** Ministerul Educaţiei Republicii**

**Moldovei**

**Universitatea Tehnică a Moldovei**

Catedra: Calculatoare

**Raport**

Lucrare de laborator nr.4

## Tema: Mostenirea si compozitia

Varianta 8

A efectuat: Nicolenco Eugeniu, Gr. C-162

A verificat: Lector.Univ. Constantin Plotnic

2017

**Scopul lucrării:**

* studierea moştenirii, avantajele şi dezavantajele;
* studierea compoziţiei;
* studierea regulilor de definire a moştenirii şi compoziţiei;
* studierea formelor de moştenire;
* studierea iniţializatorilor;
* principiul de substituţie;
* moştenirea şi compoziţia – ce să alegem.
* Varianta 8
* а) De creat o ierarhie a claselor *clădire de învăţământ* clasa de bază şi clădirea preşcolară, medie şi superioară – clase derivate. Determinaţi fluxul de ieşire şi fluxul de intrare, constructorul de copiere, operatorul de atribuire prin funcţia corespunzătoare a clasei de bază.
* b) De creat clasa *motor,* care are o firmă producătoare, tip şi putere. Determinaţi funcţiile de definire, schimbare a parametrilor motoarelor. Creaţi o ierarhie a claselor: corabie – clasa de bază şi vapor pentru pasageri – derivată. Corabia are un motor, capacitate de încărcare, măsurător de apă, denumire, portul unde este înscris. Pentru sarcina cîmpurilor textuale, utilizaţi operatorul new.

**Programa cod:**

#include <iostream>

#include <string.h>

using namespace std;

class cladire{

public:

char adresa[20];

int sala;

int buget;

cladire(){

strcpy(adresa,"chisinau");

sala=1;

buget=1000;

}

cladire(char \*a,int s, int b){

strcpy(adresa,a);

sala=s;

buget=b;

}

cladire(cladire &a){

strcpy(adresa,a.adresa);

sala=a.sala;

buget=a.buget;

}

void Afisare(){

// cout<<"\t\t\tDatele despre cladire\n";

cout<<"Adresa: "<<adresa<<endl;

cout<<"Nr de sali: "<<sala<<endl;

cout<<"Bugetul institutiei :"<<buget<<" lei"<<endl;

}

void Init(){

cout<<"Dati Adresa cladirii: "; cin>>adresa;

cout<<"Cite sali are cladirea: "; cin>>sala;

cout<<"Care este bugetul cladirii: "; cin>>buget;

}

cladire& operator=(cladire &a){

strcpy(adresa,a.adresa);

sala=a.sala;

buget=a.buget;

return \*this;

}

};

class prescolar:public cladire{

int ani;

char denumire[20];

public:

prescolar(){

ani=3;

strcpy(denumire,"gradinita");

}

prescolar(char \*a,int s, int b, int an, char \*nn):cladire(a,s,b){

ani=an;

strcpy(denumire,nn);

}

prescolar(prescolar &ap,char \*a,int s,int b):cladire(a,s,b){

ani=ap.ani;

strcpy(denumire,ap.denumire);

}

void Afisare(){

cladire::Afisare();

// cout<<"\t\tDate despre institutie prescolara:\n";

cout<<"\tInstitutia prescolara: "<<denumire<<endl;

cout<<"\tDurata: "<<ani<<endl;

}

void Init(){

cladire::Init();

cout<<"Dati numele institutiei prescolare: "; cin>>denumire;

cout<<"Dati durata studiilor: "; cin>>ani;

}

prescolar operator=(prescolar &c){

strcpy(denumire,c.denumire);

ani=c.ani;

strcpy(adresa,c.adresa);

sala=c.sala;

buget=c.buget;

return \*this;

}

};

class medie:public cladire{

int absolvit;

char denumire[20];

public:

medie(){

absolvit=2015;

strcpy(denumire,"Scoala medie");

}

medie(char \*a,int s, int b,int ab,char \*d):cladire(a,s,b){

strcpy(denumire,d);

absolvit=ab;

}

medie(medie &A,char \*a,int s,int b):cladire(a,s,b){

strcpy(denumire,A.denumire);

absolvit=A.absolvit;

}

void Afisare(){

cladire::Afisare();

// cout<<"\t\tDate despre institutie medie:\n";

cout<<"\tScoala medie: "<<denumire;

cout<<endl<<"\tAbsolvit in anul: "<<absolvit<<endl;

}

void Init(){

cladire::Init();

cout<<"Dati denumirea scolii: "; cin>>denumire;

cout<<"Absolvit in anul: "; cin>>absolvit;

}

medie& operator=(medie &c){

strcpy(denumire,c.denumire);

absolvit=c.absolvit;

strcpy(adresa,c.adresa);

sala=c.sala;

buget=c.buget;

return \*this;

}

};

class superioara:public cladire{

int ciclu;

char nume[20];

char facultate[20];

public:

superioara(){

ciclu=1;

strcpy(nume,"UTM");

strcpy(facultate,"FCIM");

}

superioara(char \*a,int s, int b,int c,char \*n,char \*f):cladire(a,s,b){

ciclu=c;

strcpy(nume,n);

strcpy(facultate,f);

}

superioara(superioara &A,char \*a,int s,int b):cladire(a,s,b){

ciclu=A.ciclu;

strcpy(nume,A.nume);

strcpy(facultate,A.facultate);

}

void Afisare(){

cladire::Afisare();

// cout<<"\t\tDate despre institutie superioara:\n";

cout<<"\tUniversitatea :"<<nume;

cout<<endl<<"\tFacultatea :"<<facultate;

cout<<endl<<"\tAnul :"<<ciclu;

cout<<endl;

}

void Init(){

cladire::Init();

cout<<"Numele universitatii :"; cin>>nume;

cout<<"Facultatea :"; cin>>facultate;

cout<<"Anul :"; cin>>ciclu;

}

superioara operator=(superioara &c){

strcpy(nume,c.nume);

strcpy(facultate,c.facultate);

ciclu=c.ciclu;

strcpy(adresa,c.adresa);

sala=c.sala;

buget=c.buget;

return \*this;

}

};

int main(){

prescolar p1("Ismail 10",6,24000,3,"Licurici");

prescolar p2;

// p2.Init();

cout<<endl;

p1.Afisare();

cout<<endl;

p2.Afisare();

cout<<endl;

medie m1("Nicolae Titulescu 18",52,154000,2012,"Iulia Hasdeu");

medie m2;

// m2.Init();

cout<<endl;

m1.Afisare();

m2.Afisare();

superioara s1("Studentilor 9/1",112,1154000,2,"UTM","CIM");

superioara s2;

//s2.Init();

cout<<endl;

s1.Afisare();

cout<<endl;

s2.Afisare();

medie m3(m1);

cout<<endl;

m3.Afisare();

cout<<"\n\n";

prescolar p3(p1);

cout<<endl;

p3.Afisare();

cladire c1("Independentei 10/2",15,11000);

cladire c2;

c2=c1;

cout<<endl;

c1.Afisare();

cout<<endl;

c2.Afisare();

cout<<endl;

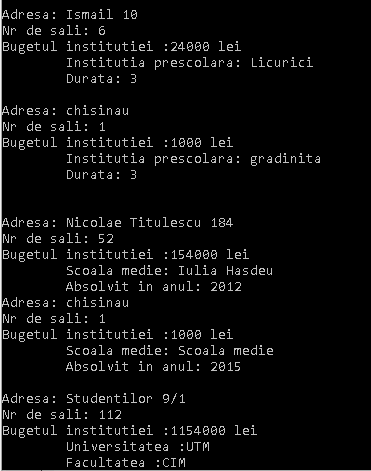
medie m4=m1;

m4.Afisare();

return 0;

}

**Rezultate afisate:**



Partea b)

**Programa cod:**

#include <iostream>

#include <string.h>

#define q endl

using namespace std;

class motor{

protected:

char \*fp=new char[20];

char \*tip=new char[20];

int putere;

public:

motor(){

strcpy(fp,"Maybach");

strcpy(tip,"Diesel");

putere=1500;

}

motor(char \*firma,char \*t,int p){

strcpy(fp,firma);

strcpy(tip,t);

putere=p;

}

virtual void init()=0;

virtual void show()=0;

void cfp(char \*firma){

strcpy(fp,firma);

}

void ct(char \*t){

strcpy(tip,t);

}

void cp(int p){

putere=p;

}

~motor(){};

};

class corabie:public motor{

protected:

int capacitate;

int masurator;

char \*nume=new char[20];

char \*port=new char[20];

public:

corabie(){

strcpy(nume,"Speranta");

strcpy(port,"Santa-Monica");

masurator=1000;

capacitate=225;

};

corabie(char \*firma,char \*t,int p):motor(firma,t,p){

strcpy(nume,"Speranta");

strcpy(port,"Santa-Monica");

masurator=1000;

capacitate=225;

}

corabie(char \*n,char \*po,int c,int m,char \*firma,char \*t,int p):motor(firma,t,p){

strcpy(nume,n);

strcpy(port,po);

masurator=m;

capacitate=c;

}

void init(){

cout<<"\nDati parametrii motorului\n";

cout<<"Dati firma producatoare: "; cin>>fp;

cout<<"Dati tipul motorului: "; cin>>tip;

cout<<"Cai putere :"; cin>>putere;

cout<<"\nDati parametrii corabiei\n";

cout<<"Dati numele: "; cin>>nume;

cout<<"Dati portul nasterii: "; cin>>port;

cout<<"Dati masuratorul apei: "; cin>>masurator;

cout<<"Dati capacitatea: "; cin>>capacitate;

}

void show(){

cout<<"\nParametrii motorului\n";

cout<<"Firma producatoare: "<<fp<<q;

cout<<"Tipul motorului: "<<tip<<q;

cout<<"Cai putere :"<<putere<<q;

cout<<"\nParametrii corabiei\n";

cout<<"Numele: "<<nume<<q;

cout<<"Portul nasterii: "<<port<<q;

cout<<"Masuratorul apei: "<<masurator<<q;

cout<<"Capacitatea: "<<capacitate<<"Oameni"<<q;

}

~corabie(){

};

};

class vapor:public corabie{

int viteza;

int lungime;

public:

vapor(){

}

vapor(char \*n,char \*po,int c,int m,char \*firma,char \*t,int p):corabie(n,po,c,m,firma,t,p){

viteza=10;

lungime=100;

};

vapor(int v, int l,char \*n,char \*po,int c,int m,char \*firma,char \*t,int p):corabie(n,po,c,m,firma,t,p){

viteza=v;

lungime=l;

}

void init(){

corabie::init();

cout<<"\nDati parametrii vaporului\n";

cout<<"Dati viteza vaporului: "; cin>>viteza;

cout<<"Dati lungimea vaporului: "; cin>>lungime;

}

void show(){

corabie::show();

cout<<"\nParametri vaporului\n";

cout<<"Viteza: "<<viteza<<q;

cout<<"Lungimea: "<<lungime<<"metri"<<q;

cout<<"###############################";

}

~vapor(){};

};

int main(){

corabie c1;

c1.show();

cout<<q<<"########################################";

vapor v1;

v1.init();

vapor v2(20,3000,"Titanic","San-Torino",2500,1300,"Mitsubishi","diesel",13000);

v1.show();

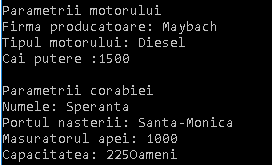
v2.show();

return 0;

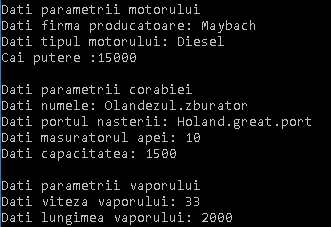
}

**Rezultate afisate:**

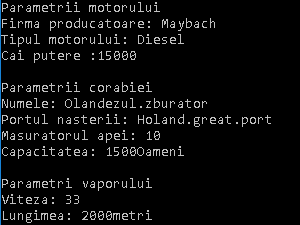
Afisarea corabiei construita in mod implicit



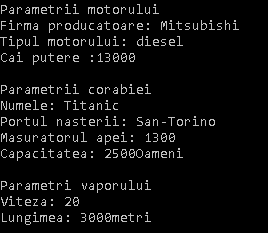
Initializarea unui vapor (creat implicit) prin functie



Afisarea acestui vapor;



Afisarea vaporului creat implicit;



**Concluzie:** In urma efectuarii laboratorului dat m-am familiarizat cu mostenirea si compozitia, de asemenea am utilizat functii abstrace pure si am inteles utilitatea lor. Lucrarea a fost usor de inteles si interesant de indeplinit. Lucrul cu obiectele devine mult mai usor utilizand constructori derivati si mostenirea multor functii din alte clase.