Họ và tên : Liên Gia Bảo  
MSSV : N22DCDK009  
Lớp : D22CQDK01-N

**Đề 2014**

Câu 1:

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

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;

}