#include <bits/stdc++.h>

using namespace std;

* ***Class là lớp, lớp là tập hợp các đối tượng có cùng thuộc tính và phương thức.***
* ***Giống struct ở chỗ cho phép định nghĩa ra 1 kiểu dữ liệu mới.***
* ***Class ngoài việc có các thuộc tính (ví dụ ở dưới, class phân số sẽ có thuộc tính tử số và mẫu số) thì class còn cho phép định nghĩ các phương thức hay các hàm(nhập, xuất, rút gọn, cộng, nhân).***
* ***Ở dưới xuất hiện từ private và public, được gọi là phạm vi truy cập của thuộc tính hoặc phương thức trong class.*** 
  + ***Private là riêng tư, nghĩa là không cho phép truy cập từ bên ngoài class***
  + ***Public là công cộng thì cho phép truy cập từ mọi nơi.***
  + ***Như trong code, tử số và mẫu số là 2 thuộc tính riêng tư nên không thể truy cập tới chúng ở trong main, hoặc trong các hàm không được định nghĩa trong class.***

class PhanSo

{

private:

int iTuSo;

int iMauSo;

private:

int UCLN(int a, int b);

public:

PhanSo();

PhanSo(int iTu, int iMau);

void Nhap();

void Xuat();

void RutGon();

PhanSo Cong (PhanSo b);

PhanSo Nhan (PhanSo b);

***Khi khai báo các hàm của class, có thể liệt kê toàn bộ các tên hàm , kiểu dữ liệu và đối số trong class trước rồi mới triển khai hàm ở dưới hoặc có thể triển khai hàm ngay trong class.***

};

PhanSo::PhanSo() ***Class có 2 hàm đặc biệt là hàm tạo và hàm hủy, trong code chỉ có hàm tạo.***

{ ***Hàm tạo là hàm khi chúng ta dung new PhanSo() để tạo mới phân số***

iTuSo = 0; ***thì ngay lập tức tử số và mẫu số sẽ được gán như bên, ts = 0, ms=1***

iMauSo = 1;

}

PhanSo::PhanSo(int iTu, int iMau)

{

iTuSo = iTu;

if (iMau != 0)

{

iMauSo = iMau;

}

else

{

iMauSo = 1;

}

}

void PhanSo::Nhap()

{

cout << "\tNhap tu So: "; cin >> iTuSo;

cout << "\tNhap mau so: "; cin >> iMauSo;

}

void PhanSo::Xuat()

{

cout << iTuSo << "/" << iMauSo << "\n";

}

int PhanSo::UCLN(int a, int b) {

if (a==0 ||b==0) return a+b;

while (a !=b) {

if(a>b) a=a-b;

else b=b-a;

}

return a;

}

void PhanSo::RutGon()

{

int iUCLN = UCLN(iTuSo, iMauSo);

iTuSo = iTuSo / iUCLN;

iMauSo = iMauSo / iUCLN;

}

***Hàm cộng và nhân, ở trong đối có truyền đối số là một phân số, hàm có nghĩa là lấy phân số hiện tại đang dung cộng với phân số b.***

PhanSo PhanSo::Cong(PhanSo b)

{

return PhanSo((iTuSo\*b.iMauSo) + (iMauSo\*b.iTuSo),iMauSo\*b.iMauSo);

***Ở dòng trên, iTuSo có nghĩa là tử số của phân số hiện tại mình dùng để gọi hàm cộng.***

***Ví dụ: ở main chúng ta gọi hàm cộng như sau: ps1.Cong(ps2) nghĩa là lấy ps1 cộng với ps2***

}

PhanSo PhanSo::Nhan(PhanSo b)

{

return PhanSo(this->iTuSo\*b.iTuSo,this->iMauSo\*b.iMauSo);

***Từ this có ý nghĩa là đối tượng hiện tại, ví dụ gọi hàm: ps1.Nhan(ps2) thì this = ps1 và this->iTuSo = ps1.iTuSo***

}

int main()

{

PhanSo ps1,ps2;

cout << "Phan so 1:\n";

ps1.Nhap();

cout << "Phan so 2:\n";

ps2.Nhap();

PhanSo psTong = ps1.Cong(ps2);

cout << "\nCong hai phan so: ";

psTong.RutGon();

psTong.Xuat();

PhanSo psTich = ps1.Nhan(ps2);

cout << "\nNhan hai phan so: ";

psTich.RutGon();

psTich.Xuat();

}

# *Coppyright©vanmanh296*