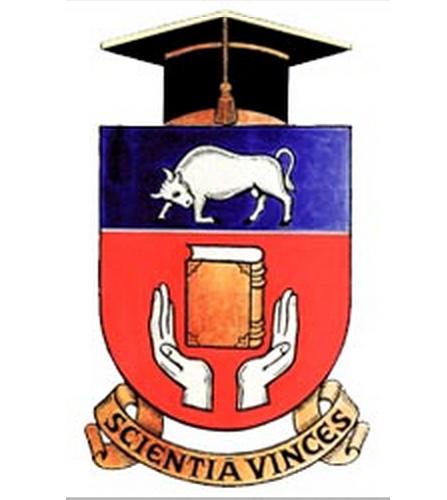
Universitatea de Stat din Tiraspol

Facultatea de Fizica Matematica si Informatica



Laboratorul #3

**Elaborat de**: Dordea Pavel, student gr.2i

**Verificat**: dr.conf.univ. Ala Gasnaș

**Chișinău, 2021**

**2. Constructori**

*Clasă-pereche* este numită o clasă cu două date, care de obicei sunt două numere *n1* şi *n2*. Este necesar de a elabora un program, prin intermediul căruia se va crea tipul de date *clasă-pereche*. În toate sarcinile trebuie să fie prezente:

1. Trei tipuri de constructori: fără parametri, cu parametri, de copiere;
2. Destructorul;
3. Metoda de citire;
4. Metoda de afisare;

Toate aceste condiţii necesită a fi respectate pentru următoarele cazuri.

* 1. Cîmpul *n1* - un număr întreg, Cîmpul *n2* – număr real nenul. Creaţi metoda

*power*() – va returna numărul *n2* la puterea *n1*.

**Codul C++ :**

#include <bits/stdc++.h>

using namespace std;

class MyClass{

int n1;

float n2;

public:

MyClass(){

n1 = 0;

n2 = 1;

}

MyClass(int a, float b){

n1 = a;

n2 = b;

}

MyClass(MyClass &);

void citire();

void afisare();

~ MyClass(){

cout << "";

}

float power();

};

void MyClass:: citire(){

cout << "Dati primul numar intreg: " << endl;

cin >> n1;

cout << "Dati numarul float: " << endl;

cin >> n2;

}

float MyClass::power(){

return pow(n2,n1);

}

MyClass::MyClass(MyClass & Object){

n1 = Object.n1;

n2 = Object.n2;

}

void MyClass::afisare(){

cout << "Iata primul numar: " << n1 << endl;

cout << "Iata al 2-lea numar: " << n2 << endl;

cout << "Numarul " << n2 << " la puterea " << n1 << ": " << power()<<endl;

}

main(){

MyClass nr;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "Constructor ce citeste valori" << endl;

nr.citire();

nr.afisare();

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "Constructor cu parametri" << endl;

MyClass nr2(7,2);

nr2.afisare();

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "Constructor de copiere" << endl;

MyClass Object(nr2);

nr2.afisare();

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "Construtor pointer" << endl;

MyClass \*n\_r;

n\_r = new MyClass;

n\_r -> afisare();

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

delete n\_r;

cout << "Constructor pointer cu parametri" << endl;

n\_r = new MyClass(7,4);

n\_r -> afisare();

delete n\_r;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

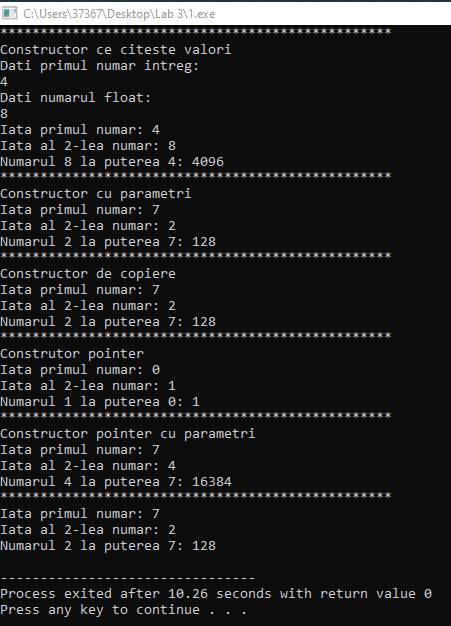
n\_r = new MyClass(nr2);

n\_r -> afisare();

delete n\_r;

}

**Screenshot-uri:**



* 1. Cîmpul *n1* - un număr întreg, Cîmpul *n2* – număr real. Creaţi metoda *intreg*() - care va returna partea întreagă a fracţiei *n1/n2*. Metoda trebuie să verifice inegalitatea numitorului cu zero.

**Codul C++ :**

#include <bits/stdc++.h>

using namespace std;

class MyClass{

int n1;

float n2;

public:

MyClass(){

n1 = 0;

n2 = 1;

}

MyClass(int a, float b){

n1 = a;

n2 = b;

}

MyClass(MyClass &);

void citire();

void afisare();

~ MyClass(){

cout << "";

}

int intreg();

};

void MyClass:: citire(){

cout << "Dati primul numar: " << endl;

cin >> n1;

cout << "Dati numarul: " << endl;

cin >> n2;

}

int MyClass::intreg(){

if(n2 != 0){

return (n1/n2);

} else cout << "Impartirea nu poate fi efectuata" << endl;

return 0;

}

MyClass::MyClass(MyClass & Object){

n1 = Object.n1;

n2 = Object.n2;

}

void MyClass::afisare(){

cout << "Iata primul numar: " << n1 << endl;

cout << "Iata al 2-lea numar: " << n2 << endl;

if(intreg() != 0){

cout << "Numarul " << n1 << " impartit la " << n2 << ": " << intreg()<<endl;

} else cout << "Impartirea nu are loc" << endl;

}

main(){

MyClass nr;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "Constructor ce citeste valori" << endl;

nr.citire();

nr.afisare();

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "Constructor cu parametri" << endl;

MyClass nr2(7,2);

nr2.afisare();

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "Constructor de copiere" << endl;

MyClass Object(nr2);

nr2.afisare();

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "Construtor pointer" << endl;

MyClass \*n\_r;

n\_r = new MyClass;

n\_r -> afisare();

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

delete n\_r;

cout << "Constructor pointer cu parametri" << endl;

n\_r = new MyClass(7,4);

n\_r -> afisare();

delete n\_r;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

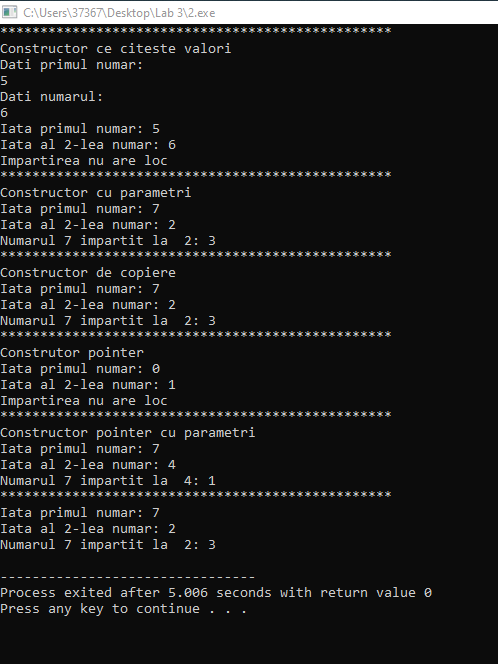
n\_r = new MyClass(nr2);

n\_r -> afisare();

delete n\_r;

}

**Screenshot-uri:**



* 1. Cîmpul *n1* - un număr real pozitiv (preţul bunurilor), Cîmpul *n2* – număr întreg pozitiv (numărul de bunuri). Creaţi metoda *cost*() – va returna numărul ce reprezintă valoarea totală a bunurilor.

**Codul C++ :**

#include <bits/stdc++.h>

using namespace std;

class MyClass{

float n1;

int n2;

public:

MyClass(){

n1 = 0;

n2 = 1;

}

MyClass(int a, float b){

n1 = a;

n2 = b;

}

MyClass(MyClass &);

void citire();

void afisare();

~ MyClass(){

cout << "";

}

float cost();

};

void MyClass:: citire(){

cout << "Dati pretul: " << endl;

cin >> n1;

cout << "Dati numarul bunurilor: " << endl;

cin >> n2;

}

float MyClass::cost(){

if(n1 > 0){

return (n1\*n2);

} else cout << "Pretul bunurilor nu este admisibil " << endl;

return 0;

}

MyClass::MyClass(MyClass & Object){

n1 = Object.n1;

n2 = Object.n2;

}

void MyClass::afisare(){

cout << "Iata pretul bunurilor: " << n1 << endl;

cout << "Iata numarul bunurilor: " << n2 << endl;

if(n1 > 0){

cout << "Costul este: " << cost() << endl;

} else cout << "Imposibil de a calcula costul" << endl;

}

main(){

MyClass nr;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "Constructor ce citeste valori" << endl;

nr.citire();

nr.afisare();

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "Constructor cu parametri" << endl;

MyClass nr2(7,2);

nr2.afisare();

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "Constructor de copiere" << endl;

MyClass Object(nr2);

nr2.afisare();

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "Construtor pointer" << endl;

MyClass \*n\_r;

n\_r = new MyClass;

n\_r -> afisare();

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

delete n\_r;

cout << "Constructor pointer cu parametri" << endl;

n\_r = new MyClass(7,4);

n\_r -> afisare();

delete n\_r;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

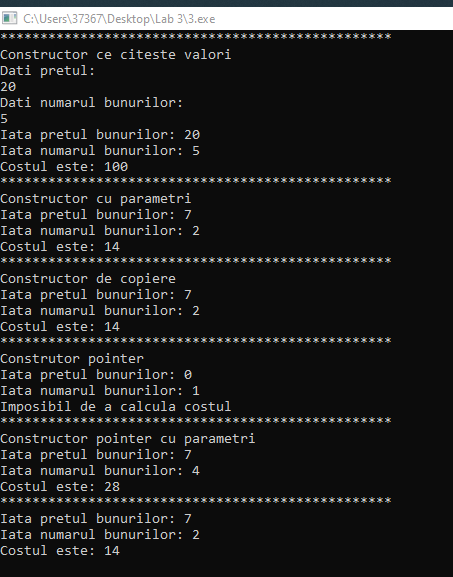
n\_r = new MyClass(nr2);

n\_r -> afisare();

delete n\_r;

}

**Screenshot-uri:**



* 1. Cîmpul *n1* - un număr întreg pozitiv (calorii la 100g), Cîmpul *n2* – număr real (greutatea produsului în kilograme). Creaţi metoda *calorii*() – va returna numărul de calorii ale produsului.

**Codul C++ :**

#include <bits/stdc++.h>

using namespace std;

class MyClass{

int n1;

float n2;

public:

MyClass(){

n1 = 0;

n2 = 1;

}

MyClass(int a, float b){

n1 = a;

n2 = b;

}

MyClass(MyClass &);

void citire();

void afisare();

~ MyClass(){

cout << "";

}

float calorii();

};

void MyClass:: citire(){

cout << "Dati nr de calorii la 100 grame: " << endl;

cin >> n1;

cout << "Dati greutatea produsului: " << endl;

cin >> n2;

}

float MyClass::calorii(){

if(n1 > 0){

return (n1 \* 10 \* n2);

} else cout << "Nr de calorii nu este pozitiv" << endl;

return 0;

}

MyClass::MyClass(MyClass & Object){

n1 = Object.n1;

n2 = Object.n2;

}

void MyClass::afisare(){

cout << "Iata nr de calorii: " << n1 << endl;

cout << "Iata greutatea produsului: " << n2 << endl;

if(calorii() != 0){

cout << "Nr de calorii : " << calorii() << endl;

} else cout << "Imposibil de a calcula nr de calorii" << endl;

}

main(){

MyClass nr;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "Constructor ce citeste valori" << endl;

nr.citire();

nr.afisare();

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "Constructor cu parametri" << endl;

MyClass nr2(7,2);

nr2.afisare();

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "Constructor de copiere" << endl;

MyClass Object(nr2);

nr2.afisare();

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "Construtor pointer" << endl;

MyClass \*n\_r;

n\_r = new MyClass;

n\_r -> afisare();

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

delete n\_r;

cout << "Constructor pointer cu parametri" << endl;

n\_r = new MyClass(7,4);

n\_r -> afisare();

delete n\_r;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

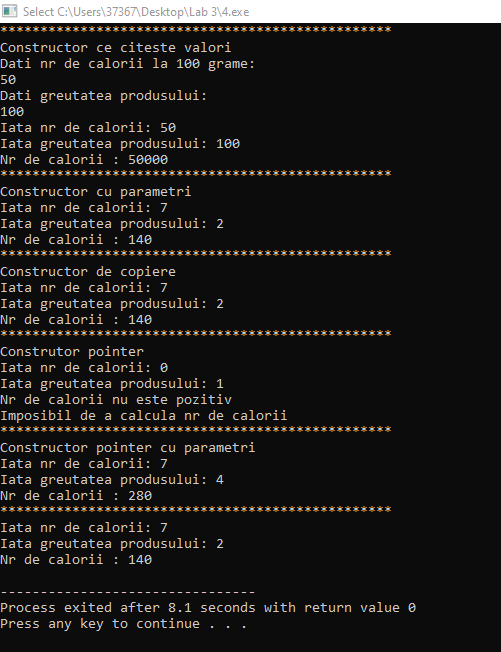
n\_r = new MyClass(nr2);

n\_r -> afisare();

delete n\_r;

}

**Screenshot-uri:**



* 1. Cîmpul *n1* - un număr real (suma depozitul în lei), Cîmpul *n2* – număr real (procentul anual). Creaţi metoda *valoare*(*int*) – va returna suma de bani împreună cu calcularea dobînzii timp de 1, 2, 3, 4 ani.

**Codul C++ :**

#include <bits/stdc++.h>

using namespace std;

class MyClass{

float n1;

float n2;

public:

MyClass(){

n1 = 0;

n2 = 1;

}

MyClass(int a, float b){

n1 = a;

n2 = b;

}

MyClass(MyClass &);

void citire();

void afisare();

~ MyClass(){

cout << "";

}

float valoare(int nr\_ani);

};

void MyClass:: citire(){

cout << "Dati suma depozitului: " << endl;

cin >> n1;

cout << "Dati procentul anual: " << endl;

cin >> n2;

int nr\_ani;

cout << "Dati nr anilor: " << endl;

cin >> nr\_ani;

}

float MyClass::valoare(int nr\_ani){

return n1 + ((n1\*n2) / 100) \* nr\_ani;

}

MyClass::MyClass(MyClass & Object){

n1 = Object.n1;

n2 = Object.n2;

}

void MyClass::afisare(){

cout << "Iata suma depozitului: " << n1 << endl;

cout << "Iata procentul anual: " << n2 << endl;

for(int i = 0; i < 4; i++){

cout << "Depozitul bancar dupa anul " << (i+1) << ": " << valoare(i) << endl;

}

}

main(){

MyClass nr;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "Constructor ce citeste valori" << endl;

nr.citire();

nr.afisare();

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "Constructor cu parametri" << endl;

MyClass nr2(7,2);

nr2.afisare();

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "Constructor de copiere" << endl;

MyClass Object(nr2);

nr2.afisare();

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "Construtor pointer" << endl;

MyClass \*n\_r;

n\_r = new MyClass;

n\_r -> afisare();

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

delete n\_r;

cout << "Constructor pointer cu parametri" << endl;

n\_r = new MyClass(7,4);

n\_r -> afisare();

delete n\_r;

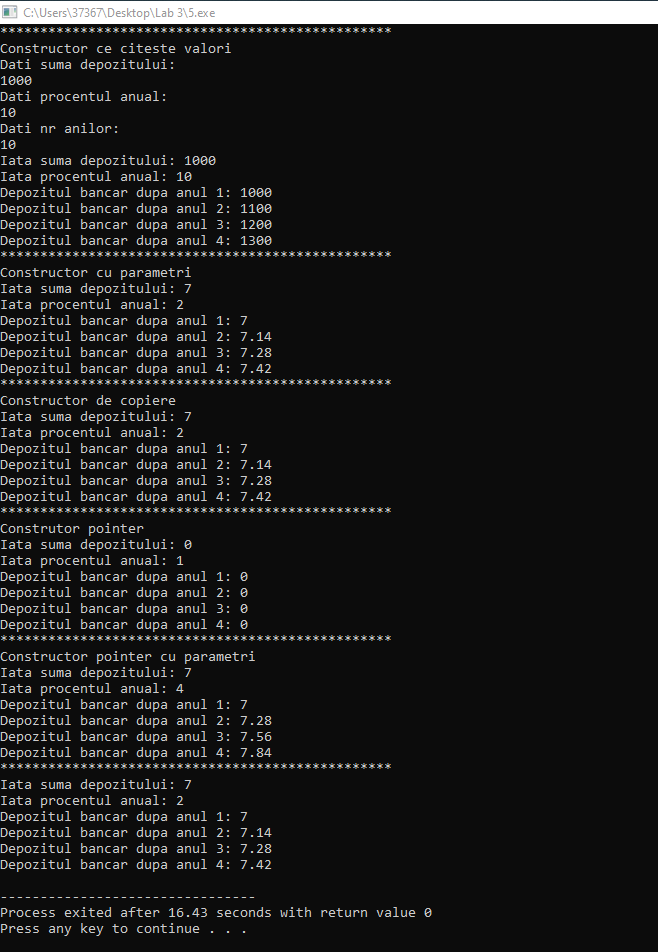
cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

n\_r = new MyClass(nr2);

n\_r -> afisare();

delete n\_r;}

**Screenshot-uri:**



* 1. Cîmpul *n1* - un număr întreg pozitiv (numărul de ore), Cîmpul *n2* – un număr întreg pozitiv (numărul de minute). Creaţi metoda *secunde*() – va returna numărul total al secundelor scrise în cele două cîmpuri.

**Codul C++ :**

#include <bits/stdc++.h>

using namespace std;

class MyClass{

int n1;

int n2;

public:

MyClass(){

n1 = 0;

n2 = 1;

}

MyClass(int a, float b){

n1 = a;

n2 = b;

}

MyClass(MyClass &);

void citire();

void afisare();

~ MyClass(){

cout << "";

}

float secunde();

};

void MyClass:: citire(){

cout << "Dati nr de ore: " << endl;

cin >> n1;

cout << "Dati nr de minute: " << endl;

cin >> n2;

}

MyClass::MyClass(MyClass & Object){

n1 = Object.n1;

n2 = Object.n2;

}

float MyClass::secunde(){

return n1\*3600 + n2\*60;

}

void MyClass::afisare(){

cout <<"Iata nr de ore introdus: " << n1 << endl;

cout << "Iata nr de minute introdus: " << n2 << endl;

cout << "Iata timpul total in secunde: " << secunde() << endl;

}

main(){

MyClass nr;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "Constructor ce citeste valori" << endl;

nr.citire();

nr.afisare();

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "Constructor cu parametri" << endl;

MyClass nr2(7,2);

nr2.afisare();

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "Constructor de copiere" << endl;

MyClass Object(nr2);

nr2.afisare();

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "Construtor pointer" << endl;

MyClass \*n\_r;

n\_r = new MyClass;

n\_r -> afisare();

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

delete n\_r;

cout << "Constructor pointer cu parametri" << endl;

n\_r = new MyClass(7,4);

n\_r -> afisare();

delete n\_r;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

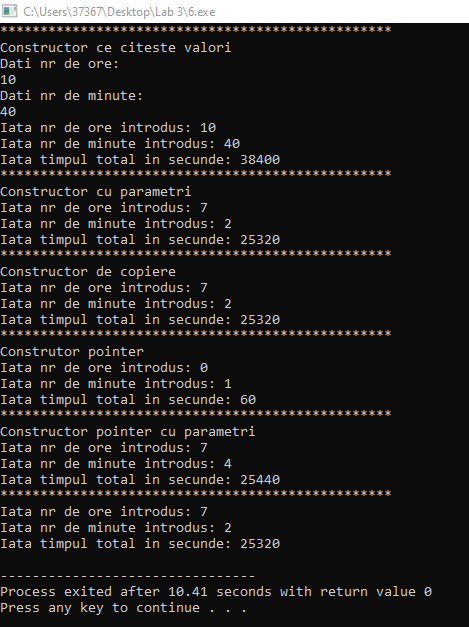
n\_r = new MyClass(nr2);

n\_r -> afisare();

delete n\_r;

}

**Screenshot-uri:**



* 1. Cîmpul *n1* - un număr real (coordonata x), Cîmpul *n2* – număr real (coordonata y). Creaţi metoda *distanta*() – va returna distanţa de la punctul (*0,0*) la punctul cu coordonatele (*n1, n2*).

**Codul C++ :**

#include <bits/stdc++.h>

using namespace std;

class MyClass{

float n1;

float n2;

public:

MyClass(){

n1 = 0;

n2 = 1;

}

MyClass(int a, float b){

n1 = a;

n2 = b;

}

MyClass(MyClass &);

void citire();

void afisare();

~ MyClass(){

cout << "";

}

float distanta();

};

void MyClass:: citire(){

cout << "Dati prima coordonata: " << endl;

cin >> n1;

cout << "Dati a 2-a coordonata: " << endl;

cin >> n2;

}

MyClass::MyClass(MyClass & Object){

n1 = Object.n1;

n2 = Object.n2;

}

float MyClass::distanta(){

return sqrt(n1\*n1 + n2\*n2);

}

void MyClass::afisare(){

cout <<"Iata prima coordonata: " << n1 << endl;

cout << "Iata a 2-a coordonata: " << n2 << endl;

cout << "Iata distanta de la punctul (0,0) la punctul (" << n1 << "," << n2 << ") : " << distanta() << endl;

}

main(){

MyClass nr;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "Constructor ce citeste valori" << endl;

nr.citire();

nr.afisare();

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "Constructor cu parametri" << endl;

MyClass nr2(7,2);

nr2.afisare();

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "Constructor de copiere" << endl;

MyClass Object(nr2);

nr2.afisare();

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "Construtor pointer" << endl;

MyClass \*n\_r;

n\_r = new MyClass;

n\_r -> afisare();

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

delete n\_r;

cout << "Constructor pointer cu parametri" << endl;

n\_r = new MyClass(7,4);

n\_r -> afisare();

delete n\_r;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

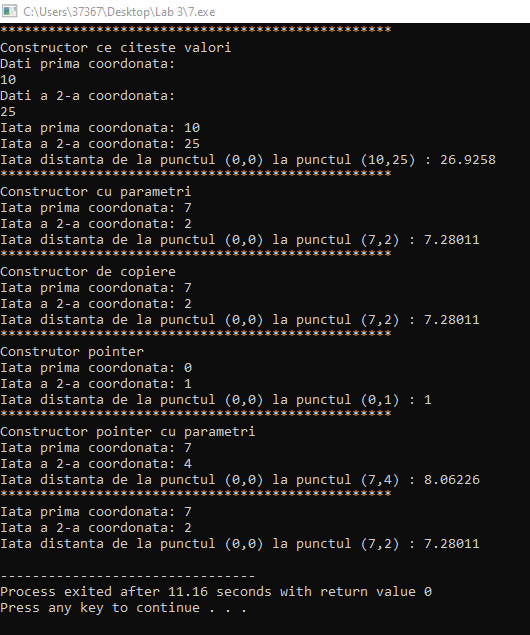
n\_r = new MyClass(nr2);

n\_r -> afisare();

delete n\_r;

}

**Screenshot-uri:**



* 1. Cîmpul *n1* - un număr real (cateta *a* a unui triunghi dreptunghic), Cîmpul *n2*

număr real (cateta *b*). Creaţi metodele *ipotenuza*() – va returna lungimea ipotenuzei triunghiului dreptunghic;

*arie*() - va returna aria triunghiului;

*perimetru*() - va returna perimetrul triunghiului.

**Codul C++ :**

#include <bits/stdc++.h>

using namespace std;

class MyClass{

float n1;

float n2;

public:

MyClass(){

n1 = 0;

n2 = 1;

}

MyClass(int a, float b){

n1 = a;

n2 = b;

}

MyClass(MyClass &);

void citire();

void afisare();

~ MyClass(){

cout << "";

}

float ipotenuza();

float arie();

float perimetrul();

};

void MyClass:: citire(){

cout << "Dati prima cateta: " << endl;

cin >> n1;

cout << "Dati a 2-a cateta: " << endl;

cin >> n2;

}

MyClass::MyClass(MyClass & Object){

n1 = Object.n1;

n2 = Object.n2;

}

float MyClass::ipotenuza(){

return sqrt(n1\*n1 + n2\*n2);

}

float MyClass::arie(){

return (n1\*n2) / 2;

}

float MyClass::perimetrul(){

return n1 + n2 + ipotenuza();

}

void MyClass::afisare(){

cout <<"Iata prima cateta: " << n1 << endl;

cout << "Iata a 2-a cateta: " << n2 << endl;

cout << "Iata ipotenuza: " << ipotenuza() << endl;

cout << "Iata perimetrul: " << perimetrul() << endl;

cout << "Iata aria: " << arie() << endl;

}

main(){

MyClass nr;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "Constructor ce citeste valori" << endl;

nr.citire();

nr.afisare();

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "Constructor cu parametri" << endl;

MyClass nr2(6,9);

nr2.afisare();

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "Constructor de copiere" << endl;

MyClass Object(nr2);

nr2.afisare();

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "Constructor pointer" << endl;

MyClass \*n\_r;

n\_r = new MyClass;

n\_r -> afisare();

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

delete n\_r;

cout << "Constructor pointer cu parametri" << endl;

n\_r = new MyClass(5,7);

n\_r -> afisare();

delete n\_r;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

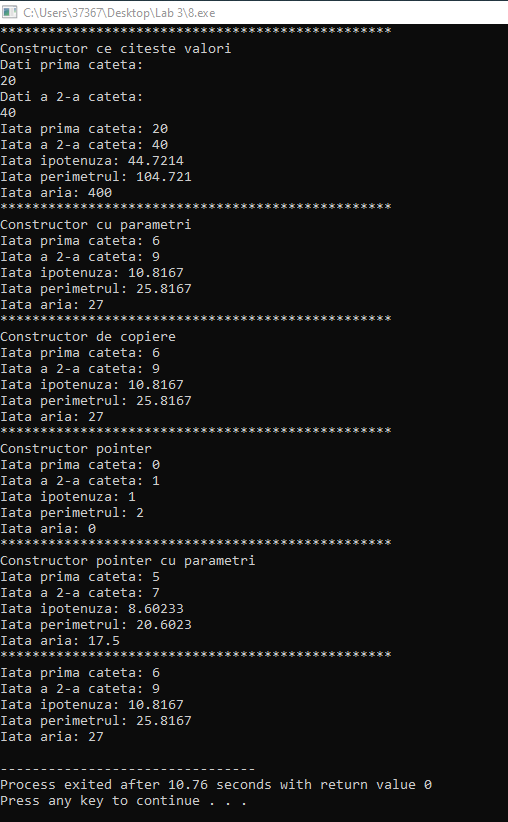
n\_r = new MyClass(nr2);

n\_r -> afisare();

delete n\_r;

}

**Screenshot-uri:**



* 1. Cîmpul *n1* - un număr real (salariul pentru o zi), Cîmpul *n2* – număr întreg (numărul de zile lucrate timp de o lună). Creaţi metoda *suma*() – va returna salariul pentru un an al muncitorului.

**Codul C++ :**

#include <bits/stdc++.h>

using namespace std;

class MyClass{

float n1;

int n2;

public:

MyClass(){

n1 = 0;

n2 = 1;

}

MyClass(int a, float b){

n1 = a;

n2 = b;

}

MyClass(MyClass &);

void citire();

void afisare();

~ MyClass(){

cout << "";

}

float suma();

};

void MyClass:: citire(){

cout << "Dati salariul pentru o zi : " << endl;

cin >> n1;

cout << "Dati numarul de zile lucrate pe luna: " << endl;

cin >> n2;

}

MyClass::MyClass(MyClass & Object){

n1 = Object.n1;

n2 = Object.n2;

}

float MyClass::suma(){

return n1\*n2\*12;

}

void MyClass::afisare(){

cout <<"Iata salariul pentru o zi: " << n1 << endl;

cout << "Iata numarul de zile lucrate pe luna: " << n2 << endl;

cout << "Iata salariul pe un an conform datelor introduse: "<< suma() << endl;

}

main(){

MyClass nr;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "Constructor ce citeste valori" << endl;

nr.citire();

nr.afisare();

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "Constructor cu parametri" << endl;

MyClass nr2(7,2);

nr2.afisare();

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "Constructor de copiere" << endl;

MyClass Object(nr2);

nr2.afisare();

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "Construtor pointer" << endl;

MyClass \*n\_r;

n\_r = new MyClass;

n\_r -> afisare();

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

delete n\_r;

cout << "Constructor pointer cu parametri" << endl;

n\_r = new MyClass(7,4);

n\_r -> afisare();

delete n\_r;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

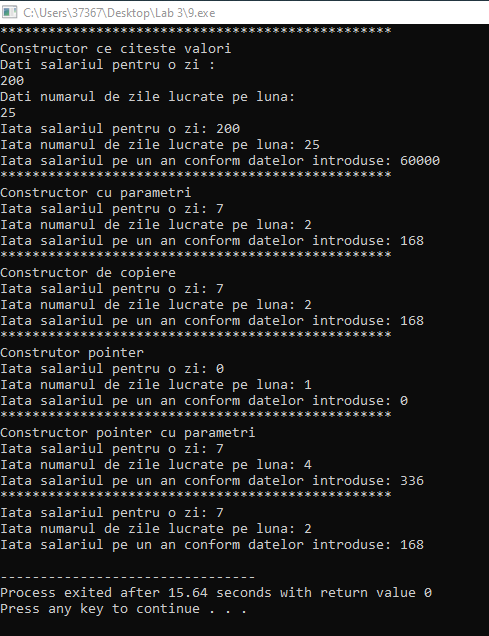
n\_r = new MyClass(nr2);

n\_r -> afisare();

delete n\_r;

}

**Screenshot-uri:**



* 1. Cîmpul *n1* - un număr întreg pozitiv (durata unui apel telefonic în secunde), Cîmpul *n2* – număr real (costul în lei al unui minut). Creaţi metoda *cost*() – va returna costul total al apelului.

**Codul C++ :**

#include <bits/stdc++.h>

using namespace std;

class MyClass{

int n1;

float n2;

public:

MyClass(){

n1 = 0;

n2 = 1;

}

MyClass(int a, float b){

n1 = a;

n2 = b;

}

MyClass(MyClass &);

void citire();

void afisare();

~ MyClass(){

cout << "";

}

float cost();

};

void MyClass:: citire(){

cout << "Dati durata apelului in secunde : " << endl;

cin >> n1;

cout << "Dati costul/minut: " << endl;

cin >> n2;

}

MyClass::MyClass(MyClass & Object){

n1 = Object.n1;

n2 = Object.n2;

}

float MyClass::cost(){

return (n2 / 60) \* n1;;

}

void MyClass::afisare(){

cout <<"Iata durata apelului in secunde: " << n1 << endl;

cout << "Iata costul per minut: " << n2 << endl;

cout << "Iata costul total al apelului conform datelor introduse: "<< cost() << endl;

}

main(){

MyClass nr;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "Constructor ce citeste valori" << endl;

nr.citire();

nr.afisare();

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "Constructor cu parametri" << endl;

MyClass nr2(7,2);

nr2.afisare();

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "Constructor de copiere" << endl;

MyClass Object(nr2);

nr2.afisare();

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "Construtor pointer" << endl;

MyClass \*n\_r;

n\_r = new MyClass;

n\_r -> afisare();

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

delete n\_r;

cout << "Constructor pointer cu parametri" << endl;

n\_r = new MyClass(7,4);

n\_r -> afisare();

delete n\_r;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

n\_r = new MyClass(nr2);

n\_r -> afisare();

delete n\_r;

}

**Screenshot-uri:**

