

Nội dung tuần 07 + 08

Xây dựng các lớp đối tượng có quan hệ kế thừa.

Hướng dẫn

Cách gọi phương thức của lớp đối tượng cơ sở từ lớp đối tượng con (trùng tên phương thức)

Xét ví dụ sau với lớp đối tượng cơ sở là **NhanVien** và lớp đối tượng dẫn xuất là **NhanVienKyThuat**.

```
class NhanVien
{
private:
    string hoTen, diaChi;

public:
    NhanVien(void);
    NhanVien(const string& ht, const string& dc);
    ~NhanVien(void);

    friend ostream& operator<<(ostream& os, const NhanVien& nv);
};

NhanVien::NhanVien(void)
{
    hoTen = "";
    diaChi = "";
}

NhanVien::NhanVien(const string& ht, const string& dc)
{
    hoTen = ht;
    diaChi = dc;
}

NhanVien::~~NhanVien(void)
{
}

ostream& operator<<(ostream& os, const NhanVien& nv)
{
    os << nv.hoTen << " (" << nv.diaChi << ")";
    return os;
}
```

Cài đặt lớp đối tượng **NhanVienKyThuat**

```
class NhanVienKyThuat :
    public NhanVien
{
```

```
private:
    string chungChiNganh;
public:
    NhanVienKyThuat(void);
    NhanVienKyThuat(const string& ht, const string& dc, const string& cc);
    ~NhanVienKyThuat(void);

    friend ostream& operator<<(ostream& os, const NhanVienKyThuat& nvkt);
};

NhanVienKyThuat::NhanVienKyThuat(void)
    : NhanVien()
{
    chungChiNganh = "";
}

NhanVienKyThuat::NhanVienKyThuat(const string& ht, const string& dc, const string& cc)
    : NhanVien(ht, dc)
{
    chungChiNganh = cc;
}

NhanVienKyThuat::~NhanVienKyThuat(void)
{
}

ostream& operator<<(ostream& os, const NhanVienKyThuat& nvkt)
{
    //làm sao để xuất thông tin private ở trong class NhanVien?
    //dòng lệnh sau bị lỗi vì KHÔNG được phép truy suất vào thành
    //phần private của lớp đối tượng NhanVien
    os << nvkt.hoTen << " (" << nvkt.diaChi << ")";

    os << " [" << nvkt.chungChiNganh << "]";
    return os;
}
```

Ở đây vấn đề khó khăn là ở cả 2 lớp đối tượng NhanVien và NhanVienKyThuat đều có toán tử xuất (<<). Giải quyết vấn đề như sau

```
ostream& operator<<(ostream& os, const NhanVienKyThuat& nvkt)
{
    //làm sao để xuất thông tin private ở trong class NhanVien?
    //vì quan hệ IS-A nên hoàn toàn có thể coi NhanVienKyThuat
    //là một loại NhanVien
    os << (NhanVien)nvkt;

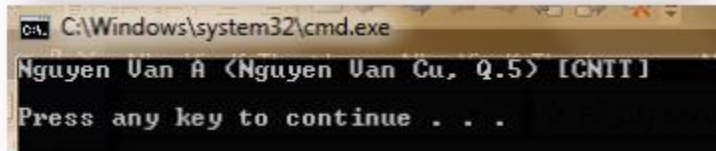
    os << " [" << nvkt.chungChiNganh << "]";
    return os;
}
```

Viết tiếp hàm **main** để chạy chương trình ta có kết quả

```
void main()
{
    NhanVienKyThuat nvkt("Nguyen Van A", "Nguyen Van Cu, Q.5", "CNTT");
}
```

Hướng dẫn thực hành PP LT hướng đối tượng

```
    cout << nvkt << endl << endl;  
}
```



Vấn đề khi cần gọi đúng phương thức của lớp đối tượng dẫn xuất

Xét ví dụ về cài đặt phương thức tính diện tích cho **HìnhChuNhat** và **HìnhTron** là dẫn xuất của **HìnhHocPhang**.

```
class HìnhHocPhang  
{  
public:  
    HìnhHocPhang(void);  
    ~HìnhHocPhang(void);  
  
    float TinhDienTich();  
};  
  
HìnhHocPhang::HìnhHocPhang(void)  
{  
}  
  
HìnhHocPhang::~HìnhHocPhang(void)  
{  
}  
  
float HìnhHocPhang::TinhDienTich()  
{  
    return 0;  
}
```

```
class HìnhChuNhat :  
    public HìnhHocPhang  
{  
private:  
    float chieuDai, chieuRong;  
  
public:  
    HìnhChuNhat(void);  
    HìnhChuNhat(const float& cd, const float& cr);  
    ~HìnhChuNhat(void);  
  
    float TinhDienTich();  
};  
  
HìnhChuNhat::HìnhChuNhat(void)  
{  
}  
  
HìnhChuNhat::HìnhChuNhat(const float& cd, const float& cr)
```

Hướng dẫn thực hành PP LT hướng đối tượng

```
{
    chieuDai = (cd==0)? 1 : abs(cd);
    chieuRong = (cr==0)? 1 : abs(cr);
}

HinhChuNhat::~HinhChuNhat(void)
{
}

float HinhChuNhat::TinhDienTich()
{
    return chieuDai * chieuRong;
}
```

```
#define PI 3.14159
class HinhTron :
    public HinhHocPhang
{
private:
    float banKinh;

public:
    HinhTron(void);
    HinhTron(const float& bk);
    ~HinhTron(void);

    float TinhDienTich();
};

HinhTron::HinhTron(void)
{
}

HinhTron::HinhTron(const float& bk)
{
    banKinh = (bk==0)? 1 : abs(bk);
}

HinhTron::~HinhTron(void)
{
}

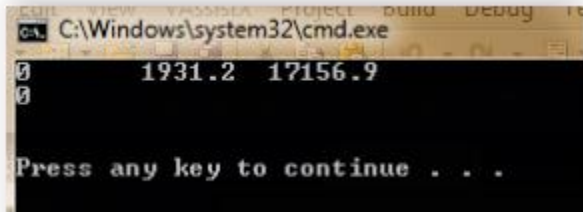
float HinhTron::TinhDienTich()
{
    return PI * banKinh * banKinh;
}
```

Cài đặt hàm main:

```
void main()
{
    HinhHocPhang hhp;
    HinhChuNhat hcn(34, 56.8);
    HinhTron ht(73.9);
    cout << hhp.TinhDienTich() << "\t" << hcn.TinhDienTich()
        << "\t" << ht.TinhDienTich();
}
```

Hướng dẫn thực hành PP LT hướng đối tượng

```
cout << endl;
//vì HìnhChuNhat là một loại HìnhHocPhang nên có thể gán
hhp = hcn;
//lúc này phương thức tính diện tích là của lớp đối tượng nào?
cout << hhp.TinhDienTich() << endl;
//Giải pháp???
cout << endl << endl;
}
```



Sử dụng phương thức với từ khóa virtual (ảo hóa)

Giải pháp cho vấn đề kể trên (mong muốn gọi đúng phương thức của đối tượng thật sự khi gọi) là sử dụng từ khóa **virtual** cho phương thức cần **override**.

Thay đổi trong lớp đối tượng cơ sở

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

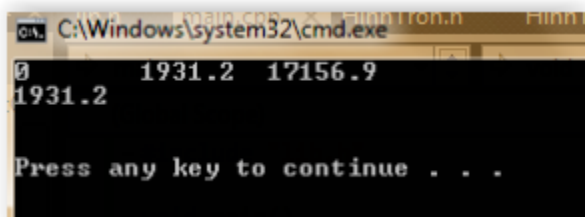
    virtual float TinhDienTich();
};
```

Ngoài ra cần phải sử dụng cấp phát động khi sử dụng thể hiện của lớp đối tượng.

Thay đổi ở hàm main:

```
void main()
{
    //sử dụng cấp phát động
    HìnhHocPhang *hhp = new HìnhHocPhang();
    HìnhChuNhat *hcn = new HìnhChuNhat(34, 56.8);
    HìnhTron *ht = new HìnhTron(73.9);
    cout << hhp->TinhDienTich() << "\t" << hcn->TinhDienTich()
        << "\t" << ht->TinhDienTich();
    cout << endl;
    //vì HìnhChuNhat là một loại HìnhHocPhang nên có thể gán
    delete hhp;
    hhp = hcn;
    cout << hhp->TinhDienTich() << endl;
    cout << endl << endl;
    delete hcn;
    delete ht;
}
```

Kết quả đã phù hợp



Bài tập

Bài 1

- Cài đặt lớp PS1 gồm có:
 - ☐ Dữ liệu: tử số, mẫu số
 - ☐ Phương thức: nhập ps(mẫu khác 0), in ps, tối giản, cộng 2 ps
 - ☐ Chương trình chính: nhập 2 ps ab, in ra $c=a+b$
- Cài đặt lớp PS2 kế thừa PS1 và bổ sung:
 - ☐ Dữ liệu: dấu của phân số
 - ☐ Phương thức: toán tử $>>$, $<<$, phép $-$, $*$, $/$, phép so sánh: $==$, $!=$, $>$, $>=$, $<$, $<=$, $++$, $--$

Chương trình chính: nhập 2 ps, thông báo kết quả so sánh.

Bài 2

- Cài đặt lớp SP1 gồm có:
 - ☐ Dữ liệu: phần thực, phần ảo
 - ☐ Phương thức: nhập, in
 - ☐ Chương trình chính: nhập 2 sp ab, in ra a, b
- Cài đặt lớp SP2 kế thừa SP1 và bổ sung:
 - ☐ Dữ liệu:
 - ☐ Phương thức: toán tử $>>$, $<<$, phép $-$, $*$, $/$, phép so sánh: $==$, $!=$, $>$, $>=$, $<$, $<=$, $++$, $--$

Hướng dẫn thực hành PP LT hướng đối tượng

- ❑ Chương trình chính: nhập 2 sp, thông báo kết quả so sánh, các phép tính $a+b$, $a-b$, $a*b$, a/c

Bài 3

- Cài đặt lớp NGUOI gồm có:
 - ❑ Dữ liệu: họ tên, ngày sinh, địa chỉ
 - ❑ Phương thức: nhập, in một người
- Cài đặt lớp NV kế thừa NGUOI và bổ sung:
 - ❑ Dữ liệu: phòng ban, hệ số lương, phụ cấp
 - ❑ Phương thức: định nghĩa lại pt nhập, in để có đầy đủ thông tin

Chương trình chính: nhập mảng các NV có n người ($n < 20$), in danh sách nhân viên.

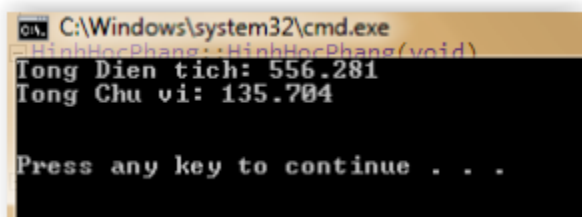
Bài 4

Xây dựng lớp đối tượng **QuanLyHinhHocPhang** cho phép thêm các loại đối tượng dẫn xuất của **HinhHocPhang** (như **HinhChuNhat** [chiều dài và chiều rộng], **HinhTron** [bán kính], **HinhTamGiac** [3 cạnh], **HinhThangVuong** [đáy lớn, đáy bé và chiều cao]), tính tổng các diện tích và chu vi sao cho hàm **main** sau chạy đúng.

```
void main()
{
    QuanLyHinhHocPhang ql;
    ql.AddHinh(new HinhChuNhat(5.6, 2.3));
    ql.AddHinh(new HinhTamGiac(-1.5, 6.7, 5.9));
    ql.AddHinh(new HinhTron(12.7));
    ql.AddHinh(new HinhThangVuong(10.1, 8.6, 3.5));

    cout << "Tong Dien tich: " << ql.TongDienTich() << endl;
    cout << "Tong Chu vi: " << ql.TongChuVi() << endl;
    cout << endl << endl;
}
```

Kết quả



Bài 5

Xây dựng class **CongTy** quản lý ds các **NhanVien** [họ tên và địa chỉ] với 2 loại **NhanVienThuKy** [chứng chỉ ngoại ngữ và số lượng báo cáo đã hoàn thành] và **NhanVienKyThuat** [chứng chỉ ngành và số lượng sáng kiến trong năm]. **CongTy** cho phép thêm nhân viên mới, xuất ds nhân viên của công ty và ds nhân viên xuất sắc (được đánh giá theo tiêu chí **NhanVienThuKy** phải hoàn thành ít nhất **12 báo cáo**, **NhanVienKyThuat** phải có ít nhất **6 sáng kiến trong năm**). Áp dụng cho hàm main sau

```
void main()
{
    CongTy cty;
    cty.ThemNhanVien(new NhanVienThuKy("Nguyen Thi A", "Tp Ho Chi Minh", "Toefl iBT 100", 15));
    cty.ThemNhanVien(new NhanVienKyThuat("Tran Van B", "Tp Ho Chi Minh", "CNTT", 8));
    cty.ThemNhanVien(new NhanVienThuKy("Le Nguyen Z", "Tp Ha Noi", "TOEIC 700", 10));
    cty.ThemNhanVien(new NhanVienKyThuat("Do Lan T", "Binh Duong", "Hoa thuc pham", 4));
    cty.ThemNhanVien(new NhanVienKyThuat("Hoang Van D", "Khanh Hoa", "Dien tu", 10));
    cty.ThemNhanVien(new NhanVienThuKy("Ngo E", "Tp Ha Noi", "IELTS 7.0", 12));

    cout << "DS Nhan vien trong cong ty:" << endl;
    cty.DanhSachNV(cout);
    cout << endl << "DS Nhan vien xuất sắc:" << endl;
    cty.DanhSachNVXuatSac(cout);
    cout << endl << endl;
}
```

Kết quả

```
DS Nhan vien trong cong ty:
Nguyen Thi A <Tp Ho Chi Minh> [INN: Toefl iBT 100], bao cao hoan thanh: 15
Tran Van B <Tp Ho Chi Minh> [Nganh: CNTT], sang tao: 8
Le Nguyen Z <Tp Ha Noi> [INN: TOEIC 700], bao cao hoan thanh: 10
Do Lan T <Binh Duong> [Nganh: Hoa thuc pham], sang tao: 4
Hoang Van D <Khanh Hoa> [Nganh: Dien tu], sang tao: 10
Ngo E <Tp Ha Noi> [INN: IELTS 7.0], bao cao hoan thanh: 12

DS Nhan vien xuất sắc:
Nguyen Thi A <Tp Ho Chi Minh> [INN: Toefl iBT 100], bao cao hoan thanh: 15
Tran Van B <Tp Ho Chi Minh> [Nganh: CNTT], sang tao: 8
Hoang Van D <Khanh Hoa> [Nganh: Dien tu], sang tao: 10
Ngo E <Tp Ha Noi> [INN: IELTS 7.0], bao cao hoan thanh: 12

Press any key to continue . . .
```