Ministerul Educaţiei, Culturii și Cercetării

al Republicii Moldova   
Universitatea Tehnică a Moldovei

FCIM  
  
  
  
  
  
  
  
RAPORT

# LUCRARE DE LABORATOR NR. 1

**LA DISCIPLINA „PROGRAMAREA ORIENTATĂ PE OBIECTE”**

A efectuat: st. gr. TI-211 Popa Cătălin   
  
A verificat: Lisnic Inga

UTM, Chișinău 2022

**Tema:**

Structura – mecanism de abstractizare.

**Scopul lucrării:**

Studierea programării prin abstractizarea datelor;

Studierea regulilor de definire şi utilizare a structurilor de date;

Crearea variabilelor de tip structură, accesarea componentelor unei structuri.

**Problema**

10. Să se creeze tipul abstract de date (structura) – vector, care are pointer spre *long* şi numărul de elemente. Să se definească funcţiile de iniţializare, de eliminare a vectorului, de setare/modificare a dimensiunii, de acces la elementele vectorului, de calcul a sumei elementelor negative ale vectorului. Ca exemplu, în funcţia *main* să se realizeze adunarea a doi vectori.

#include<iostream>

#include<string.h>

#include<stdlib.h>

#include<conio.h>

using namespace std;

struct vector{

float \*element;

long number;};

////////////////

int v2[50],x;

void distance(vector \*z)

{

cout<<"Introduceti dimensiunea vectorului: \n";

cin>> z->number;

z->element = new float[z->number];

}

/////////////////

void elements(vector \*z)

{

int i;

cout<<"Introduceti elementele vectorului: \n";

for(i=0;i<z->number;i++)

{

cout<<"Elementul["<<i+1<<"]:: ";

cin>>z->element[i];

}

}

/////////////

void second\_vector\_create(vector \*z)

{

cout <<"Introduceti dimensiunea: \n";

cin >> x;

cout <<"Introduceti elementele vectorului secundar: \n";

for(int i=0; i<x; i++)

{

cout <<"Elementul[" <<i+1<< "]= ";

cin >>v2[i];

}

}

//////////////////////

delete\_vector(vector \*z)

{

delete[] z->element;

z->element = NULL;

z->number = 0;

cout <<"\nAti sters vectorul cu succes!!!\n";

}

//////////////////////

void show(vector \*z)

{

if(z->number != 0)

{

cout <<"Vectorul este: \n";

for(int i=0; i<z->number; i++)

cout << z->element[i] <<" ";

if(z->number != 0)

{

cout << "\n\n Sunt "<<z->number<<" numere";

}

else

{

cout <<"\n !!!Nu sunt numere!!!";

}

}

else

{

cout <<"\n Vectorul nu a fost creat";

}

}

//////////////////

re\_vector(vector \*z)

{

int new\_dimension;

cout <<"Introduceti dimensiunea noua a vectorului: \n";

cin >>new\_dimension;

if(z->number > new\_dimension)

{

for(int i=new\_dimension+1; i<z->number; i++)

{

delete[] z->element;

z->element == NULL;

}

z->number = new\_dimension;

}

}

//////////////////

check\_element(vector \*z)

{

int position;

cout <<"Introduceti pozitia elementului dorit: \n";

cin >>position;

if(position > z->number)

{

cout << "\n Nu exista element pe aceasta pozitie!!!";

}

else

{

cout <<"Elementul accesat este: ";

for(int i=0; i<=z->number; i++)

{

if(i == position)

{

cout << z->element[i-1];

}

}

}

}

/////////////////

suma(vector \*z)

{

int sum=0;

for(int i=0; i<z->number;i++)

if(z->element[i] < 0)

sum+=z->element[i];

cout<<"Suma elementelor negative ale vectorului este: "<<sum;

}

/////////////////////

void Menu()

{

cout<<"\n[1]-Initializarea vectorului\n";

cout<<"[2]-Afisare\n";

cout<<"[3]-Eliminarea vectorului\n";

cout<<"[4]-Modificarea dimensiunii vectorului\n";

cout<<"[5]-Acces la elementele vectorului\n";

cout<<"[6]-Suma elementelor negative\n";

cout<<"[7]-Crearea vectorului secundar\n";

cout<<"[8]-Suma a 2 vectori\n";

}

////////////////////////////////////////////////////////////

main()

{

vector v;

int i;

int ve[100];

int sum = 0;

///Menu

menu:Menu();

int choice;

char confirm;

do

{

cout<<"\n\nAlege optiunea dorita:\n";

cin>>choice;

switch(choice)

{

///////Initializare

case 1: system("cls");distance(&v),elements(&v);

cout<<"\n";

cout<<"\n Enter - Inapoi\n";

case0: confirm = getch();

system("cls");

if (confirm == 13)

goto menu;

else

goto case0;

break;

////// Afisare

case 2:

system("cls");

show(&v);

cout<<"\n";

cout<<"\n Enter - Inapoi\n";

case1: confirm = getch();

system("cls");

if (confirm == 13)

goto menu;

else

goto case1;

break;

///// Eliminarea

case 3: delete\_vector(&v);

cout<<"\n";

cout<<"\n Enter - Inapoi\n";

case2: confirm = getch();

system("cls");

if (confirm == 13)

goto menu;

else

goto case2;

break;

///// Modificarea

case 4: re\_vector(&v);

cout<<"\n";

cout<<"\n Enter - Inapoi\n";

case3: confirm = getch();

system("cls");

if (confirm == 13)

goto menu;

else

goto case3;

break;

///////Accesarea

case 5: check\_element(&v);

cout<<"\n";

cout<<"\n Enter - Inapoi\n";

case4: confirm = getch();

system("cls");

if (confirm == 13)

goto menu;

else

goto case4;

break;

/////// Suma elementelor negative

case 6: suma(&v);

cout<<"\n";

cout<<"\n Enter - Inapoi\n";

case5: confirm = getch();

system("cls");

if (confirm == 13)

goto menu;

else

goto case5;

break;

///// Al 2-lea - vector

case 7: second\_vector\_create(&v);

cout<<"\n";

cout<<"\n Enter - Inapoi\n";

case6: confirm = getch();

system("cls");

if (confirm == 13)

goto menu;

else

goto case6;

break;

/////Adunarea vectorilor

case 8:

if((v2[0] == 0) || (v.number == 0))

{

cout<<"\n Nu exsita vectori in destul pentru operatiune\n";

getch();

goto menu;

}

else if(x == v.number)

{

cout<<"\nSuma vectorilor: \n";

for(i=0; i<v.number; i++)

{

sum = v2[i] + v.element[i];

cout<<sum<<" ";

}

}

else

{

cout <<"\n Vectorii sunt de dimensiune diferita!!\n";

}

cout<<"\n";

cout<<"\n Enter - Inapoi\n";

case7: confirm = getch();

system("cls");

if (confirm == 13)

goto menu;

else

goto case7;

break;

default: cout<<"invalid"; break;

system("cls");

}

cin>>confirm;

}

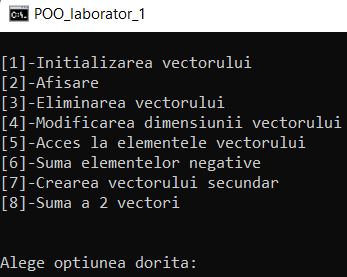
while (confirm == 13);

return 0;

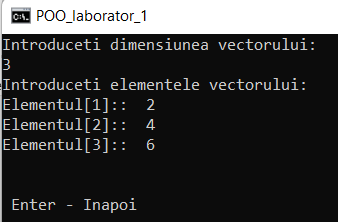
}

///////////////////////////////////////////////////////////

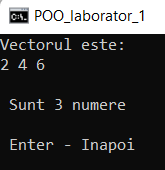
1.Menu



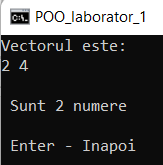
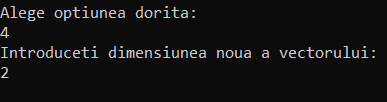
2.Initializarea vectorului



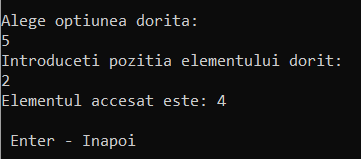
3. Afisarea elementelor



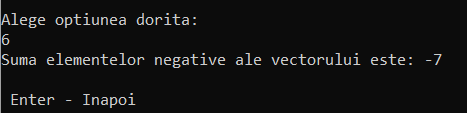
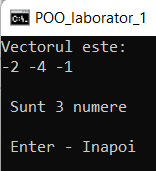
4. Modificarea dimensiunii vectorului



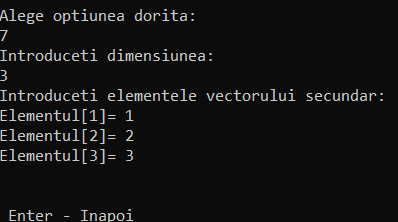
5. Accesarea elementelor



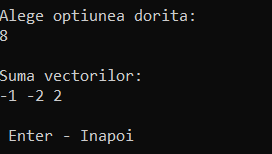
6.Suma elementelor negative



7.Crearea vectorului secundar



8. Suma a doi vectori



**Concluzie**

In lucrarea respectiva am realizat tema ,,Structura-mecanism de abstractizare’’. In urma lucrarii am studiat programarea prin abstractizarea datelor, am studiat regulile de definire si utilizare a structurilor de date si am creat variabile de tip structura, precum si accesarea componentelor structurii respective. Deci, limbajul de programare C++, este foarte comod si liber de inteles, pentru a lucra cu datele, cu structuri sau orice variabile. Limbajul respectiv ne permite rapid si usor sa organizam informatia.