của lớp cơ sở (lớp A) sẽ được gọi.

8.1.4. Phương thức hủy (destruction) trong kế thừa

Khi một đối tượng thuộc lớp kế thừa bị hủy, hàm hủy của lớp kế thừa được gọi thực hiện trước, Sau đó mới đến hàm hủy của lớp cơ sở.

8.1.5. Ví dụ

Ví dụ 1: Cho 2 lớp đối tượng HinhChuNhat và HinhVuong. Hãy thực hiện các yêu cầu sau:

a. Khai báo 2 lớp đối tượng này.

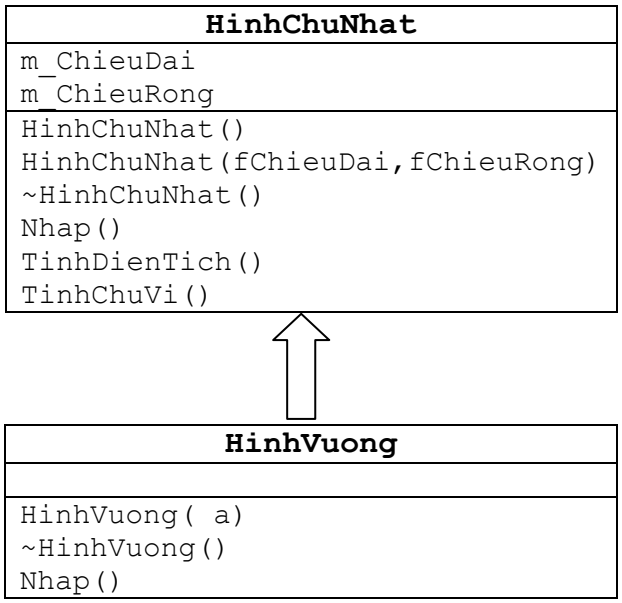
b. Viết các phương thức tạo lập

c. Viết các phương thức hủy.

d. Viết phương thức tính diện tích

e. Viết phương thức tính chu vi.

Hình minh họa bằng UML:



Phần 1: Khai báo

***Tập tin HinhChuNhat.h**

```
#ifndef _HINHCHUNHAT_
#define _HINHCHUNHAT_

class HinhChuNhat
{
protected:
	float m_fChieuDai;
	float m_fChieuRong;
public:
	HinhChuNhat();
	HinhChuNhat(float fChieuDai, float fChieuRong);
	~HinhChuNhat();
	void Nhap();
	float TinhDienTich();
	float TinhChuVi();
};
#endif
```

***Tập tin HinhVuong.h**

```
#ifndef _HINHVUONG_
#define _HINHVUONG_

#include "HinhChuNhat.h"

class HinhVuong : public HinhChuNhat
{
public:
	HinhVuong(float a=0);
	~HinhVuong();
	void Nhap();
};
```

```
#endif
```

Phần 2: Cài đặt cụ thể

*** Tập tin HinhChuNhat.cpp**

```
#include "HinhChuNhat.h"
#include <iostream>
using namespace std;

HinhChuNhat::HinhChuNhat()
{
	this->m_fChieuDai = 0;
	this->m_fChieuRong = 0;
}
HinhChuNhat::HinhChuNhat(float fChieuDai, float fChieuRong)
{
	this->m_fChieuDai  = fChieuDai;
	this->m_fChieuRong = fChieuRong;
}
HinhChuNhat::~HinhChuNhat()
{
	this->m_fChieuDai = 0;
	this->m_fChieuRong = 0;
	cout<<"Da huy doi tuong hinh chu nhat!"<<endl;
}

void HinhChuNhat::Nhap()
{
	cout<<"Nhap chieu dai: ";
	cin>>this->m_fChieuDai;
	cout<<"Nhap chieu rong: ";
	cin>>this->m_fChieuRong;
}
float HinhChuNhat::TinhDienTich()
{
	float dt;
	dt = this->m_fChieuDai * this->m_fChieuRong;
	return dt;
}
float HinhChuNhat::TinhChuVi()
{
	float cv;
	cv = (this->m_fChieuDai + this->m_fChieuRong)*2;
	return cv;
}
```

*** Tập tin HinhVuong.cpp**

```
#include "HinhVuong.h"
#include <iostream>
using namespace std;
```

```
HinhVuong::HinhVuong(float a)
{
	this->m_fChieuDai = a;
	this->m_fChieuRong = a;
}
HinhVuong::~HinhVuong()
{
	cout<<"Da huy doi tuong hinh vuong!!"<<endl;
}
void HinhVuong::Nhap()
{
	float canh;
	cout<<"Nhap canh hinh vuong: ";
	cin>>canh;
	this->m_fChieuDai = canh;
	this->m_fChieuRong = canh;
}
```

*** Tập tin ChuongTrinhChinh.cpp**

```
#include "HinhVuong.h"
#include "HinhChuNhat.h"
#include "conio.h"
#include <iostream>
using namespace std;

void main()
{
	HinhChuNhat *hcn;
	HinhVuong *hv;

	cout<<"** Hinh chu nhat: "<<endl;
	hcn = new HinhChuNhat();
	hcn->Nhap();
	cout<<"Dien tich: "<<hcn->TinhDienTich();
	cout<<"\nChu vi: "<<hcn->TinhChuVi();

	cout<<"\n\n** Hinh vuong: "<<endl;
	hv= new HinhVuong();
	hv->Nhap();
	cout<<"Dien tich: "<<hv->TinhDienTich();
	cout<<"\nChu vi: "<<hv->TinhChuVi();

	cout<<"\n\n** Ket thuc"<<endl;
	delete hcn;
	delete hv;
	getch();

}
```

8.2. Các loại quan hệ giữa các lớp đối tượng

Trong lập trình hướng đối tượng, chúng ta thường sử dụng 2 loại quan hệ cơ bản sau giữa các lớp đối tượng:

+ Quan hệ HAS-A hay PART-OF

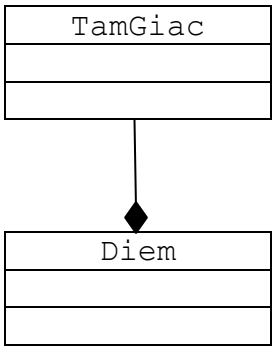
+ Quan hệ IS-A

8.2.1. Quan hệ HAS-A, PART-OF

Hai lớp đối tượng được gọi là có quan hệ HAS-A, PART-OF khi thuộc tính của lớp đối tượng này có chứa đối tượng của lớp kia. Quan hệ HAS-A, PART-OF còn gọi là quan hệ bao hàm hay bộ phận.

Ví dụ 2: Thiết kế lớp Tam giác với mỗi tam giác có 3 Đỉnh. Khi đó, lớp **Tam Giác** chứa 3 **Điểm** (tức là 3 đỉnh).

Hình minh họa bằng UML:



Phần 1: Khai báo

*** Tập tin Diem.h**

```
#ifndef _DIEM_
#define _DIEM_

class Diem
{
private:
	float m_x;
	float m_y;
public:
	float GetX();
	float GetY();
	void SetX(float x);
	void SetY(float y);
	void NhapDiem();
};

#endif
```

*** Tập tin TamGiac.h**

```
#ifndef _TAMGIAC_
#define _TAMGIAC_

#include "Diem.h"

class TamGiac
{
private:
	Diem m_arrDiem[3]; //tam giac co 3 dinh
public:
	void NhapTamGiac();
	float TinhDienTich();
	float TinhChuVi();
};

#endif
```

Phần 2: Cài đặt cụ thể

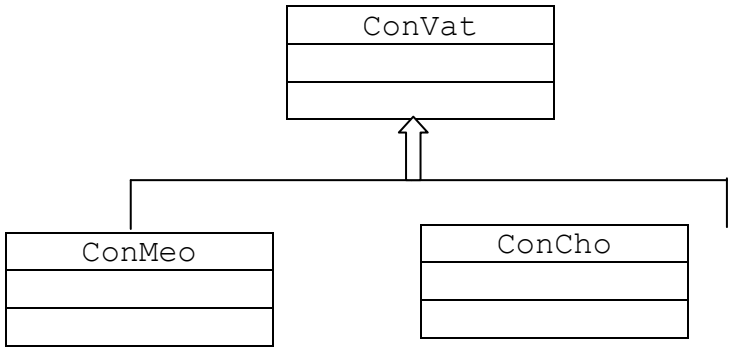
SV tự làm, xem như bài tập, có thể bổ sung thêm một số phương thức nếu cần thiết.

8.2.2. Quan hệ IS-A

Hai lớp đối tượng được gọi là có quan hệ IS-A khi một trường hợp lớp đối tượng này kế thừa từ lớp đối tượng kia. Hay nói cách khác: lớp đối tượng này là đặc biệt hóa (tổng quát hóa) của lớp kia.

Ví dụ 3: Lớp Con Vật là trường hợp tổng quát hóa của lớp Con Mèo và lớp Con Chó. Khi đó, Con Mèo là một Con Vật hay Con Chó là một Con Vật (quan hệ IS-A)

Hình minh họa bằng UML:



Phần 1: Khai báo

*** Tập tin ConVat.h**

```
#ifndef _CONVAT_
#define _CONVAT_

class ConVat
{
public:
	void Keu();
};
```

```
#endif
```

*** Tập tin ConMeo.h**

```
#ifndef _CONMEO_
#define _CONMEO_

#include "ConVat.h"

class ConMeo:public ConVat
{
public:
	void Keu();
};

#endif
```

*** Tập tin ConCho.h**

```
#ifndef _CONCHO_
#define _CONCHO_

#include "ConVat.h"

class ConCho:public ConVat
{
public:
	void Keu();
};

#endif
```

Phần 2: Cài đặt cụ thể

*** Tập tin ConVat.cpp**

```
#include "ConVat.h"
#include <iostream>
using namespace std;

void ConVat::Keu()
{

}
```

*** Tập tin ConMeo. cpp**

```
#include "ConMeo.h"
#include <iostream>
using namespace std;

void ConMeo::Keu()
{
	cout<<"Meo meo!";
```

```
}
```

*** Tập tin ConCho. cpp**

```
#include "ConCho.h"
#include <iostream>
using namespace std;

void ConCho::Keu()
{
	cout<<"Gau gau!";
}
```

*** Tập tin ChuongTrinhChinh.cpp**

```
#include "ConVat.h"
#include "ConMeo.h"
#include "ConCho.h"
#include "conio.h"
#include <iostream>
using namespace std;

void main()
{
	ConMeo *m = new ConMeo();
	ConCho *c = new ConCho();

	m->Keu();
	c->Keu();

	getch();

}
```

8.3. Bài tập

Hãy thực hiện các bài tập dưới đây trong cùng 1 Solution với tên **<MSSV>_<HoTen>_BTTuan8** với mỗi bài là 1 Project với tên **Bai_<xx>**

Ví dụ:

- Tên Solution: 306111999_LeVanQuang_BTTuan8

- Tên các Project: Bai_01, Bai_02, Bai_03

Đề bài:

Bài 01: Cho Hình Chữ Nhật và Hình Vuông (xem lại Ví dụ 1). Hãy thực hiện các yêu cầu sau:

a. Khai báo lớp HinhChuNhat

b. Khai báo lớp HinhVuong (kế thừa từ HinhChuNhat)

c. Trong lớp HinhChuNhat và HinhVuong, viết các phương thức sau:

c1. Phương thức khởi tạo

c2. Phương thức hủy

c3. Toán tử nhập: >>

c4. Toán tử xuất: <<

c5. Tính diện tích

c6. Tính chu vi

Bài 02: Cho hình Ellipse và Hình tròn. Hãy thực hiện các yêu cầu sau:

a. Khai báo lớp Ellipse

b. Khai báo lớp HinhTron (kế thừa từ lớp Ellipse)

c. Trong lớp Ellipse và HinhTron, viết các phương thức sau:

c1. Phương thức khởi tạo

c2. Phương thức hủy

c3. Toán tử nhập: >>

c4. Toán tử xuất: <<

c5. Tính diện tích

c6. Tính chu vi

Bài 03: Cho toạ độ 3 đỉnh (Điểm) của một tam giác trong mặt phẳng Oxy (xem lại Ví dụ 2).

Hãy thực hiện các yêu cầu sau:

a. Khai báo lớp Diem

b. Khai báo lớp TamGiac

c. Trong lớp Diem, viết các phương thức sau:

c1. Phương thức khởi tạo

c2. Phương thức hủy

c3. Toán tử nhập: >>

c4. Toán tử xuất: <<

d. Trong lớp TamGiac, viết các phương thức sau:

d1. Phương thức khởi tạo

d2. Phương thức hủy

d3. Toán tử nhập: >>

d4. Toán tử xuất: <<

d5. Tính khoảng cách giữa 2 điểm

d6. Tính diện tích tam giác

d7. Tính chu vi tam giác

d8. Tìm trọng tâm tam giác

BÀI 9: ĐA HÌNH

9.1. Giới thiệu về đa hình

Đa hình (hay Đa xạ - Polymorphism) là kỹ thuật trong hướng đối tượng tạo điều kiện cho các lập trình viên gia tăng khả năng tái sử dụng những đoạn mã nguồn được viết một cách tổng quát và có thể thay đổi cách ứng xử một cách linh hoạt tùy theo từng loại đống tượng.

9.2. Ví dụ

Ví dụ 1: Cho 5 lớp đối tượng Hinh, HinhChuNhat, HinhVuong, Ellipse, HinhTron. Hãy thực hiện các yêu cầu sau:

a. Khai báo 5 lớp đối tượng này thể hiện mối quan hệ kế thừa.

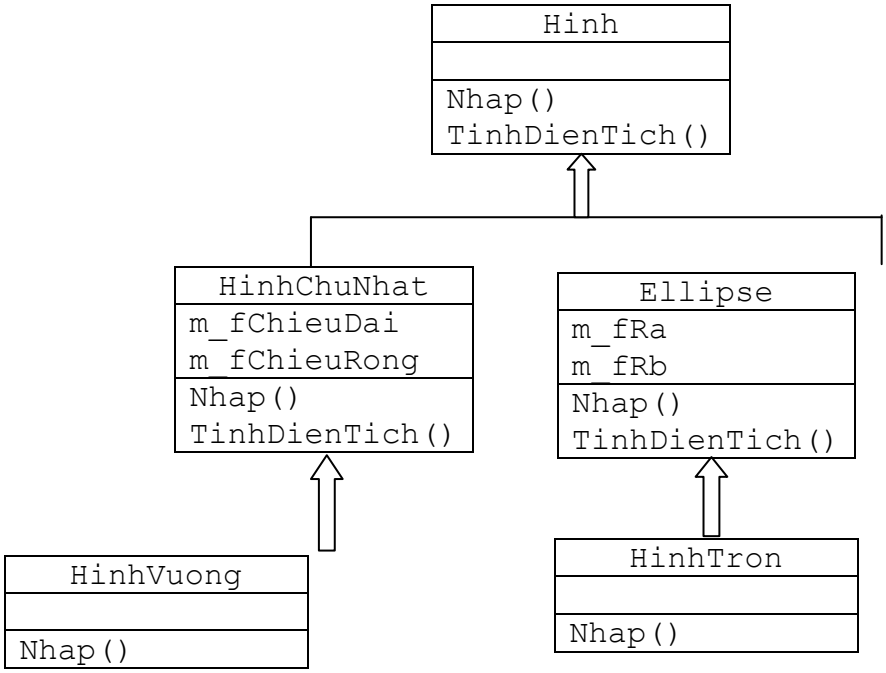
b. Viết các phương thức tạo lập

c. Viết các phương thức hủy.

d. Viết phương thức tính diện tích

e. Viết tìm diện tích lớn nhất trong mảng các Hinh.

Hình minh họa bằng UML:



Hai lớp HinhChuNhat và Ellipse là đặc biệt hóa của lớp Hinh.

Lớp HinhVuong là đặc biệt hóa của lớp HinhChuNhat

Lớp HinhTron là đặc biệt hóa của Ellipse

Lớp Hinh là lớp tổng quát và là lớp trừu tượng (abstract class)

Phương thức Nhap( ), TinhDienTich( ) trong lớp Hinh, HinhChuNhat, Ellipse là 2 phương thức ảo, cần thêm từ khóa virtual trong khai báo.

***Tập tin Hinh.h**

```
#ifndef _HINH_
#define _HINH_

class Hinh
{
public:
	Hinh();
	~Hinh();
	virtual void Nhap();
	virtual float TinhDienTich();
};

#endif
```

***Tập tin Hinh.cpp**

```
#include "Hinh.h"

Hinh::Hinh()
{

}
Hinh::~Hinh()
{

}
void Hinh::Nhap()
{

}
float Hinh::TinhDienTich()
{
	return 0;
}
```

***Tập tin HinhChuNhat.h**

```
#ifndef _HINHCHUNHAT_
#define _HINHCHUNHAT_

#include "Hinh.h"
```

```
class HinhChuNhat : public Hinh
{
protected:
	float m_fChieuDai;
	float m_fChieuRong;
public:
	HinhChuNhat();
	HinhChuNhat(float fChieuDai, float fChieuRong);
	~HinhChuNhat();
	virtual void Nhap();
	virtual float TinhDienTich();
};

#endif
```

***Tập tin HinhChuNhat.cpp**

```
#include "HinhChuNhat.h"
#include <iostream>
using namespace std;

HinhChuNhat::HinhChuNhat()
{
	this->m_fChieuDai = 0;
	this->m_fChieuRong = 0;
}

HinhChuNhat::HinhChuNhat(float fChieuDai, float fChieuRong)
{
	this->m_fChieuDai  = fChieuDai;
	this->m_fChieuRong = fChieuRong;
}
HinhChuNhat::~HinhChuNhat()
{
	this->m_fChieuDai = 0;
	this->m_fChieuRong = 0;
	//cout<<"Da huy doi tuong hinh chu nhat!"<<endl;
}
void HinhChuNhat::Nhap()
{
	cout<<"*Nhap hinh chu nhat"<<endl;
	cout<<"Nhap chieu dai: ";
	cin>>this->m_fChieuDai;
	cout<<"Nhap chieu rong: ";
	cin>>this->m_fChieuRong;
}
float HinhChuNhat::TinhDienTich()
{
	float dt;
	dt = this->m_fChieuDai * this->m_fChieuRong;
	return dt;
}
```

***Tập tin HinhVuong.h**

```
#ifndef _HINHVUONG_
#define _HINHVUONG_

#include "HinhChuNhat.h"

class HinhVuong : public HinhChuNhat
{
public:
	HinhVuong(float a=0);
	~HinhVuong();
	virtual void Nhap();
};

#endif
```

***Tập tin HinhVuong.cpp**

```
#include "HinhVuong.h"
#include <iostream>
using namespace std;

HinhVuong::HinhVuong(float a)
{
	this->m_fChieuDai = a;
	this->m_fChieuRong = a;
}

HinhVuong::~HinhVuong()
{
	cout<<"Da huy doi tuong hinh vuong!!"<<endl;
}

void HinhVuong::Nhap()
{
	float canh;
	cout<<"*Nhap hinh vuong"<<endl;
	cout<<"Nhap canh hinh vuong: ";
	cin>>canh;
	this->m_fChieuDai = canh;
	this->m_fChieuRong = canh;
}
```

***Tập tin Ellipse.h**

```
#ifndef _ELLIPSE_
#define _ELLIPSE_

#define PI (float)3.14159

#include "Hinh.h"

class Ellipse : public Hinh
{
protected:
	float m_fRa; //ban kinh dai
	float m_fRb; //ban kinh ngan
public:
	Ellipse();
	Ellipse(float fRa, float fRb);
	~Ellipse();
	virtual void Nhap();
	virtual float TinhDienTich();
};

#endif
```

***Tập tin Ellipse.cpp**

```
#include "Ellipse.h"
#include <iostream>
using namespace std;

Ellipse::Ellipse()
{
	this->m_fRa = 0;
	this->m_fRb = 0;
}
Ellipse::Ellipse(float fRa, float fRb)
{
	this->m_fRa = fRa;
	this->m_fRb = fRb;
}
Ellipse::~Ellipse()
{
	this->m_fRa = 0;
	this->m_fRb = 0;
	//cout<<"Da huy doi tuong hinh Ellipse!"<<endl;
}
void Ellipse::Nhap()
{
	cout<<"*Nhap Ellipse"<<endl;
	cout<<"Nhap ban kinh dai: ";
	cin>>this->m_fRa;
	cout<<"Nhap ban kinh rong: ";
	cin>>this->m_fRb;
```

```
}
float Ellipse::TinhDienTich()
{
	float dt;
	dt = PI*this->m_fRa*this->m_fRb;
	return dt;
}
```

***Tập tin HinhTron.h**

```
#ifndef _HINHTRON_
#define _HINHTRON_

#include "Ellipse.h"

class HinhTron : public Ellipse
{
public:
	HinhTron(float fR=0);
	~HinhTron();
	virtual void Nhap();
};

#endif
```

***Tập tin HinhTron.cpp**

```
#include "HinhTron.h"
#include <iostream>
using namespace std;

HinhTron::HinhTron(float fR)
{
	this->m_fRa = fR;
	this->m_fRb = fR;
}

HinhTron::~HinhTron()
{
	//cout<<"Da huy doi tuong hinh tron!!"<<endl;
}

void HinhTron::Nhap()
{
	float fR;
	cout<<"*Nhap hinh tron"<<endl;
	cout<<"Nhap ban kinh hinh tron: ";
	cin>>fR;
	this->m_fRa = fR;
	this->m_fRb = fR;
}
```

***Tập tin ChuongTrinhChinh.cpp**

```
#include "HinhVuong.h"
#include "HinhChuNhat.h"
#include "Ellipse.h"
#include "HinhTron.h"
#include "conio.h"
#include <iostream>
using namespace std;

Hinh* TimHinhCoDienTichLonNhat(Hinh* arrHinh[], int iSoHinh)
{
	Hinh* HinhMax = NULL;

	if (iSoHinh > 0)
	{
		HinhMax = arrHinh[0];
		for (int i = 0; i < iSoHinh; i++)
		{
			if (HinhMax->TinhDienTich() < arrHinh[i]-
>TinhDienTich())
			{
				HinhMax = arrHinh[i];
			}
		}
	}

	return HinhMax;
}

void main()
{
	Hinh *arrHinh[50];
	int iSoHinh;

	cout<<"Nhap so hinh: ";
	cin>>iSoHinh;

	cout<<"\nChon loai hinh: ";
	cout<<"\n\t1.Hinh chu nhat";
	cout<<"\n\t2.Hinh vuong";
	cout<<"\n\t3.Ellipse";
	cout<<"\n\t4.Hinh tron"<<endl;

	for (int i = 0; i < iSoHinh; i++)
	{
		int loai;
		cout<<"Nhap loai cua hinh thu "<<i<<": ";
		cin>>loai;

		if (loai == 1)
			arrHinh[i] = new HinhChuNhat();
		if (loai == 2)
			arrHinh[i] = new HinhVuong();
		if (loai == 3)
			arrHinh[i] = new Ellipse();
```

```
            if (loai == 4)
                  arrHinh[i] = new HinhTron();
      }



      //ung dung da hinh: nhap
      for (int i = 0; i < iSoHinh; i++)
      {
            cout<<"Nhap hinh thu "<<i<<":"<<endl;
            arrHinh[i]->Nhap();
      }

      //ung dung da hinh: tim hinh co dien tich lon nhat
      Hinh* HinhMax = TimHinhCoDienTichLonNhat(arrHinh, iSoHinh);
      cout<<"Dien tich lon nhat: "<<HinhMax->TinhDienTich();

      getch();
}
```

BÀI 10: BÀI TẬP TỔNG HỢP 1 - QUẢN LÝ TRƯỜNG TRUNG HỌC PHỔ THÔNG

Đề bài:

- Một trường Trung học Phổ thông cần quản lý thông tin của giáo viên và học sinh trong trường. Trong đó: tất cả các thành viên (Người) trên đều có những thông tin chung như: Họ và tên, Giới tính, Năm sinh, Nơi sinh, Địa chỉ.
- Giáo viên còn có thêm những thông tin riêng của mình như : Năm bắt đầu giảng dạy, Chuyên môn.
- Học sinh có những thông tin riêng như : Điểm Toán, Điểm Văn, Điểm Ngoại ngữ.

Hãy thực hiện các yêu cầu sau:

a. Khai báo và thiết kế các lớp đối tượng (**Nguoi**, **GiaoVien**, **HocSinh**)

b. Trong lớp **Nguoi**, viết các phương thức sau:

b1. Phương thức khởi tạo

b2. Phương thức hủy

b3. Nhập thông tin

b4. Xuất thông tin

b5. Toán tử nhập: >>

b6. Toán tử xuất: <<

b7. Tính tuổi hiện tại

b8. Loại thành viên (GIAO_VIEN, HOC_SINH) (Phương thức thuần ảo)

c. Trong lớp **GiaoVien**, viết các phương thức sau:

c1. Phương thức khởi tạo

c2. Phương thức hủy

c3. Nhập thông tin

c4. Xuất thông tin

c5. Toán tử nhập: >>

c6. Toán tử xuất: <<

c7. Loại thành viên

c8. Tính số năm giảng dạy

d. Trong lớp **HocSinh**, viết các phương thức sau:

d1. Phương thức khởi tạo

d2. Phương thức hủy

d3. Nhập thông tin

d4. Xuất thông tin

d5. Toán tử nhập: >>

d6. Toán tử xuất: <<

d7. Loại thành viên

d8. Tính điểm trung bình. Biết rằng ĐTB = (Toán*2 + Văn*2 + NgoạiNgữ)/5

d9. Xếp loại học sinh dựa vào điểm Trung bình. Xếp loại được tính tiêu chuẩn như sau:

+ Giỏi : ĐTB trên 8 và không có môn dưới 6.5

+ Khá : ĐTB trên 6.5 và không có môn dưới 5

+ Trung Bình : ĐTB trên 5 và không có môn dưới 3

+ Kém : ĐTB dưới 5

e. Thiết kế lớp **Truong** đơn giản gồm n là số lượng thành viên, mảng arrThanhVien (Nguoi). Hãy thực hiện các yêu cầu sau:

e1. Khai báo lớp **Truong**

e2. Viết các phương thức khởi tạo

e3. Viết phương thức hủy

e4. Viết phương thức Nhập thông tin gồm n và mảng arrThanhVien (cho phép chọn loại đối tượng nhập (GiaoVien, HocSinh) - tham khảo thêm Ví dụ của HDThucHanhTuan9)

e5. Viết phương thức Xuất thông tin.

e6. Xuất danh sách học sinh xếp loại Khá.

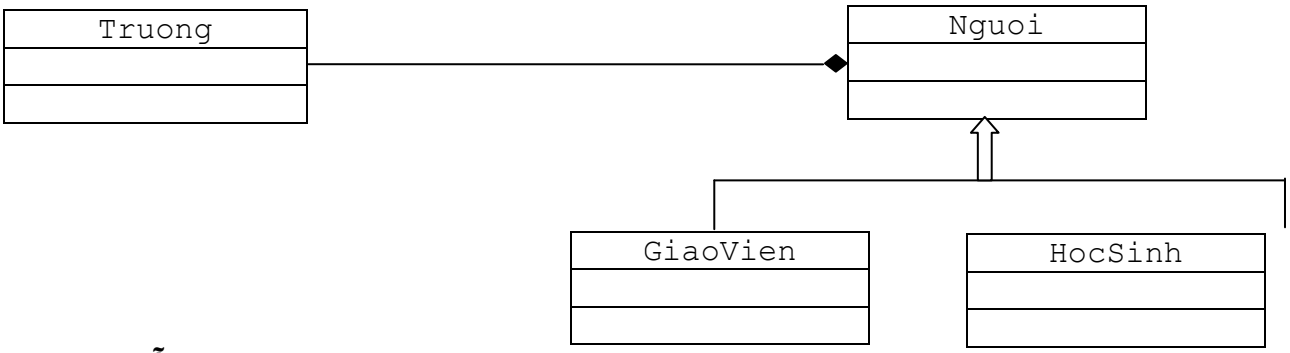
e7. Xuất danh sách học sinh xếp loại Giỏi.

e8. Xuất ra thông tin Thành viên (Nguoi) có tuổi lớn nhất.

e9. Xuất danh sách giáo viên có thâm niên trên 5 năm.

e10. Xuất danh sách giáo viên có thâm niên trên 10 năm và chuyên môn là Toán hoặc Lý.

Mô hình UML:



Hướng dẫn giải

1. Phần Khai báo

*** Tập tin Nguoi.h**

```
#ifndef _NGUOI_
#define _NGUOI_

#define GIAO_VIEN     1
#define HOC_SINH      2

#include "string.h"

class Nguoi
{
protected:
      char m_sHoTen[20];
      bool m_bGioiTinh; //true: Nam, false: Nu
      int  m_iNamSinh;
      char m_sNoiSinh[50];
      char m_sDiaChi[200];
public:
      Nguoi();
      Nguoi(char sHoTen[20], bool bGioiTinh, int iNamSinh, char
sNoiSinh[50],
            char sDiaChi[200]);
      ~Nguoi();

      virtual void Nhap();
      virtual void Xuat();

      int TinhTuoiHienTai();
      virtual int LoaiThanhVien() = 0; //phuong thuc thuan ao
};
#endif
```

*** Tập tin GiaoVien.h**

```
#ifndef _GIAOVIEN_
#define _GIAOVIEN_

#include "string.h"
```

```
#include "Nguoi.h"

class GiaoVien : public Nguoi
{
protected:
	int	m_iNamBatDauGD;
	char m_sChuyenMon[100];
public:
	GiaoVien();
	GiaoVien(char sHoTen[20], bool bGioiTinh, int iNamSinh, char
sNoiSinh[50],
		char sDiaChi[200], int iNamBatDauGD, char
sChuyenMon[100]);
	~GiaoVien();


	char* GetChuyenMon();

	virtual void Nhap();
	virtual void Xuat();

	int TinhSoNamGiangDay();
	virtual int LoaiThanhVien(); //phuong thuc thuan ao
};

#endif
```

*** Tập tin HocSinh.h**

```
#ifndef _HOCSINH_
#define _HOCSINH_

#define GIOI		1
#define KHA		2
#define TRUNG_BINH	3
#define KEM		4

#include "string.h"
#include "Nguoi.h"

class HocSinh : public Nguoi
{
protected:
	float m_fDiemToan;
	float m_fDiemVan;
	float m_fDiemNgoaiNgu;
public:
	HocSinh();
	HocSinh(char sHoTen[20], bool bGioiTinh, int iNamSinh, char
sNoiSinh[50], char sDiaChi[200], float fDiemToan, float fDiemVan,
float fDiemNgoaiNgu);
	~HocSinh();

	virtual void Nhap();
```

```
	virtual void Xuat();

	float TinhDiemTrungBinh();
	int XepLoaiHocSinh();
	virtual int LoaiThanhVien(); //phuong thuc thuan ao
};

#endif
```

*** Tập tin Truong.h**

```
#ifndef _TRUONG_
#define _TRUONG_

#include "Nguoi.h"

#define MAX 100

class Truong
{
protected:
	int	m_iN;  //so thanh vien
	Nguoi *m_arrThanhVien[MAX];  //mang cac lop doi tuong Nguoi
(GiaoVien, HocSinh)
public:
	Truong();
	~Truong();

	void NhapThongTinTruong();
	void XuatThongTinTruong();
	void XuatHocSinhXepLoaiKha();
	void XuatHocSinhXepLoaiGioi();

	void XuatThanhVienLonTuoiNhat();
	void XuatGiaoVienThamNienTren5Nam();
	void XuatGiaoVienThamNienTren10Nam_CMToan_Ly();
};

#endif
```

2. Phần Cài đặt cụ thể

*** Tập tin Nguoi.cpp**

```
#include "Nguoi.h"
#include "string.h"
#include <ctime>
#include <iostream>
using namespace std;

Nguoi::Nguoi()
{
	strcpy(m_sHoTen, "");
	m_bGioiTinh = true; //nam
```

```
	m_iNamSinh = 1900;  //mac dinh
	strcpy(m_sNoiSinh, "");
	strcpy(m_sDiaChi, "");
}
Nguoi::Nguoi(char sHoTen[20], bool bGioiTinh, int iNamSinh, char
sNoiSinh[50], char sDiaChi[200])
{
	strcpy(m_sHoTen, sHoTen);
	m_bGioiTinh = bGioiTinh;
	m_iNamSinh = iNamSinh;
	strcpy(m_sNoiSinh, sNoiSinh);
	strcpy(m_sDiaChi, sDiaChi);
}
Nguoi::~Nguoi()
{
	strcpy(m_sHoTen, "");
	m_bGioiTinh = true; //nam
	m_iNamSinh = 1900;
	strcpy(m_sNoiSinh, "");
	strcpy(m_sDiaChi, "");
}

// lưu ý: để đơn giản, ko kiểm tra dữ liệu nhập
void Nguoi::Nhap()
{
	flushall(); //xóa bộ nhớ đệm

	cout<<"Nhap ho ten: ";
	gets(this->m_sHoTen);

	flushall();

	int gt;
	cout<<"Nhap gioi tinh (0: Nu, 1: Nam): ";
	cin>>gt;

	if (gt == 0)
		this->m_bGioiTinh = false;
	else
		this->m_bGioiTinh = true;

	cout<<"Nhap nam sinh: ";
	cin>>this->m_iNamSinh;

	flushall();
	cout<<"Nhap noi sinh: ";
	gets(this->m_sNoiSinh);

	flushall();
	cout<<"Nhap dia chi: ";
	gets(this->m_sDiaChi);
}

void Nguoi::Xuat()
```

```
{
	cout<<"\nHo ten: "<<this->m_sHoTen;
	cout<<"\nGioi tinh: "<<this->m_bGioiTinh;
	cout<<"\nNam sinh: "<<this->m_iNamSinh;
	cout<<"\nNoi sinh: "<<this->m_sNoiSinh;
	cout<<"\nDia chi: "<<this->m_sDiaChi;
}

int Nguoi::TinhTuoiHienTai()
{
	int iTuoiHienTai;

	//Lấy năm hiện tại
	int iNamHienTai;
	time_t t = time(0); //lấy thời gian hiện tại

	//chuyển đổi sang thời gian hiện tại
	struct tm *now = localtime(&t);
	iNamHienTai = now->tm_year+1900;


	iTuoiHienTai = iNamHienTai - this->m_iNamSinh;
	/*
		Để đơn giản trong việc tính tuổi hiện tại thì
		iTuoiHienTai = 2012 - this->m_iNamSinh;
		Tuy nhiên, cách tính này sẽ sai nếu chạy chương trình
vào năm 2013!
	*/

	return iTuoiHienTai;
}
```

*** Tập tin GiaoVien.cpp**

```
#include "GiaoVien.h"
#include "Nguoi.h"
#include "string.h"
#include <ctime>
#include <iostream>
using namespace std;

GiaoVien::GiaoVien():Nguoi()
{
	this->m_iNamBatDauGD = 1900;
	strcpy(this->m_sChuyenMon,"");
}



GiaoVien::GiaoVien(char sHoTen[20], bool bGioiTinh, int iNamSinh,
char sNoiSinh[50], char sDiaChi[200], int iNamBatDauGD, char
sChuyenMon[100])
:Nguoi(sHoTen, bGioiTinh, iNamSinh, sNoiSinh, sDiaChi)
```

```
{
	this->m_iNamBatDauGD = iNamBatDauGD;
	strcpy(this->m_sChuyenMon,"");
}

GiaoVien::~GiaoVien()
{
	this->m_iNamBatDauGD = 1900;
	strcpy(this->m_sChuyenMon,"");
}

char* GiaoVien::GetChuyenMon()
{
	return this->m_sChuyenMon;
}

void GiaoVien::Nhap()
{
	Nguoi::Nhap();
	cout<<"Nhap nam bat dau giang day: ";
	cin>>this->m_iNamBatDauGD;
	flushall();
	cout<<"Nhap chuyen mon: ";
	gets(this->m_sChuyenMon);
}
void GiaoVien::Xuat()
{
	Nguoi::Xuat();
	cout<<"\nNam bat dau giang day: "<<this->m_iNamBatDauGD;
	cout<<"\nChuyen mon: "<<this->m_sChuyenMon;
}

int GiaoVien::TinhSoNamGiangDay()
{
	int iSoNamGD;

	//Lấy năm hiện tại
	int iNamHienTai;
	time_t t = time(0); //lấy thời gian hiện tại
	struct tm *now = localtime(&t); //chuyển đổi sang thời gian
hiện tại
	iNamHienTai = now->tm_year+1900;


	iSoNamGD = iNamHienTai - this->m_iNamBatDauGD;
	/*
		Để đơn giản trong việc tính số năm giảng dạy thì
		iSoNamGD = 2012 - this->m_iNamBatDauGD;
		Tuy nhiên, cách tính này sẽ sai nếu chạy chương trình
vào năm 2013!
	*/

	return iSoNamGD;
}
```

```
int GiaoVien::LoaiThanhVien()
{
	return GIAO_VIEN;
}
```

*** Tập tin HocSinh.cpp**

```
#include "HocSinh.h"
#include "Nguoi.h"
#include "string.h"
#include <ctime>
#include <iostream>
using namespace std;

HocSinh::HocSinh():Nguoi()
{
	this->m_fDiemToan = 0;
	this->m_fDiemVan = 0;
	this->m_fDiemNgoaiNgu = 0;
}

HocSinh::HocSinh(char sHoTen[20], bool bGioiTinh, int iNamSinh,
		char sNoiSinh[50], char sDiaChi[200], float
fDiemToan, float fDiemVan, float fDiemNgoaiNgu)
:Nguoi(sHoTen, bGioiTinh, iNamSinh, sNoiSinh, sDiaChi)
{
	this->m_fDiemToan = fDiemToan;
	this->m_fDiemVan = fDiemVan;
	this->m_fDiemNgoaiNgu = fDiemNgoaiNgu;
}

HocSinh::~HocSinh()
{
	this->m_fDiemToan = 0;
	this->m_fDiemVan = 0;
	this->m_fDiemNgoaiNgu = 0;
}

void HocSinh::Nhap()
{
	Nguoi::Nhap();

	cout<<"Nhap diem toan: ";
	cin>>this->m_fDiemToan;
	cout<<"Nhap diem van: ";
	cin>>this->m_fDiemVan;
	cout<<"Nhap diem ngoai ngu: ";
	cin>>this->m_fDiemNgoaiNgu;
}

void HocSinh::Xuat()
{
	Nguoi::Xuat();
```

```
	cout<<"\nDiem toan: "<<this->m_fDiemToan;
	cout<<"\nDiem van: "<<this->m_fDiemVan;
	cout<<"\nDiem ngoai ngu: "<<this->m_fDiemNgoaiNgu;
}
float HocSinh::TinhDiemTrungBinh()
{
	float DiemTB;

	DiemTB =(m_fDiemToan*2 + m_fDiemVan*2 + m_fDiemNgoaiNgu)/5;

	return DiemTB;
}

int HocSinh::XepLoaiHocSinh()
{
	int kq;

	float DiemTB;
	DiemTB = TinhDiemTrungBinh();

	if ((DiemTB >= 8) && (this->m_fDiemToan >=6.5) &&
	    (this->m_fDiemVan >= 6.5) &&
	    (this->m_fDiemNgoaiNgu >= 6.5))
	{
		kq = GIOI;
	}
	else
	{
		if ((DiemTB >= 6.5) && (this->m_fDiemToan >=5) &&
		    (this->m_fDiemVan >=5) &&
		    (this->m_fDiemNgoaiNgu >=5))
		{
			kq = KHA;
		}
		else
		{
			if ((DiemTB >= 5) && (this->m_fDiemToan >=3) &&
			    (this->m_fDiemVan >=3) &&
			    (this->m_fDiemNgoaiNgu >=3))
			{
				kq = TRUNG_BINH;
			}
			else
			{
				kq = KEM;
			}
		}
	}

	return kq;
}
int HocSinh::LoaiThanhVien()
{
```

```
	return HOC_SINH;
}
```

*** Tập tin Truong.cpp**

```
#include "GiaoVien.h"
#include "HocSinh.h"
#include "Nguoi.h"
#include "Truong.h"
#include "string.h"
#include <ctime>
#include <iostream>
using namespace std;

Truong::Truong()
{
	this->m_iN = 0;
	for (int i = 0; i < MAX; i++)
		m_arrThanhVien[i] = NULL;
}

Truong::~Truong()
{
	if (this->m_iN > 0)
	{
		for (int i = 0; i < this->m_iN; i++)
		{
			delete m_arrThanhVien[i];
			m_arrThanhVien[i] = NULL;
		}

		this->m_iN = 0;
	}
}

void Truong::NhapThongTinTruong()
{
	cout<<"Nhap so thanh vien: ";
	cin>>this->m_iN;

	cout<<"Chon loai thanh vien: ";
	cout<<"\n\t1. Giao vien";
	cout<<"\n\t2. Hoc sinh"<<endl;

	for (int i = 0; i < this->m_iN; i++)
	{
		int loai;
		cout<<"Nhap loai cua thanh vien thu "<<i<<": ";
		cin>>loai;

		if (loai == 1)
			this->m_arrThanhVien[i] = new GiaoVien();
		if (loai == 2)
			this->m_arrThanhVien[i] = new HocSinh();
```

```
	}

	//ứng dụng đa hình
	for (int i = 0; i < this->m_iN; i++)
		this->m_arrThanhVien[i]->Nhap();
}

void Truong::XuatThongTinTruong()
{
	cout<<"\n\nTong so thanh vien: "<<this->m_iN<<endl;

	for (int i = 0; i < this->m_iN; i++)
	{
		cout<<"\n*Thanh vien thu "<<i<<":";
		this->m_arrThanhVien[i]->Xuat();
	}
}
void Truong::XuatHocSinhXepLoaiKha()
{
	cout<<"\n\nDanh sach hoc sinh xep loai kha: "<<endl;
	for (int i = 0; i < this->m_iN; i++)
	{
		if (this->m_arrThanhVien[i]->LoaiThanhVien() ==
		    HOC_SINH)
		{
			HocSinh *hs = (HocSinh*)this->m_arrThanhVien[i];
			if (hs->XepLoaiHocSinh() == KHA)
			{
				cout<<endl;
				hs->Xuat();
			}
		}
	}
}
void Truong::XuatHocSinhXepLoaiGioi()
{
	cout<<"\n\nDanh sach hoc sinh xep loai gioi: "<<endl;
	for (int i = 0; i < this->m_iN; i++)
	{
		if (this->m_arrThanhVien[i]->LoaiThanhVien() ==
HOC_SINH)
		{
			HocSinh *hs = (HocSinh*)this->m_arrThanhVien[i];
			if (hs->XepLoaiHocSinh() == GIOI)
			{
				cout<<endl;
				hs->Xuat();
			}
		}
	}
}
void Truong::XuatThanhVienLonTuoiNhat()
{
	int max;
```

```
	Nguoi *ThanhVienLonNhat;
	cout<<"\n\nThanh vien lon tuoi nhat: "<<endl;
	max = this->m_arrThanhVien[0]->TinhTuoiHienTai();
	ThanhVienLonNhat = this->m_arrThanhVien[0];

	for (int i = 0; i < this->m_iN; i++)
	{
		if (this->m_arrThanhVien[i]->TinhTuoiHienTai() > max)
		{
			ThanhVienLonNhat = this->m_arrThanhVien[i];
			max = this->m_arrThanhVien[i]->TinhTuoiHienTai();
		}
	}
	ThanhVienLonNhat->Xuat();
	cout<<"\nSo tuoi lon nhat la: "<<max;
}

void Truong::XuatGiaoVienThamNienTren5Nam()
{
	cout<<"\n\nDanh sach giao vien tham nien tren 5 nam: "<<endl;

	for (int i = 0; i < this->m_iN; i++)
	{
		if (this->m_arrThanhVien[i]->LoaiThanhVien() ==
		    GIAO_VIEN)
		{
			GiaoVien *gv = (GiaoVien*)this->m_arrThanhVien[i];
			if (gv->TinhSoNamGiangDay() >= 5)
			{
				cout<<endl;
				gv->Xuat();
			}
		}
	}
}
void Truong::XuatGiaoVienThamNienTren10Nam_CMToan_Ly()
{
	cout<<"\n\nDanh sach giao vien tham nien tren 10 nam co
chuyen mon Toan
	      hoac Ly: "<<endl;

	for (int i = 0; i < this->m_iN; i++)
	{
		if (this->m_arrThanhVien[i]->LoaiThanhVien() ==
GIAO_VIEN)
		{
			GiaoVien *gv = (GiaoVien*)this->m_arrThanhVien[i];
			if (gv->TinhSoNamGiangDay() >= 10 &&
    (strcmp(gv->GetChuyenMon(),"Toan")==0||strcmp(gv-
>GetChuyenMon(),"Ly")==0))
			{
				cout<<endl;
				gv->Xuat();
			}
```

```
            }
      }
}
```

*** Tập tin ChuongTrinhChinh.cpp**

```
#include "conio.h"
#include "string.h"
#include "Truong.h"
#include <iostream>
using namespace std;

void main()
{
      Truong *tr = new Truong();

      tr->NhapThongTinTruong();

      tr->XuatThongTinTruong();

      tr->XuatHocSinhXepLoaiKha();

      tr->XuatHocSinhXepLoaiGioi();

      tr->XuatThanhVienLonTuoiNhat();

      tr->XuatGiaoVienThamNienTren5Nam();

      tr->XuatGiaoVienThamNienTren10Nam_CMToan_Ly();

      getch();
}
```

BÀI 11: BÀI TẬP TỔNG HỢP 2 - QUẢN LÝ NÔNG TRẠI

Đề bài:

- Một nông trại chăn nuôi có 3 loại gia súc: Bò, Cừu, và Dê. Mỗi loại gia súc đều có thể sinh con, cho sữa và phát ra tiếng kêu riêng của chúng. Biết rằng:

+ Số lượng con của mỗi loại gia súc là một số ngẫu nhiên từ 1->5

+ Số lít sữa mỗi gia súc cho là ngẫu nhiên. Cụ thể như sau:

* Bò: 0->20 lít

* Cừu: 0->5 lít

* Dê: 0->10 lít

+ Khi đói, các gia súc sẽ phát ra tiếng kêu để đòi ăn.

- Hãy thực hiện các yêu cầu sau:

a. Khai báo và thiết kế các lớp đối tượng (**GiaSuc**, **Bo**, **Cuu, De**)

b. Trong các lớp trên, viết các phương thức sau:

b1. Phương thức khởi tạo

b2. Phương thức hủy

b3. Nhập thông tin

b4. Xuất thông tin

c. Thiết kế lớp **NongTrai** đơn giản gồm n là số lượng gia súc, mảng **arrGiaSuc** (GiaSuc). Hãy thực hiện các yêu cầu sau:

c1. Khai báo lớp **NongTrai**

c2. Viết các phương thức khởi tạo

c3. Viết phương thức hủy

c4. Viết phương thức Nhập thông tin gồm n và mảng **arrGiaSuc** (cho phép chọn loại đối tượng nhập (Bo, Cuu, De) - tham khảo thêm Ví dụ của HDThucHanhTuan9)

c5. Viết phương thức Xuất thông tin.

c6. Viết phương thức xuất ra tiếng kêu của tất cả các gia súc.

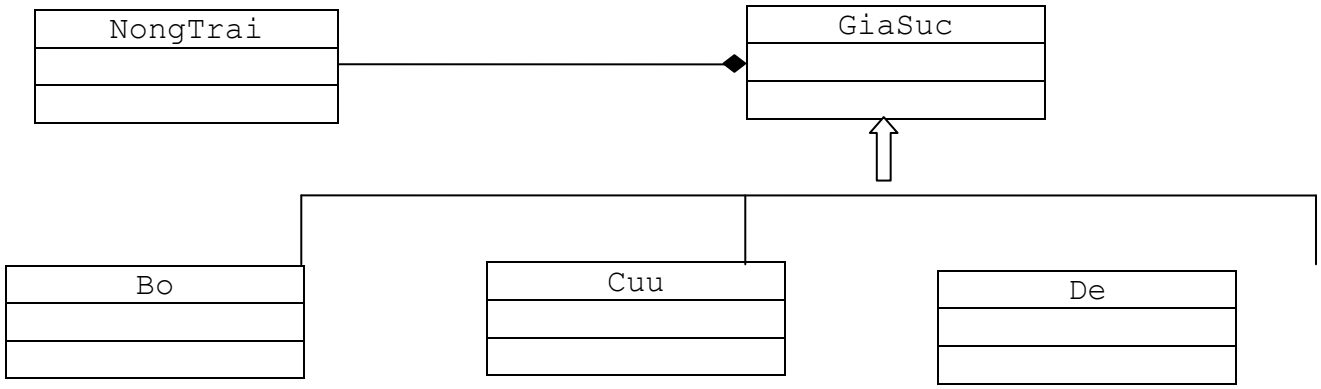
c7. Viết phương thức xuất ra số lượng các gia súc của mỗi loại.

c8. Viết phương thức cho biết tổng số lít sữa của tất cả các gia súc.

c9. Viết phương thức cho biết Gia súc nào sinh con nhiều nhất và số lượng con là bao nhiêu?

c10. Viết phương thức cho biết Gia súc nào cho số lít sữa ít nhất và số lít sữa là bao nhiêu?

Mô hình UML:



Gợi ý:

Vấn đề tạo số ngẫu nhiên trong Visual C++

```
#include "conio.h"
#include "stdlib.h"  //hàm rand
#include "time.h"    //hàm time
#include <iostream>
using namespace std;

void main()
{
	//khởi tạo bộ ngẫu nhiên (tạo ra các số khác nhau mỗi lần chạy)
	srand((unsigned)time(NULL));

	//Tạo 5 số, mỗi số có giá trị ngẫu nhiên từ 0->20
	for (int i = 0; i < 5; i++)
	{
		int n;
		n = rand() % 20;
		cout<<n<<" ";
	}

	getch();
}
```

BÀI 12: BÀI TẬP TỔNG HỢP 3 - QUẢN LÝ NHÂN SỰ

Đề bài:

- Công ty ABC cần xây dựng một ứng dụng quản lý nhân sự và tính lương cho nhân viên trong công ty như sau:

+ Quản lý thông tin về nhân viên: (Họ tên, ngày sinh, địa chỉ, điện thoại)

+ Tính lương cho nhân viên

Hiện công ty có 3 loại nhân viên và cách tính lương tương ứng cho từng loại nhân viên như sau:

+ Nhân viên sản xuất: số sản phẩm x 3,000đ

+ Nhân viên công nhật: số ngày công x 70,000đ

+ Nhân viên quản lý: lương cơ bản x hệ số lương

- Hãy thực hiện các yêu cầu sau:

a. Khai báo và thiết kế các lớp đối tượng (**NhanVien**, **NhanVienSanXuat**, **NhanVienCongNhat, NhanVienQuanLy**)

b. Trong các lớp trên, viết các phương thức sau:

b1. Phương thức khởi tạo

b2. Phương thức hủy

b3. Nhập thông tin

b4. Xuất thông tin

b5. Tính lương.

c. Thiết kế lớp **QuanLy** đơn giản gồm n là số lượng nhân viên, mảng **arrNhanVien** (NhanVien). Hãy thực hiện các yêu cầu sau:

c1. Khai báo lớp **QuanLy**

c2. Viết các phương thức khởi tạo

c3. Viết phương thức hủy

c4. Viết phương thức Nhập thông tin gồm n và mảng **arrNhanVien** (cho phép chọn loại đối tượng nhập (**NhanVienSanXuat**, **NhanVienCongNhat, NhanVienQuanLy**) - tham khảo thêm Ví dụ của HDThucHanhTuan9)

c5. Viết phương thức Xuất thông tin.

c6. Viết phương thức xuất ra thông tin của toàn bộ nhân viên.

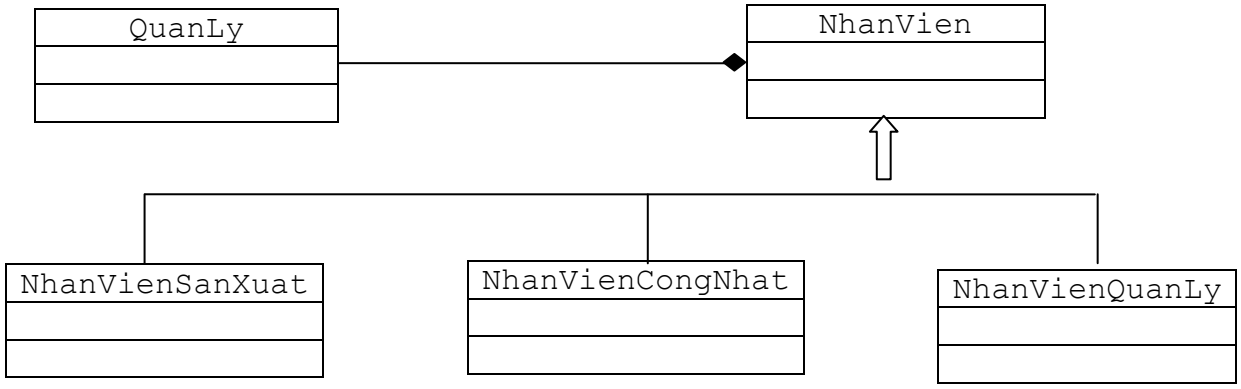
c7. Viết phương thức xuất ra số lượng nhân viên của mỗi loại.

c8. Viết phương thức cho biết tổng lương của tất cả nhân viên trong công ty.

c9. Viết phương thức cho biết nhân viên có lương cao nhất và số tiền là bao nhiêu?

c10. Viết phương thức sắp xếp tăng dần mảng arrNhanVien theo lương.

Mô hình UML:



Tài liệu tham khảo

1. Nguyễn Tấn Trần Minh Khang, Bài tập Kỹ Thuật Lập Trình (tập 1).
2. Trần Đan Thư (chủ biên), Nhập môn lập trình, NXB KHKT, 2011.