// 3.a In programul urmator, supradefiniti operatorul << astfel incat cout<<i sa afiseze valoarea atributului i.x

// 3.b Precizati si explicati rezultatele afisate la executarea programului astfel obtinut

#include <iostream.h>

class C{

public:

C(int i=0){x=i;}

C& operator++(){++x; return \*this;}

C operator--(){--x; return \*this;}

private:

int x;

};

void main(){

C i;

cout<<i<<endl;

cout<<++(++i)<<endl<<i<<endl;

cout<<--(--i)<<endl<<i<<endl;

}

// 4.

4.1 Inlocuiti . . . in clasa Stack, astfel incat metodele push si pop sa asigure tratarea exceptiilor. Numarul maxim de elemente din vectorul supporteste dat de expresia suport.length.

4,2 Scrieti o aplicatie in care sa tratati exceptiile lansate de push si pop.

class Stack{

int varf;

Object suport[];

void push(Object x). . .{. . .}

Object pop(). . .{. . .}

void init(int s){

varf=0;

support=new Object[s];

}

Stack(int s){. . .}

}

//5.

Declarati si implementati clasele din urmatoarea diagrama UML (Unified Modeling Language)

Student

+Student(char \*n, char \*f)

+virtual void afisare()

-char \*facultate

Profesor

+Profesor(char \*n, char \*f)

+virtual void afisare()

+stabileste\_specialiatea(char \*s)

-char \*facultate

-char \*specialitate

Persoana

+Persoana(char \*n)

+ virtual void afisare()

-char \*nume

6 . Fie programul urmator:

class Person{

public:

Person(char \*n){name=n;}

private:

char \* name;

};

void f(char \*n){

Person \*h=new Person (n);

delete h;

}

void main(){

char \*a;

a=new char[10];

for(int i=0;i<10;i++)a[i]='X';

while(1) f(a);

}

1. Explicati situatia anormala ce apare la executarea acestui program
2. Modificati clasa Person, astfel incat situatia anormala de le punctual a sa nu mai apara.
3. Explicati de ce, in cazul unui program Java similar, situatia anormala de la punctual a nu apare.
4. Cand se termina programul modificat de la punctual b si programul de la Java de la punctual c?