Đề 2011

Câu 1:

a. Trong hàm main(), có các phương thức khởi tạo tại các dòng sau:

Dòng 2: Phương thức khởi tạo mặc định không tham số => Cho MayLoc()

Dòng 5: Phương thức khởi tạo mặc định có tham số => cho MayLyTam(float, float)

Dòng 9: Phương thức hủy đối tượng: Đối tượng MayLyTam m1 bị hủy khi ra khỏi đoạn lệnh từ dòng 4 đến dòng 9 (hay chu kì sống của m1 chỉ tồn tại trong đoạn lệnh 4 - 9)

Dòng 10: Phương thức hủy đối tượng: Đối tượng *pm.

b. Các lỗi thực thi như sau:

/*1*/: Lỗi cú pháp => Lỗi đã truy cập vào thành phần riêng tự của lớp cơ sở.

/*2*/: Lỗi cú pháp => Trong main chỉ có thể truy cập vào cá thuộc tính và thành phần public của lớp. Các thành phần private và protected thì không thể được.

/*3*/: Lỗi thực thi => pm đã được cấp phát một vùng nhớ có kiểu MayLoc do đó ở đây ta cho pm quản lý địa chỉ của m1 là sai, mặc khác m1 có chu kì sống chỉ trong đoạn lệnh 4-9

/*4*/: Lỗi thực thi => pm đã đánh mất địa chỉ quản lý ban đầu do đó nếu ta delete pm thì chương trình sẽ không thể gaiir phóng vùng nhớ ban đầu đã cấp phát.

c. /*1*/ được chữa lại như sau:

```
this->setTgian(t)
// Cách khác: sữa từ khóa private trong MayLoc thành protected.
```

d. Kết quả hiện:

```
Luong nuoc = 819.0000
```

e. Dời lên dòng 03:

```
// Cách khác dời lên dòng 4.
// Cách khác: dời lên dòng 5.
```

Câu 2:

Lớp đối tượng được cài đặt như sau:

```
float MayLoc:: getThoiGian(){
    return this->m_tgian
}
class MayXucTac : public MayLoc {
private:
    int _luongHoaChat;
    float _congSuatLoc;
public:
    void nhap(){
        float thoiGian=0;
        cout<<"Nhap thoi gian hoat dong:";
        cin>>thoiGian;
        this->setTgian(thoiGian);
        cout<<"Nhap Luong hoa chat:";
        cin>>this->_luongHoaChat;
        cout<<"Nhap cong suat loc cua may:";
        cin>>this->_congSuatLoc;
    }
    float CongSuatThucTe(){
        if (this->getThoiGian()>=10){
            return this->_congSuatLoc * (this->_luongHoaChat/100)/(this-getThoiGian()/10);
        }
        else {
            return this->_congSuatLoc * (this->_luongHoaChat/100);
        }
    }
    long DON_GIA_HOA_CHAT;
    long DON_GIA_THUE;
    long tinhChoPhiThueMoiMay(){
        return DON_GIA_THUE * this->getThoiGian();
    }
    long tinhChiPhi(){
        return tinhChoPhiThueMoiMay() + this->_luongHoaChat*10000;
    }
    float tinhLuongNuoc(){
        return this->getThoiGian()*this->CongSuatThucTe();
    }
}
long MayXucTac::DON_GIA_HOA_CHAT=10000;
long MayXucTac::DON_GIA_THUE=80000;
```

Câu 3:

a. Vẽ sơ đồ tự vẽ nhé: trong sơ đồ sẽ có nhưng phương thức tương ứng phù hợp cho tất cả các phương thức đã vẽ ở trên.

b.

```
class DanhSachMayLoc {
private:
    MayLoc* _danhSach;
    int _N;
public:
    DanhSachMayLoc (){
        this->_N=0;
        this->_danhSach=NULL;
    }
    ~DanhSachMayLoc(){
        for(MayLoc* temp=this->_danhSach[0];temp<=this->_danhSach[_N-1];temp++){
            if(temp){
                delete temp;
                temp=NULL;
            }
            else {
                break;
            }
        }
    }
    long DON_GIA_THUE_LY_TAM;
    long DON_GIA_THUE_XUC_TAC;
    void nhap(){
        cout<<"Nhap luong may:";
        cin>>this->_N;
        for(int i=0;i<_N;i++){
            int choice=0;
            cout<<"Nhap loai may: 1 = may ly tam, 2 = may xuc tac:\n";
            cin>>choice;
            if(choice==1){
                this->_danhSach[i]=new MayLyTam();

            }
```

```
            else {
                this->_danhSach[i]=new MayXucTac();
            }
            this->_danhSach[i]->nhap();
        }
    }
    float tinhLuongNuoc(){
        float sum=0.0;
        for(int i=0;i<this->_N;i++){
            sum+=this->_danhSach[i]->tinhLuongNuoc();
        }
        return sum;
    }
    long tonhChiPhi(){
        long sum=0;
        for(int i=0;i<this->_N;i++){
            sum+=this->_danhSach[i]->tinhChiPhi();
        }
        return sum;
    }
}
long Ao::DON_GIA_THUE_LY_TAM=50000;
long Ao::DON_GIA_THUE_XUC_TAC=80000;
class Ao{
private:
    int _M;
    DanhSachMayLoc _ds;
public:
    void nhap(){
        cout<<"Nhap luong nuoc: ";
        cin>>this->_M;
        cout<<"Nhap danh sach:\n";
        this->_ds.nhap();
    }
    bool AoDuocLocHetKhong(){
        if(this->_M>this->_ds.tinhLuongNuoc()){
            return false;
        }
        else {
            return true;
        }
    }
    long tongChiPhi(){
        return this->_ds.tinhChiPhi();
    }
    Ao(){
        this->_M=0;
    }
    ~Ao(){
        this->_M=0;
        this->_ds.~DanhSachMayLoc();
    }
};
```

Đề 2012

Bài 1:

a. Kết quả xuất là:

```
c = 1/3 + 8 = 25/3
++c: c = 28/3
a = 1/3 + 28/3
a = 29/3
b = 29/3
```

b.

```
PhanSo();
PhanSo(const int&,const int&);
PhanSo operator+(const PhanSo&);
PhanSo operator+=(const PhanSo&);
PhanSo operator+(const int&);
PhanSo operator++();
operator int(int);
friend ostream& operator<<(ostream&, const PhanSo&);
~PhanSo();
```

c. Viết lại 5 phương thức:

```
PhanSo(){
    this->tu=0;
    this->mau=1;
}
    this->tu=0;
    this->mau=1;
}
PhanSo(const int&tu,const int& mau){
    this->tu=tu;
    this->mau=mau;
    if(this->mau==0){
        cout<<"Mau khong duoc bang 0\n";
        this->tu=0;
        this->mau=0;
    }
}
PhanSo operator+(const PhanSo& ps){
    PhanSo newPhanSo(this->tu*ps.mau+this->mau*ps.tu,this->mau*ps.mau);
    return newPhanSo;

}
PhanSo operator+=(const PhanSo& ps){
    this->tu=this->tu*ps.mau+this->mau*ps.tu;
    this->mau*=ps.mau;
    return *this;
}
PhanSo operator+(const int& iNum){
    PhanSo newPhanSo(this->tu+iNum*this->mau,this->mau);
    return newPhanSo;
}
PhanSo operator++(){
    this->tu+=this->mau;
    return *this;
}
```

```
        operator int(int iNum){
            PhanSo newPhanSo(iNum,1);
            return newPhanSo;
        }
        friend ostream& operator<<(ostream& os, const PhanSo& ps){
            os<<ps.tu<<"/"<<ps.mau;
            return os;
        }
        ~PhanSo(){
            this->tu=0;
            this->mau=1;
        }
```

Bài 2:

Mấy câu lý thuyết tự xử.

Bài 3:

a. Ý nghĩa như sau:

Từ khóa virtual: Giúp cái lớp dẫn xuất có thể thừa kế lại hoàn toàn tính chất này hoặc cơ thể viết chồng ngay lên (override)

Gán bằng 0: Bắt buộc dẫn xuất phải viết lại tính lại, nếu không viết lại sẽ báo lỗi cú pháp.

b. Lớp Cos như sau:

```
class Cos:public Function {
public:
    double Value(double x) {
    return cos(x);
    }
    double Derive(double x) {
    return -sin(x);
    }
};
```

c.

```
double DaoHamThuong(Function* f, Function* g, double x){
    if (g != NULL) {
        return (f->Derive()*g->Value()-f->Value()*g->Derive())/(g->Value()*g->Value())
    }
    return 0;
}

double DaoHamCuaHamHop(Function* f, Function* g, double x){
    if (f != NULL && g != NULL){
        return f->Derive(g->Value())*g->Derive();
    }
    return 0;
}
```

d.

Đề xuất sử dụng mẫu thiết kế singleton: Trong giáo trình có

Đề 2013

Câu 1:

a. Khai báo:// có thể làm theo viết phương thức bình thường, viết theo con trỏ hàm cũng được, ở đây trình bày cách nạp chồng toán tử.

```
        class A_B_Sqrt_X{
        private:
            int _a;
            int _b;
            int _SO_DUOI_DAU_CAN;
        public:
            A_B_Sqrt_X operator+(const A_B_Sqrt_X&);
            A_B_Sqrt_X operator-(const A_B_Sqrt_X&);
            A_B_Sqrt_X operator-();
            A_B_Sqrt_X operator*(const A_B_Sqrt_X&);
            friend ostream& operator<<(ostream&,const A_B_Sqrt_X&);
            friend istream& operator>>(istream&, A_B_Sqrt_X&);
        };
```

b. Cài đặt:

```
        int A_B_Sqrt_X::_SO_DUOI_DAU_CAN=7;
        A_B_Sqrt_X(const int& a, const int& b){
            this->_a=a;
            this->_b=b;
        }
        A_B_Sqrt_X A_B_Sqrt_X::operator+(const A_B_Sqrt_X& a_b_Sq_x){
            A_B_Sqrt_X result(this->_a+a_b_Sq_x._a,this->_b+a_b_Sq_x._b);
            return result;
        }
        A_B_Sqrt_X A_B_Sqrt_X::operator-(const A_B_Sqrt_X& a_b_Sq_x){
            A_B_Sqrt_X result(this->_a-a_b_Sq_x._a,this->_b-a_b_Sq_x._b);
            return result;
        }
        A_B_Sqrt_X A_B_Sqrt_X::operator-(){
            A_B_Sqrt_X result(-this->_a,-this->_b);
            return result;
        }
        A_B_Sqrt_X A_B_Sqrt_X::operator*(const A_B_Sqrt_X& a_b_Sq_x){
            A_B_Sqrt_X result;
            result._a=this->_a*a_b_Sq_x._a + this->_b*a_b_Sq_x._b*this->_SO_DUOI_DAU_CAN;
            result._b=this->_a*a_b_Sq_x._b + this->_b*a_b_Sq_x._a;
            return result;
        }
        friend ostream& operator<<(ostream& os, const A_B_Sqrt_X& a_b_Sq_x){
            os<<a_b_Sq_x._a<<" + ("<<a_b_Sq_x._b<<")*sqrt("<<a_b_Sq_x._SO_DUOI_DAU_CAN<<")";
            return os;
        }
        friend istream& operator>>(istream& is, A_B_Sqrt_X& a_b_Sq_x){
            is>>a_b_Sq_x._a;
            is>>a_b_Sq_x._b;
            return is;
        }
```

Câu 2: Lý thuyết tự đọc.

Câu 3:

a. Không thể lớp absList được. Lý Do: Vì đây là lớp trừu tượng với dấu hiệu addFirst(int pId), addFirst(int pId), showAll(ostream&) là phương thức thuần ảo.
Lớp absList được gọi là abstract Class tức là lớp trừu tượng.
b. Trả lời:
    Số lần hủy: Hủy 9 lần.
    Giải thích: Vì trong mỗi lần phương thức addFirst(int) hoạt động thì lnkLst trỏ vào một con trỏ mới và từ con trỏ đầu tiên ta truy xuất vào thành phần bên trong. Về mặt khác ta có thể nói trong lớp linearList absList* subLst chính là pNext của danh sách liên kết tự định nghĩa.
c. Cài đặt countAll như sau:
```
int countAll(){
    int count=0;
    for(absList* temp=this;temp;temp=temp->subLst){
        count++;
    }
    return count;
}
```
d. cài đặt void showAll(ostream&):
```
void showAll(ostream& outDev){
    for(absList* temp=this;temp;temp=temp->subLst){
        outDev<<temp->dataId<<" ";
    }
    outDev<<endl;
    return;
}
```
e. Giải:
    Cấu trúc dữ liệu: Ngăn xếp: stack
    Nhận xét: Phần này đợi hỏi người khác nhưng theo ý t thì t sẽ viết phương thức hủy danh sách.
```
~linearList(){
    while(this){
        LinearList* temp=this;
        this=this->subLst;
        delete temp;
        temp=NULL;
    }
}
```

Đề 2014

Câu 1:
    a.
    Bổ sung các phương thức sau:
```
Bike(){
    char s[4]="Bike";
    this->brand=new char[5];
    for(int i=0;i<5;i++){
        this->brand[i]=s[i];
    }
}
Bike(char s[]){
    this->brand=new char[strlen(s)+1];
    for(int i=0;i<strlen(s);i++){
        this->brand[i]=s[i];
    }
    this->brand[strlen(s)]=0;
}
~Bike(){
    if(this->brand){
        delete[] this->brand;
    }
}
EBike(char s[]):Bike("EBike"){}
```
    b. Kết quả xuất ra màn hình là:
        EBike: 48 EBike: 48
    c. Nhận xét:
        *Sự khác biệt:
        *Liệt kê các lớp đối tượng và các đối tượng.
            Lớp đối tượng: EBike.
            Các đối tượng: b1, b2. Khi truyền vào hàm display() thì ta vẫn giữ y các đối tượng đã liệt kê vid ở đây đã truyền vào theo kiểu tham chiếu nên vào hàm ta không tạo bản sao.
Câu 2:
    a. Lớp EyedFace:
```
class EyedFace:public IFace{
private:
    string shape;
    int eyes;
public:
    virtual void show();
    IFace* clone(){
        IFace* Clone=new EyedFace();
        Clone->eyes=this->eyes;
        Clone->shape=this->shape;
        return Clone;
    }
    virtual ~EyedFace(){
        this->eyes=0;
        this->shape="";
    }
    EyedFace(const string& sh, const int& e){
        this->shape=sh;
        this->eyes=e;
```

```
        }
        ~EyedFace(){
            this->shape="";
            this->eyes=0;
        }
    };
```

b. // không chắc đúng câu này.

Lớp Face thừa kế IFace theo tầm vực public. Do đó ta nhận thấy rằng nếu không viết lại các phương thức thuần ảo của IFace thì Face cũng là lớp trừu tượng. Vì vậy chương trình sẽ báo lỗi không thể tạo được đối tượng fc và fc1. Cách giải quyết ta cần viết lại các phương thức thuần ảo trong lớp Face.

Cách chữa lỗi cho main(), như sau:

Viết lại phương thức clone() trong Face như sau:

```
IFace* clone(){
    IFace* Clone=new Face();
    Clone->shape=this->shape;
    return Clone;
}
```

Kết quả xuất ra màn hình:

```
Shape: Rectangle
Shape: Rectangle
Shape: Rectangle
The same 3 line?
```

c. Cải tiến như sau:

Lỗi bộ nhớ trong hàm texstFace(): chữa như sau:

```
void testFace(IFace* fc) {
    IFace* a[3] = { fc, fc->clone(), fc->clone() };// đã tạo bộ nhớ nhưng lại chưa xóa trước khi kết thúc hàm.
    for(int i=0; i<3; i++) {
        a[i]->show();
    }
    cout << “The same 3 lines?”;
    delete a[1];
    delete a[2];
    return;
}
```

Thêm mã vào lớp EyesFace như sau:

```
.....
public:
    static int COUNT_OBJECT;
    EyedFace(const string& sh, const int& e){{
        .....
        COUNT_OBJECT++;
    }
    ~EyedFace(){
        ........
        COUNT_OBJECT--;
    }
};
// trong file .cpp
int EyedFace::COUNT_OBJECT=0;
// trong main()
int main(){
    ........
    cout<<"So doi tuong thuoc EyedFace"<<EyedFace::COUNT_OBJECT<<endl;
    return 0;
}
```

Câu 3:

```
class Phone_Fee{
private:
    float _time;
public:
    static long PRICE_PHONE;
    long calFee(){
        return PRICE_PHONE* this->_time;
    }
};
long Phone_Fee::PRICE_PHONE = 1000;

class Internet_Fee{
private:
    long _luongTruyCap;
public:
    static long PRICE_INTERNET;
    long calFee(){
        return PRICE_INTERNET*this->_luongTruyCap;
    }
};
long Internet_Fee::PRICE_INTERNET = 200;

class Customer{
private:
    string FullName;
    string ID;
    string Address;
public:
    void dangKy(){
        cout<<"Nhap ho va ten:";
        getline(cin,this->FullName);
        cin.ignore();
        cin.clear();
        cout<<"Nhap so chung minh:";
        cin>>this->ID;
        cout<<"Nhap dia chi:";
        getline(cin,this->Address);
    }
};


class Cost{
protected:
    Phone_Fee _phone_fee;
public:
    static long VAT;
    virtual long calFee()=0;
};
```

```
class Basic:public Cost{
private:
    Internet_Fee _internet_fee;
public:
    long calFee(){
        return this->_phone_fee.calFee()+this->_internet_fee.calFee() + 0.1 * VAT;
    }
};


class Data_Fee:public Data_Fee{
private:
    long _luongTruyCap;
public:
    static long NGUONG_MIEN_PHI;
    static long CUOC_THUE_BAO;
    long calFee(){
        long phone_fee=this->_phone_fee.calFee();
        long internet_fee=0;
        if(_luongTruyCap<=NGUONG_MIEN_PHI){
            internet_fee=CUOC_THUE_BAO;
        }
        else {
            Internet_Fee temp(this->_luongTruyCap-NGUONG_MIEN_PHI);
            internet_fee=CUOC_THUE_BAO+temp.calFee();
        }
        return phone_fee+internet_fee;
    }
};


class Data_Fix:public Cost{
public:
    static long MUC_CO_DINH;
    long calFee(){
        return (float)0.9*(this->_phone_fee.calFee())+ MUC_CO_DINH;
    }
};
long Data_Fix::MUC_CO_DINH = 1000000;

class Contract{
private:
    Customer _cus;
    Cost* _cost;
public:
    void dangKy(){
        this->_cus.dangKy();
        cout<<"Chon goi cuoc: 1-Basic,2-Data_Fee, 3-Data_Fix:\n";
        int choice;
        cin>>choice;
        if (choice==1){
            this->_cost=new Basic();
        }
        else if(choice==2){
            this->_cost=new Data_Fee();
        }
        else if(choice==3){
            this->_cost=new Data_Fix();
        }
        else {
            this->_cost =new Basic();
        }
    }
    void thongBao(){
        cout<<"Khach hang:\n";
        this->_cus.xuat();
        cout<<"Tien cuoc goi cuoc la:"<<this->_cost->calFee();
        cout<<endl;
    }
    ~Contract(){
        if(this->_cost){
            delete this->_cost;
            this->_cost=NULL;
        }
    }
};
class QuanLy{
    vector<Contract*> _ds;
public:
    void dangKy(){
        int n;
        cout<<"Nhap luong hop dong:";
        cin>>n;
        for(int i=0;i<n;i++){
            Contract* c=new Contract();
            c->dangKy();
            this->_ds.push_back(c);
        }
    }
    void thongBao(){
        for(int i=0;i<this->_ds.size();i++){
            this->_ds[i]->thongBao();
        }
    }
    ~QuanLy(){
        for(int i=0;i<this->_ds.size();i++){
            if(this->_ds[i]){
                this->_ds[i]->~Contract();
                delete this->_ds[i];
            }
        }
        this->_ds.resize(0);
    }
}
```

Đề 2015

Câu 1:

a. Chương trình hiện như sau:

3 4

Giải thích: Khi vào hàm doSomeThing(A a, B b); thì ta thấy các tham số được truyền theo kiểu tham trị nên khi vào hàm sẽ tạo bản sao, điều này dẫn đến x kiểu B khi truyền vào hàm sẽ tạo một bản sao và bản sao này được ép kiểu của lớp cơ sở nên kết quả trên màn hình là vậy.

b. Mã nguồn gặp vấn đề cấp phát mà quên giải phóng trước khi cho con trỏ gán đến một vùng địa chỉ khác. Cách giải quyết: Hàm init được viết lại như sau:

```
void init( A &a ) {
    A b;
    a.~A();
    a = b;
}
```

c. Câu này lý thuyết tự giải nhé.

Câu 2: Cài đặt BigInterger như sau:

```
class BigInterger{
    string _bigNum;
public:
    BigInterger(){
        this->_bigNum="";
    }
    BigInterger(string s){
        for(int i=0;i<s.size();i++){
            if(s[i]==' '){
                for(int j=i;j<s.size();j++){
                    s[i]=s[i+1];
                }
                i--;
                s[s.size()-1]=0;
            }
        }
        this->_bigNum=s;
    }
    friend istream& operator>>(istream& is, BigInterger& bigNum){
        getline(is,this->_bigNum);
        return is;
    }
    bool operator==(const BigInterger& bNum){
        if(this->_bigNum==bNum._bigNum){
            return true;
        }
        else {
            return false;
        }
    }
};
```

câu 3:

```
class Sap{
protected:
    int _stt;
    float _dt;
    long _doanhThu;
public:
    static long DON_GIA_THUE;
    virtual long tinhTienThue(){
        return DON_GIA_THUE*this->_dt;
    }
    virtual long tinhThueDoanhThu()=0;
    virtual long tinhTongTien()=0;
    virtual void nhapThongTin(){
        cin>>this->_stt;
        cin>>this->_dt;
        cin>>this->_doanhThu;
    }
    virtual ~Sap(){
        this->_stt=0;
        this->_dt=0;
        this->_doanhThu=0;
    }
};
long Sap::DON_GIA_THUE=40000000;
class SapThucPham:public Sap{
private:
    long _phiDongLanh;
public:
    static float PHAN_TRAM_THUE_SAP_THUC_PHAM;
    long tinhThueDoanhThu(){
        return (float)PHAN_TRAM_THUE_SAP_THUC_PHAM*this->_doanhThu;
    }
    long tinhTongTien(){
        return this->_phiDongLanh+this->tinhThueDoanhThu()+this->tinhTienThue;
    }
    virtual void nhapThongTin(){
        Sap::nhapThongTin();
        cin>>_phiDongLanh;
    }
};
long SapThucPham::PHAN_TRAM_THUE_SAP_THUC_PHAM=0.05;
class SapQuanAo:public Sap{
public:
    static float PHAN_TRAM_THUE_SAP_QUAN_AO;
    long tinhThueDoanhThu(){
        return (float)PHAN_TRAM_THUE_SAP_QUAN_AO*this->_doanhThu;
    }
    long tinhTongTien(){
        return this->tinhThueDoanhThu()+this->tinhTienThue();
    }
};
float SapQuanAo::PHAN_TRAM_THUE_SAP_QUAN_AO=0.1;
class SapTrangSuc:public Sap{
public:
    static long GIOI_HAN_DOANH_THU;
    static float PHAN_TRAM_THUE_DUOI_GION_HAN;
    static float PHAN_TRAM_THUE_TREN_GION_HAN;
    long tinhThueDoanhThu(){
```

```
        if(this->_doanhThu<GIOI_HAN_DOANH_THU){
            return PHAN_TRAM_THUE_DUOI_GION_HAN*this->_doanhThu;
        }
        else {
            return PHAN_TRAM_THUE_TREN_GION_HAN*this->_doanhThu;
        }
    }
    long tinhTongTien(){
        return this->tinhThueDoanhThu()+this->tinhTienThue();
    }
};
long SapTrangSuc::GIOI_HAN_DOANH_THU=100000000;
float SapTrangSuc::PHAN_TRAM_THUE_DUOI_GION_HAN=0.2;
float SapTrangSuc::PHAN_TRAM_THUE_TREN_GION_HAN=0.3;
class QuanLyDanhSach{
private:
    vector<Sap*> _ds;
public:
    void nhapThongTin(){
        cout<<"Nhap so luong sap duoc thue:";
        int n;
        cin>>n;
        for(int i=0;i<this->_ds.size();i++){
            int choice;
            cout<<"1-Sap Thuc Pham, 2-SapQuanAo, 3-SapTrangSuc\n";
            cin>>choice;
            Sap* p=NULL;
            if(choice==1){
                p=new SapThucPham();
            }
            else if(choice==2) {
                p=new SapQuanAo();
            }
            else if(choice==3){
                p=new SapTrangSuc();
            }
            else{
                cout<<"Nhap sai! Nhap lai!\n";
                i--;
                continue;
            }
            this->_ds.push_back(p);
            this->_ds[i]->nhapThongTin();
        }
    }
    long tinhTongTien(){
        long sum=0;
        for(int i=0;i<this->_ds.size();i++){
            sum+=this->ds[i]->tinhTongTien();
        }
        return sum;
    }
};
```

Đề 2016

Câu 1:

a. Chương trình xuất ra:

```
ellipse:
Ellipse(a=3,b=0.33333)
3.14
Ellipse
Ellipse
shape=&ellipse:
Ellipse(a=3,b=0.33333)
3.14
Ellipse
Shape
```

b.

* Shape::Description() và Circle::Description(): quan hệ override lại các phương thức trong lớp cơ sở cho phù hợp với lớp dẫn xuất.
* Ellipse::Scale(float) và Ellipse::Scale(float, float): quan hệ đa hình hóa đối tượng với 1 phương thức với nhiều cách dùng khác nhau.
* Shape::InterfaceType() và Circle::InterfaceType(): Quan hệ kế thừa. Trong khi lớp dẫn xuất không có phương thức đó thì có thể kế thừa lớp cơ sở (trong giới hạn của tầm vực kế thừa).
* Circle::InterfaceType() và Ellipse::InterfaceType(): Quan hệ đa hình hóa.

c. Nếu bỏ thì sẽ gặp lỗi:

+ dòng 88: lỗi vì con trỏ trong lớp shape không có phương thức nào là Scale(float,float); nên con trỏ shape sẽ không trỏ đến được phương thức này trong lớp Ellipse.
+ dòng 102+103: Shape có phương thức thuần ảo nên Shape chính là lớp trừu tượng vì vậy việc khai báo một đối tượng có thật trong main() là không thể được như vậy sẽ gây lỗi cú pháp.

d. Sữa lại như sau:

+ dòng 88:

```
ép kiểu bình thường:
(Ellipse)shape->Scale(3, 1.0/3);
// ép kiểu động:
// shape=dynamic_cast<Ellipse*>(shape)
```

+ dòng 102: sữa lại như sau đổi kiểu khai báo từ đối tượng thành con trỏ:

```
Shape* ashape;
```

+ dòng 103: sữa lại như sau:

```
cout<<ashape->Description()<<endl;
```

Câu 2:

a. Sơ đồ tự vẽ:

b. cài đặt:

```
class BenhNhan{
protected:
    int MSBN;
public:
    virtual void nhap(istream& is){
        is>>this->MSBN;
    }
    virtual void capNhatNgayXuatVien(int ngay);
    virtual bool getTrangThai(){
        return true;
    }
    virtual long tinhTien(){
        return 0;
    }
```

```
    virtual void capNhatVienPhi(){

    }
};
class BenhNhanNgoaiTru{
private:
    long _vienPhi;
public:
    BenhNhanNgoaiTru(int id){
        this->MSBN=id;
    }
    virtual void nhap(istream& is){
        is>>this->_vienPhi;
    }
    long tinhTien(){
        return this->_vienPhi;
    }
    void capNhatVienPhi(long Tien){
        this->_vienPhi+=Tien;
    }
};
class BenhNhanNoiTru{
private:
    int _loaiPhong;//a=1,b=2,c=3
    int _ngayBatDauNamVien;
    int _soNgayNamVien;
    bool _daXuatVien;//true: da, false: chua
    long _phiKhamBenhMoiNgay;
public:
    static long A;
    static long B;
    static long C;
    void nhap(istream& is){
        is>>this->_phiKhamBenhMoiNgay;
        char c;
        is>>c;
        this->_loaiPhong=c-64;
    }
    BenhNhanNoiTru(int MaSo, int ngay, long tien, char c){
        this->MSBN=MaSo;
        this->_ngayBatDauNamVien=ngay;
        this->_phiKhamBenhMoiNgay=tien;
        this->_loaiPhong=c-64;
        this->_daXuatVien=false;
    }
    void capNhatNgayXuatVien(int ngay){
        this->_soNgayNamVien=ngay-this->_ngayBatDauNamVien;
        this->_daXuatVien=true;
    }
    long tinhTien(){
        if(this->_loaiPhong==1){
            return this->A*(this->_soNgayNamVien+this->_phiKhamBenhMoiNgay);
        }
        else if(this->_loaiPhong==2){
            return this->B*(this->_soNgayNamVien+this->_phiKhamBenhMoiNgay);
        }
        else if(this->_loaiPhong==3){
            return this->C*(this->_soNgayNamVien+this->_phiKhamBenhMoiNgay);
        }
        else {
            return 0;
        }
    }
};
long BenhNhanNoiTru::A=1400000;
long BenhNhanNoiTru::B=900000;
long BenhNhanNoiTru::C=600000;


class DanhSachBenhNhan{
    vector<BenhNhan*> _ds;
public:
    void nhap(){
        freopen("input.log","rt",stdin);
        int ngay=0;
        while(!cin.eof()){
            cin>>ngay;
            int MaSo=0;
            cin>>MaSo;
            cin.ignore();
            cin.clear();
            string s="";
            cin>>s;
            BenhNhan* p=NULL;
            if(s=="KB"){
                long Tien=0;
                cin>>Tien;
                if(MaSo<=this->_ds.size()-1){
                    this->_ds[MaSo-1]->capNhatVienPhi(Tien);
                }
                else {
                    p=new BenhNhanNgoaiTru (MaSo,Tien);
                    this->_ds.push_back(p);
                }
            }
            else if(s=="NV"){
                long Tien=0;
                cin>>Tien;
                char c;
                cin>>c;
                p=new BenhNhanNoiTru(MaSo,ngay,Tien,c)
                this->_ds.push_back(p);
            }
            else if(s=="XV"){
                this->_ds[MaSo-1]->capNhatNgayXuatVien(ngay);
            }
```

```
            else if(s=="TKVP"){
                for(int i=0;i<this->_ds.size();i++){
                    if(dynamic_cast<BenhNhanNgoaiTru*>(this->_ds[i])==NULL){
                        if(this->_ds[i]->getTrangThai()==false){
                            this->_ds[i]->capNhatNgayXuatVien(ngay);
                        }
                    }
                }
            }
        }
        freopen("CON","rt",stdin);
    }
    long tinhTongTienVienPhi(){
        long sum=0;
        for(int i=0;i<this->_ds.size();i++){
            sum+=this->_ds[i]->tinhTien();
        }
        return sum;
    }
};
```

c. T làm không nỗi!