

**MINISTERUL EDUCAȚIEI, CULTURII ȘI CERCETĂRII**

**AL REPUBLICII MOLDOVA**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică şi Microelectronică**

**Departamentul Ingineria Software și Automatica**

**Cucerenco Maxim**

**Grupa: SI-191**

**Raport**

**pentru lucrarea de laborator Nr. 1**

***la cursul de “Programarea orientată pe obiecte”***

Verificat:

**Mititelu Vitalii**

**Chișinău – 2020**

Sarcina:

а) Să se creeze tipul abstract de date (structura) – produs soft (software), care are denumire, firma de producere, anul editării şi versiunea. Să se definească funcţiile de setare, de modificare a datelor, de comparare. Pentru crearea cîmpurilor textuale să se utilizeze operatorul new. Să se elibereze memoria. În main să se exemplifice căutarea produsului necesar după cîteva criterii.

b) Să se creeze tipul abstract de date (structura) – vector, care are pointer spre double 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 pare ale vectorului. Ca exemplu, în funcţia main, să se realizeze adunarea unui vector cu un număr

**a)**

**Codul**

#include <iostream>

#include <string>

#include <conio.h>

#include <cstring>

#include <sstream>

using namespace std ;

struct Software

{

char \*denumire;

char \*Firma;

int anul;

float versiunea;

void SetDenumire(const char\*str);

void SetFirma(const char\*str);

void set\_anul(int a);

void set\_versiunea(float v);

// void comparare()

};

int main()

{

int i,a,min,memi;

float v;

char c[15];

Software sof[3];

for (i=0; i<3;i++)

{

cout<<"Introduceti Denumirea Softului "<<endl;

cin >> c;

sof[i].SetDenumire(c);

};

for (i=0; i<3;i++)

{

cout<<"Introduceti firma Softului "<<sof[i].denumire<<endl;

cin >> c;

sof[i].SetFirma(c);

};

for (i=0; i<3;i++)

{

cout<<"Introduceti anul producerii Softului "<<sof[i].denumire<<endl;

cin >> a;

sof[i].set\_anul(a);

};

for (i=0; i<3;i++)

{

cout<<"Introduceti Versiunea Softului "<<sof[i].denumire<<endl;

cin >> v;

sof[i].set\_versiunea(v);

};

for (i=0; i<3;i++)

{

cout<<"Softul - "<<sof[i].denumire<< " a fost produs in anul: "<<sof[i].anul<<"de catre: "<<sof[i].Firma<<" si la moment are versiunea= "<<sof[i].versiunea;

cout<< endl;

};

min=sof[0].anul;

for (i=0; i<3 ;i++)

{

if(sof[i].anul<min)

{

memi=i;

min=sof[i].anul;

}

}

cout<<"Primul din aceste softuri este: "<<sof[memi].denumire<<" si a fost fabricat in anul: "<< min;

return 0;

}

void Software::SetDenumire(const char\*str)

{

denumire = new char[strlen(str)+1];

strcpy(denumire, str);

}

void Software::SetFirma(const char\*str)

{

Firma = new char[strlen(str)+1];

strcpy(Firma, str);

}

void Software::set\_anul(int a)

{

anul=a;

}

void Software::set\_versiunea(float v)

{

versiunea=v;

}

**Output-ul:**



**b)**

**Codul**

#include <iostream>

#include <cstring>

#include <cstddef>

using namespace std;

struct Vector{

int size;

unsigned \*arr;

};

void init(Vector &v, int n, bool keyb=true ){

v.size = n;

v.arr = new unsigned[v.size];

if(keyb){

cout << "Introduceti "<<n<<" numere: ";

for(int i = 0; i < v.size; i++)

cin >> v.arr[i];

}

}

void elim(Vector &v){

if(v.arr){

delete[]v.arr;

v.arr = NULL;

v.size=0;

}

}

void afis(Vector const &v){

if(v.size == 0) cout << "Vectorul este vid" << endl;

else {

cout << "Vectorul are " << v.size << " elemente:";

for(int i = 0;i < v.size; i++) cout <<" " << v.arr[i] << " ";

cout << endl;

}

}

unsigned elem(Vector const &v, int i){

if(i > 0 && i <= v.size) return v.arr[i-1];

else{

cout << "Element inexistent" << endl;

return 0;

}

}

void resize(Vector &v, int newSize){

if(newSize == v.size){

cout << "Vectorul are aceeasi marime" << endl;

}

else{

Vector temp;

int oldSize = v.size, i;

init(v, oldSize, false);

for(i = 0; i < oldSize; i++) temp.arr[i] = v.arr[i];

elim(v);

init(v, newSize, false);

if(newSize < oldSize){

for(i = 0; i < newSize; i++)

v.arr[i] = temp.arr[i];

cout << "In vectorul v sunt cu "<< (oldSize - newSize) <<" elem mai putine" << endl;

}

else{

for(i=0; i < oldSize; i++)

v.arr[i] = temp.arr[i];

for(i = oldSize; i < newSize; i++)

v.arr[i]=0;

cout << "In vector sunt cu " << (newSize - oldSize) << " elem mai mult" << endl;

}

afis(v);

}

}

void Adunare(Vector &v1, Vector &v2){

int temp[v1.size+1];

for(int i = 0; i < v1.size; i++){

if(v2.arr[i] == -1) v2.arr[i] = 0;

temp[i] = v1.arr[i] + v2.arr[i];

}

cout << "Vectorul 1 adunat cu vectorul d2: ";

for(int i = 0 ;i < v1.size; i++) cout << temp[i] << " ";

cout << endl;

}

void Comparare(Vector &v1, Vector &v2){

if(v1.size > v2.size) cout << "Vectorul 1 este mai mare decat vectorul 2" << endl;

if(v1.size < v2.size) cout << "Vectorul 2 este mai mare decat vectorul 1" << endl;

if (v1.size==v2.size) cout << "Vectorii sunt egali" << endl;

}

int main(){

Vector v1, v2;

init(v1,5);

init(v2,5);

afis(v1);

afis(v2);

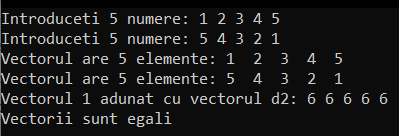
Adunare(v1,v2);

Comparare(v1, v2);

return 0;

}

**Output-ul:**



**Concluzie:**

Efectuand aceasta lucrare de laborator m-am aprofundat în structura limbajului de programare C++, am găsit diferențele dintre imbajul C și C++. Am învățat a crea clase și lucra cu obiecte, cu ajutorul a multimii de guide-uri si desigur a platformei YouTube.