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

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:

Hàm hủy ảo (virtual destructor) trong C++ được sử dụng để đảm bảo rằng hàm hủy của lớp dẫn xuất cũng được gọi khi giải phóng bộ nhớ cho đối tượng thông qua con trỏ của lớp cơ sở. Nếu không sử dụng hàm hủy ảo, có nguy cơ gọi chỉ hàm hủy của lớp cơ sở, dẫn đến mất mát dữ liệu và tài nguyên.

Ví dụ code thì trên git

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;

}

}