

Ministerul Educației, Culturii și Cercetării al Republicii Moldova
Universitatea Tehnică a Moldovei
Facultatea Calculatoare, Informatică și Microelectronică

Raport la

Lucrarea de laborator Nr. 5

Disciplina: Programare orientata pe obiecte

Tema: ” Moștenirea multiplă”

Au efectuat:

Seremet Iulian, TI-195

A verificat:

lector universitar: V.Mititelu

Chișinău – 2020

Scopul lucrării:

- Studierea regulilor de determinare a moștenirii multiple;
- Studierea avantajelor și neajunsurilor moștenirii multiple;
- Probleme legate de utilizarea moștenirii multiple;
- Studierea rezolvării problemelor.

Varianta 5

a) Să se creeze, o ierarhie de moștenire: barca cu motor, motocicletă – motocicletă de apă.

b) Să se creeze, o ierarhie de moștenire: transport – barca cu motor, motocicletă – motocicletă de apă.

a) `#include <iostream>`

`#include <string.h>`

`#include <conio.h>`

`#include <stdio.h>`

`using namespace std;`

`class barca_cu_motor{`

`public: //declararea variabilelor ca fiind publice, pentru a pute accesa din alta parte a
program.`

`char *motor; //denumirea motorului`

`float hp; //cai putere`

```

char *denumire;          //denumirea
float masa;              //masa (kg)

barca_cu_motor():motor(NULL),hp(0),denumire(NULL),masa(0){}; //CONSTR. IMPLICIT
barca_cu_motor(char *denumire,int *hp,char *motor,int masa){ // CONSTR. EXPLICIT
    motor = new char; motor = new char[strlen(motor)+1];
        strcpy(motor,motor); //alocarea de memorie p/u un char
    hp = hp;
    denumire = new char[strlen(denumire)+1]; //alocare p/u denumire
        strcpy(denumire,denumire);          //copie din denum in denumire
        masa=masa;
};

barca_cu_motor(barca_cu_motor &obj){ //CONSTR. DE COPIERE
    motor = new char;    strcpy(motor,obj.motor);
    hp=obj.hp;
    denumire = new char[strlen(denumire)+1]; strcpy(denumire,obj.denumire);
    masa=obj.masa;
};

~barca_cu_motor(){          // DESTUCTOR
delete motor; motor=NULL;    //sterge pointerul mot;
        hp=0;
        delete denumire; denumire=NULL;
        masa=0;          //masa i se atribuie (zero)
};

//supraincarcarea functiilor „prietene” de intrare/iesire:
friend ostream & operator<<(ostream &, barca_cu_motor &);
friend istream & operator>>(istream &, barca_cu_motor &);

```

```

};

//*****SUPRAINCARCAREA OPERATORULUI COUT <<*****class barca_cu_motor*****
ostream & operator<<(ostream &out, barca_cu_motor &obj){
    out<<"Denumirea: "<<obj.denumire<<endl;
    out<<"Cai puteri: "<<obj.hp<<endl;
    out<<"Motor: "<<obj.motor<<endl;
    out<<"Masa: "<<obj.masa<<endl;

    return out;
};

istream & operator>>(istream &in, barca_cu_motor &obj){
    char buf[100];
    cout<<"Denumire: ";   in>>buf;
    obj.denumire = new char [strlen(buf)+1];
    strcpy(obj.denumire,buf);
    cout<<"Cai puteri: "; in>>obj.hp;
    cout<<"Motor: ";     obj.motor = new char;
    in>>obj.motor;
    cout<<"Masa: "; in>>obj.masa;
    return in;
};

class motocicletă: public barca_cu_motor{
    char *model;
public:
    motocicletă(){model=NULL; };
    motocicletă(char *den,int hp,char *motor,int masa,char *model){

```

```

        barca_cu_motor::denumire = new char[strlen(den)+1];
        strcpy(barca_cu_motor::denumire,den);
        barca_cu_motor::hp = hp;
        barca_cu_motor::motor = new char;
        strcpy(barca_cu_motor::motor,motor);
        barca_cu_motor::masa = masa;
        model = new char[strlen(model)+1];
        strcpy(model,model);
    };

friend istream & operator>>(istream &in, motocicleta &obj);
friend ostream & operator<<(ostream &out,motocicleta &obj);
};

//*****SUPRAINCARCAREA OPERATOR DE INTRARE CIN >>
istream & operator>>(istream &in, motocicleta &obj){
char buf[50];

cout<<"Denumirea: "; in>>buf;

obj.denumire = new char[strlen(buf)+1];
strcpy(obj.denumire,buf);

cout<<"hp c/p: "; in>>obj.hp;

cout<<"Motorul: "; in>>buf;

obj.motor = new char[strlen(buf)+1];
strcpy(obj.motor,buf);

cout<<"Greutatea (kg): "; in>>obj.masa;

cout<<"Modelul: "; in>>buf;

obj.model = new char[strlen(buf)+1];
strcpy(obj.model,buf);

return in; };

```

```

//*****SUPRAINCARCAREA OPERATOR DE IESIRE COUT <<
ostream & operator<<(ostream &out, motocicleta &obj){

    out<<"Denumire:   "; out<<obj.denumire<<endl;
    out<<"hp:         "; out<<obj.hp<<endl;
    out<<"Motorul:     "; out<<obj.motor<<endl;
    out<<"Greutatea:   "; out<<obj.masa<<endl;
    out<<"Modelul:     "; out<<obj.model<<endl;

    return out;
};

class motocicleta_de_apă: public barca_cu_motor{
float viteza;

public:
    motocicleta_de_apă(){viteza=0;};
    motocicleta_de_apă(char *den,int hp,char *motor,int masa, int viteza){
        barca_cu_motor::denumire = new char[strlen(den)+1];
        strcpy(barca_cu_motor::denumire,den);
        barca_cu_motor::hp = hp;
        barca_cu_motor::motor = new char;
        strcpy(barca_cu_motor::motor,motor);
        barca_cu_motor::masa = masa;
        viteza = viteza;
    };

friend istream & operator>>(istream &, motocicleta_de_apă &);
friend ostream & operator<<(ostream &, motocicleta_de_apă &);

```

```

};

istream & operator>>(istream &in, motocicleta_de_apa &obj){
char buf[50];

cout<<"Denumirea:   "; in>>buf;

    obj.denumire = new char[strlen(buf)+1];

        strcpy(obj.denumire,buf);

cout<<"hp:   "; in>>obj.hp;

cout<<"Motorul:   "; in>>buf;

    obj.motor = new char;

        strcpy(obj.motor,buf);

cout<<"Greutatea (kg): "; in>>obj.masa;

cout<<"Viteza max pe apa:"; in>>obj.viteza;


return in;

};

ostream & operator<<(ostream &out,motocicleta_de_apa &obj){

    out<<"Denumirea:   "; out<<obj.denumire<<endl;

    out<<"hp:           "; out<<obj.hp<<endl;

    out<<"Motor:         "; out<<obj.motor<<endl;

    out<<"Greutatea:     "; out<<obj.masa<<endl;

    out<<"Viteza max pe apa:"; out<<obj.viteza<<endl;

return out;

};

int main(){

```

```

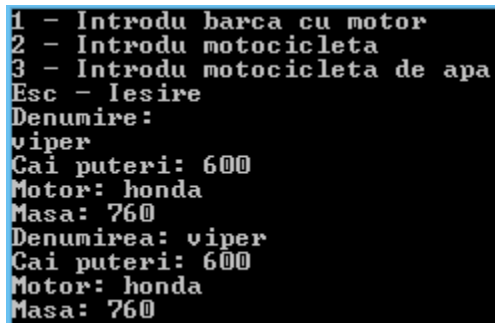
cout<<"1 "; cout<<"- Introdu barca cu motor  "<<endl;
cout<<"2 "; cout<<"- Introdu motocicleta  "<<endl;
cout<<"3 "; cout<<"- Introdu motocicleta de apa"<<endl;
cout<<"Esc "; cout<<"- Iesire"<<endl;
switch (getch()) {
    case '1': {    barca_cu_motor a; cin>>a; cout<<a; getch();    break;};
    case '2': {    motocicleta b; cin>>b;  cout<<b; getch();    break;};
    case '3': {    motocicleta_de_apa c;  cin>>c; cout<<c; getch();    break;};

}

getch();}

```

Rezultatul:



```

1 - Introdu barca cu motor
2 - Introdu motocicleta
3 - Introdu motocicleta de apa
Esc - Iesire
Denumire:
viper
Cai puteri: 600
Motor: honda
Masa: 760
Denumirea: viper
Cai puteri: 600
Motor: honda
Masa: 760

```

b)

```

#include <iostream>

#include <string.h>

#include <conio.h>

```



```

#include <stdio.h>

using namespace std;

class transport{
public:
char *motor;
float hp;
char *denumire;
float masa;

transport ():motor(NULL),hp(0),denumire(NULL),masa(0){};

transport (char *denumire,int *hp,char *motor,int masa){
    motor = new char; motor = new char[strlen(motor)+1];
        strcpy(motor,motor);
    hp = hp;
    denumire = new char[strlen(denumire)+1];
        strcpy(denumire,denumire);
        masa=masa;
};

transport(transport &obj){
    motor = new char;    strcpy(motor,obj.motor);
    hp=obj.hp;
    denumire = new char[strlen(denumire)+1]; strcpy(denumire,obj.denumire);
    masa=obj.masa;
};

~transport(){
delete motor; motor=NULL;

```

```
        hp=0;
        delete denumire; denumire=NULL;
        masa=0;
    };
```

```
friend ostream & operator<<(ostream &, transport &);
```

```
friend istream & operator>>(istream &, transport &);
```

```
};
```

```
ostream & operator<<(ostream &out, transport &obj){
```

```
    out<<"Denumirea: "<<obj.denumire<<endl;
```

```
    out<<"Cai puteri: "<<obj.hp<<endl;
```

```
    out<<"Motor: "<<obj.motor<<endl;
```

```
    out<<"Masa: "<<obj.masa<<endl;
```

```
    return out;
```

```
};
```

```
istream & operator>>(istream &in, transport &obj){
```

```
    char buf[100];
```

```
    cout<<"Denumire: ";  in>>buf;
```

```
    obj.denumire = new char [strlen(buf)+1];
```

```
    strcpy(obj.denumire,buf);
```

```
    cout<<"Cai puteri: "; in>>obj.hp;
```

```
    cout<<"Motor: ";    obj.motor = new char;
```

```
    in>>obj.motor;
```

```
    cout<<"Masa: "; in>>obj.masa;
```

```

    return in;
};

class barca_cu_motor: public transport{
char *col;
public:
    barca_cu_motor(){ col=NULL; };
    barca_cu_motor(char *den,int hp,char *motor,int masa,char *col){
transport::denumire = new char[strlen(den)+1];
        strcpy(transport::denumire,den);
transport::hp = hp;
transport::motor = new char;
        strcpy(transport::motor,motor);
transport::masa=masa;
col = new char[strlen(col)+1];
        strcpy(col,col);
    };
friend istream & operator>>(istream &in, barca_cu_motor &obj);
friend ostream & operator<<(ostream &out,barca_cu_motor &obj);
};

```

```

istream & operator>>(istream &in, barca_cu_motor & obj){
char buf[50];
cout<<"Denumirea: "; in>>buf;
obj.denumire = new char[strlen(buf)+1];
    strcpy(obj.denumire,buf);
    cout<<"hp: "; in>>obj.hp;
}

```

```

    cout<<"motor:    ";   in>>buf;
obj.motor = new char;
    strcpy(obj.motor,buf);
    cout<<"Greutatea:  ";   in>>obj.masa;
    cout<<"Culoarea : ";   in>>buf;
obj.col = new char[strlen(buf)+1];
    strcpy(obj.col,buf);
return in;
};

ostream & operator<<(ostream &out,barca_cu_motor &obj){
    out<<"Denumirea:  ";      out<<obj.denumire<<endl;
    out<<"hp:        ";      out<<obj.hp<<endl;
    out<<"Motorul:    ";      out<<obj.motor<<endl;
    out<<"Greutate kg:  ";      out<<obj.masa<<endl;
    out<<"Culoarea:   ";      out<<obj.col<<endl;

return out;
};

class motocicletă: public transport{
char *model;
public:
    motocicletă(){model=NULL; };
    motocicletă(char *den,int hp,char *motor,int masa,char *model){
        transport::denumire = new char[strlen(den)+1];
        strcpy(transport::denumire,den);
        transport::hp = hp;
        transport::motor = new char;
    }
};

```

```

        strcpy(transport::motor,motor);
transport::masa = masa;
model = new char[strlen(model)+1];
strcpy(model,model);
    };

friend istream & operator>>(istream &in, motocicleta &obj);
friend ostream & operator<<(ostream &out,motocicleta &obj);
};

```

```

istream & operator>>(istream &in, motocicleta &obj){
char buf[50];
cout<<"Denumirea: "; in>>buf;
obj.denumire = new char[strlen(buf)+1];
strcpy(obj.denumire,buf);
cout<<"hp c/p: "; in>>obj.hp;
cout<<"Motorul: "; in>>buf;
obj.motor = new char[strlen(buf)+1];
strcpy(obj.motor,buf);
cout<<"Greutatea (kg): "; in>>obj.masa;
cout<<"Modelul: "; in>>buf;
obj.model = new char[strlen(buf)+1];
strcpy(obj.model,buf);
return in; };

```

```

ostream & operator<<(ostream &out, motocicleta &obj){

```

```

out<<"Denumire:   "; out<<obj.denumire<<endl;
out<<"hp:         "; out<<obj.hp<<endl;
out<<"Motorul:     "; out<<obj.motor<<endl;
out<<"Greutatea:   "; out<<obj.masa<<endl;
out<<"Modelul:     "; out<<obj.model<<endl;

return out;

};

class motocicletă_de_apa: public transport{
float viteza;

public:

    motocicletă_de_apa(){viteza=0;};

    motocicletă_de_apa(char *den,int hp,char *motor,int masa, int viteza){
        transport::denumire = new char[strlen(den)+1];
        strcpy(transport::denumire,den);

        transport::hp = hp;

        transport::motor = new char;
        strcpy(transport::motor,motor);

        transport::masa = masa;

        viteza = viteza;

    };

friend istream & operator>>(istream &, motocicletă_de_apa &);
friend ostream & operator<<(ostream &, motocicletă_de_apa &);

};

istream & operator>>(istream &in, motocicletă_de_apa &obj){
char buf[50];

cout<<"Denumirea:   "; in>>buf;

```

```

    obj.denumire = new char[strlen(buf)+1];
        strcpy(obj.denumire,buf);
cout<<"hp: "; in>>obj.hp;
cout<<"Motorul: "; in>>buf;
    obj.motor = new char;
        strcpy(obj.motor,buf);
cout<<"Greutatea (kg): "; in>>obj.masa;
cout<<"Viteza max pe apa:"; in>>obj.viteza;


return in;
};

ostream & operator<<(ostream &out,motocicleta_de_apa &obj){

    out<<"Denumirea: "; out<<obj.denumire<<endl;
    out<<"hp: "; out<<obj.hp<<endl;
    out<<"Motor: "; out<<obj.motor<<endl;
    out<<"Greutatea: "; out<<obj.masa<<endl;
    out<<"Viteza max pe apa:"; out<<obj.viteza<<endl;
return out;
};

int main(){
    cout<<"1 "; cout<<"- Introdu transportul "<<endl;
    cout<<"2 "; cout<<"- Introdu barca cu motor "<<endl;
    cout<<"3 "; cout<<"- Introdu motocicleta "<<endl;
    cout<<"4 "; cout<<"- Introdu motocicleta de apa"<<endl;
    cout<<"Esc "; cout<<"- Iesire"<<endl;

```

```

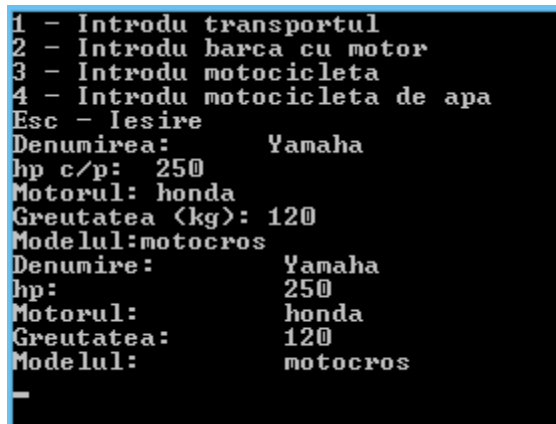
switch (getch()) {
    case '1': {    transport a; cin>>a; cout<<a; getch();    break;};
    case '2': {    barca_cu_motor b; cin>>b; cout<<b; getch();    break;};
    case '3': {    motocicleta c; cin>>c;    cout<<c; getch();    break;};
    case '4': {    motocicleta_de_apa d;  cin>>d; cout<<d; getch();    break;};

}

getch();}

```

Rezultatul:



```

1 - Introdu transportul
2 - Introdu barca cu motor
3 - Introdu motocicleta
4 - Introdu motocicleta de apa
Esc - Iesire
Denumirea:      Yamaha
hp c/p: 250
Motorul: honda
Greutatea (kg): 120
Modelul: motocros
Denumire:      Yamaha
hp:           250
Motorul:      honda
Greutatea:    120
Modelul:      motocros
_

```

Concluzi:

În urma efectuării lucrării date am determinat că mediul de programare C++ este cu mult flexibil ca C, care ne permite să manipulăm cu informația într-un mod rapid și eficient fără a induce erori grosolane, și am aflat că utilizarea claselor este cu mult mai comodă și mai rapidă în elaborarea unui program.

Am aflat că moștenirea multiplă, reprezintă prin sine moștenirea de la două sau mai multe clase, și că acest mecanism este foarte necesar. Dar, el nu se folosește în toate limbajele, dar este realizat în C++.