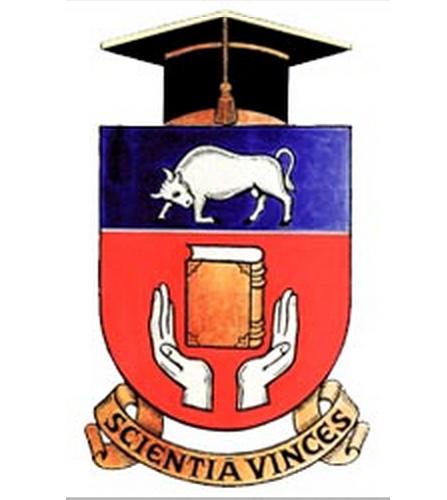
Universitatea de Stat din Tiraspol

Facultatea de Fizica Matematica si Informatica



Laboratorul #2

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

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

**Chișinău, 2021**

**Crearea de obiecte**

1. Creaţi obiectul *cerc*

date: *lungimea razei*;

metode: *citire, afişare, iniţializare, determinarea lungimii discului, a suprafeţei*

*şi a diametrului*.

Se vor introduce datele despre două cercuri şi se vor afişa datele despre cercul cu suprafaţa maximă şi cercul cu lungimea discului minimală.

**Codul C++ :**

#include <iostream>

using namespace std;

class Cerc{

private:

float raza;

float pi = 3.14159;

public:

void init(float r);

void citire();

void afisare();

float lungime();

float arie();

float diametru();

};

void Cerc :: init(float r){

raza = r;

}

void Cerc :: citire(){

cout << "Introduceti raza cercului:";

cin >> raza;

}

void Cerc :: afisare(){

cout << "\nRaza cercului: " << raza << endl;

cout << "Lungimea cercului : " << lungime() << endl;

cout << "Aria cercului: " << arie() << endl;

cout << "Diametrul cercului: " << diametru() << endl;

cout << endl;

}

float Cerc :: lungime(){

return 2 \* raza \* pi;

}

float Cerc :: arie(){

return pi \* raza \* raza;

}

float Cerc :: diametru(){

return raza \* 2;

}

int main(){

Cerc cerc1,cerc2;

cerc1.init(7);

cout << "Datele primului cerc: \n";

cerc1.afisare();

cout << "\*\*\*\n";

cout << "Datele cercului al doilea: \n";

cerc2.citire();

cerc2.afisare();

cout << "\*\*\n";

if (cerc1.arie() > cerc2.arie()){

cout << "Datele cercului cu aria maxima: \n";

cerc1.afisare();

} else {

cout << "Datele cercului cu aria maxima: \n";

cerc2.afisare();

}

if (cerc1.lungime() < cerc2.lungime()) {

cout << "Datele cercului cu lungimea minima: \n";

cerc1.afisare();

} else {

cout << "Datele cercului cu lungimea minima: \n";

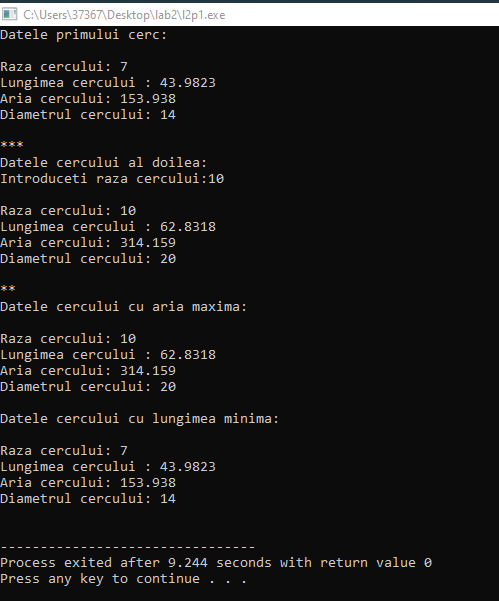
cerc2.afisare();

}

return 0;

}

**Screenshot-uri:**



1. Creaţi obiectul *trapez\_isoscel*

date: *baza mare, baza mică, înălţimea*;

metode: *citire, afişare, iniţializare, determinare a perimetrului şi ariei*.

Se vor introduce datele despre două trapeze şi se va afişa trapezul cu suprafaţa maximă şi trapezul cu perimetrul minim.

**Codul C++ :**

#include <iostream>

#include <cmath>

#include <iomanip>

using namespace std;

class Trapez{

float baza\_mare;

float baza\_mica;

float inaltime;

public:

void citire();

void afisare();

void init(float b\_mare, float b\_mica, float inalt);

float perimetru();

float arie();

};

void Trapez :: citire(){

cout << "Introduceti laturile trapezului: \n";

cout << "Introduceti baza mare : ";

cin >> baza\_mare;

cout << "Introduceti baza mica : ";

cin >> baza\_mica;

cout << "Inaltimea trapezului : ";

cin >> inaltime;

}

void Trapez :: afisare(){

cout << "Datele trapezului: \n";

cout << "Baza mare : " << baza\_mare << " cm \n";

cout << "Baza mica : " << baza\_mica << " cm \n";

cout << "Inaltimea trapezului :" << inaltime << " cm \n";

cout << "Perimetrul trapezului :" << fixed << setprecision(2) << perimetru() << " cm \n";

cout << "Aria trapezului : " << fixed << setprecision(2) << arie() << " cm \n";

}

void Trapez :: init(float b\_mare, float b\_mica, float inalt){

baza\_mare = b\_mare;

baza\_mica = b\_mica;

inaltime = inalt;

}

float Trapez :: perimetru(){

float diff = (baza\_mare - baza\_mica) / 2.0;

float lat = sqrt(diff \* diff + inaltime \* inaltime);

return baza\_mare + baza\_mica + 2 \* lat;

}

float Trapez :: arie(){

return (baza\_mare + baza\_mica) / 2.0 \* inaltime;

}

int main(){

Trapez t1,t2;

t1.init(15,10,7);

cout << "\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

cout << "Datele primului trapez \n";

t1.afisare();

cout << "\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

t2.citire();

cout << "\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

cout << "Datele celui de al doilea trapez: \n";

t2.afisare();

cout << "\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

if (t1.arie() > t2.arie()) {

cout << "Datele trapezului cu aria maxima: \n";

t1.afisare();

} else {

cout << "Datele trapezului cu aria maxima: \n";

t2.afisare();

}

cout << "\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

if (t1.perimetru() < t2.perimetru()){

cout << "Datele trapezelui cu perimetrul minim: \n";

t1.afisare();

} else {

cout << "Datele trapezului cu perimetrul minim: \n";

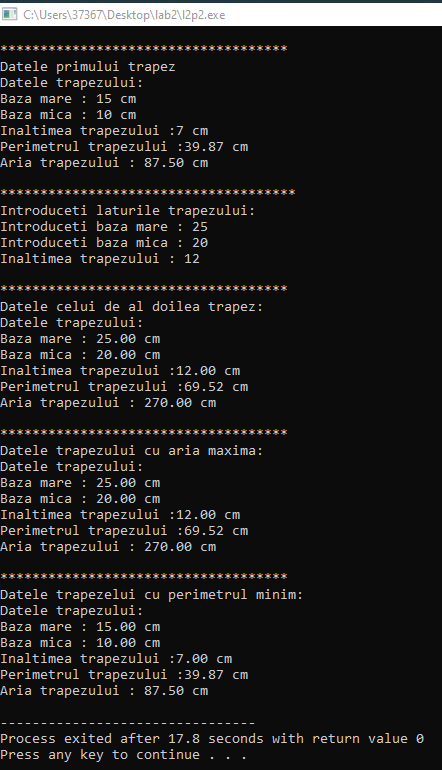
t2.afisare();

}

return 0;

}

**Screenshot-uri:**



1. Creaţi obiectul *paralelogram*

date: *lungimile laturilor şi înălţimea*;

metode: *citire, afişare, iniţializare, determinare a perimetrului şi ariei*.

Se vor introduce datele despre două paralelograme şi se vor afişa datele despre paralelogramul cu suprafaţa maximă şi paralelogramul cu perimetrul minim.

**Codul C++ :**

#include <iostream>

using namespace std;

class paralelogram{

float baza;

float inaltime;

float side1;

float side2;

public:

void citire();

void afisare();

void init();

float perimetru();

float aria();

};

void paralelogram::citire(){

cout << "Dati baza si inaltimea paralelogramului\n";

cin >> baza >> inaltime;

cout << "Introduceti lungimea laturilor adiacente ale paralelogramului\n";

cin >>side1 >>side2;

}

void paralelogram::afisare(){

cout << "Inaltime si baza paralelogramului: " << baza << "," << inaltime <<endl;

cout << "Lungimea laturilor adiacente ale paralelogramului: " << side1 << "," <<side2 <<endl;

cout << "Aria paralelogramului este: " << aria() <<endl;

cout << "Perimetru paralelogramului este: " << perimetru() <<endl;

}

float paralelogram::aria(){

return baza \* inaltime;

}

float paralelogram::perimetru(){

return 2 \* (side1 + side2);

}

int main () {

paralelogram paral1, paral2;

paral1.citire();

cout << "\n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

cout << "Datele despre paralelogram \n";

paral1.afisare();

cout << "\n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

paral2.citire();

cout << "\n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

cout << "Datele despre cel de al 2-lea paralelogram \n";

paral2.afisare();

cout << "\n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

if (paral1.aria() > paral2.aria()){

cout << "Datele paralelogramului cu aria maxima: \n";

paral1.afisare();

} else {

cout << "Datele paralelogramului cu aria maxima: \n";

paral2.afisare();

}

cout <<"\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

if (paral1.perimetru() < paral2.perimetru()) {

cout << "Datele paralelogramului cu perimetru minim: \n";

paral1.afisare();

} else {

cout << "Datele paralelogramului cu periemtru minim: \n";

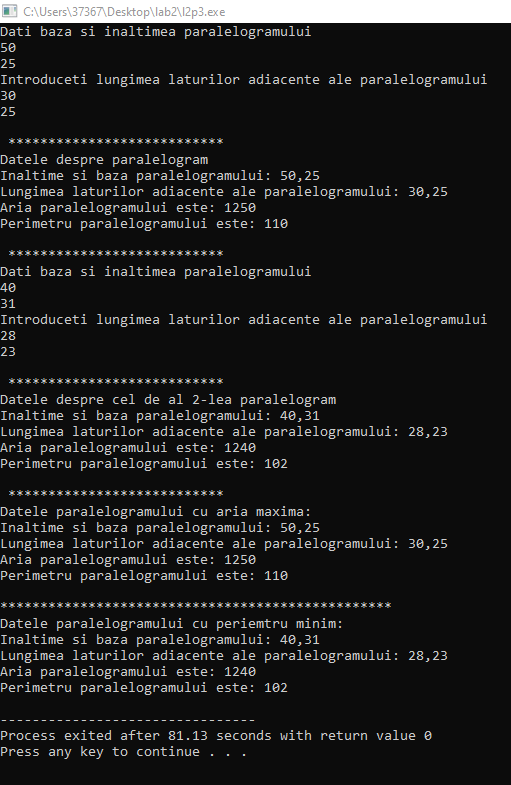
paral2.afisare();

return 0 ;

}

}

**Screenshot-uri:**



1. Creaţi obiectul *triunghi*, date: *lungimile laturilor*;

metode: *citire, afişare, iniţializare, determinare a perimetrului, a suprafeţei şi afişare a tipului triunghiului* (ascuţit, dreptunghic sau obtuzunghi).

Se vor introduce datele despre două triunghiuri şi se vor afişa datele despre triunghiul cu suprafaţa maximă şi triunghiul cu perimetrul minim. Se va implementa   o metodă pentru verificarea corectitudinii introducerii datelor de intrare. În caz că lungimile introduse nu pot forma un triunghi, atunci lungimile laturilor vor fi egale cu 0, iar metoda tip va afişa mesajul: *Nu exista aşa triunghi*.

**Codul C++ :**

#include <iostream>

#include <iomanip>

#include <cmath>

#include <algorithm>

using namespace std;

class Triunghi {

private:

float lat\_1;

float lat\_2;

float lat\_3;

public:

void citire();

void afisare();

void init(float latura\_1, float latura\_2, float latura\_3);

int check();

float perimetru();

float arie();

void tip();

void bubbleSort(float arr[],int n);

};

float Triunghi :: perimetru() {

return lat\_1 + lat\_2 + lat\_3;

}

float Triunghi :: arie(){

float p = (lat\_1 + lat\_2 + lat\_3) / 2.0;

float rs = p \* (p - lat\_1) \* (p - lat\_2) \* (p - lat\_3);

rs = sqrt(rs);

return rs;

}

void Triunghi :: citire() {

cout << "Introduceti datele triunghiului \n";

cout << "Introduceti lungimea primei lature(cm) : ";

cin >> lat\_1;

cout << "Introduceti lungimea celei de-a doua latura(cm) : ";

cin >> lat\_2;

cout << "Introduceti lungimea celei de-a treia latura(cm) : ";

cin >> lat\_3;

}

void Triunghi :: init(float latura\_1, float latura\_2, float latura\_3){

lat\_1 = latura\_1;

lat\_2 = latura\_2;

lat\_3 = latura\_3;

}

int Triunghi :: check(){

if (lat\_1 + lat\_2 > lat\_3 && lat\_1 + lat\_3 > lat\_2 && lat\_2 + lat\_3 > lat\_1){

return 1;

} else {

lat\_1 = 0;

lat\_2 = 0;

lat\_3 = 0;

return 0;

}

}

void Triunghi :: tip(){

float a[3] = {lat\_1, lat\_2, lat\_3};

bubbleSort(a,3);

if (check() == 0){

cout << "Nu exista asa triunghi";

} else {

if (lat\_1 \* lat\_1 + lat\_2 \* lat\_2 == lat\_3 \* lat\_3) {

cout << "Triunghi dreptunghic";

} else if (lat\_1 \* lat\_1 + lat\_2 \* lat\_2 < lat\_3 \* lat\_3){

cout << "Triunghi ascutitunghic";

} else if (lat\_1 \* lat\_1 + lat\_2 \* lat\_2 > lat\_3 \* lat\_3){

cout << "Triunghi obtuzunghic";

}

}

}

void Triunghi :: bubbleSort(float arr[], int n) {

int i, j;

float aux;

for (i = 0; i < n-1; i++)

for (j = 0; j < n-i-1; j++)

if (arr[j] > arr[j+1]){

aux = arr[j];

arr[j] = arr[j+1];

arr[j+1] = aux;

}

}

void Triunghi :: afisare() {

cout << "Lungimea primei lature : " << lat\_1 << " cm" << endl;

cout << "Lungimea celei de-a doua latura : " << lat\_2 << " cm" << endl;

cout << "Lungimea celei de-a treia latura : " << lat\_3 << " cm" << endl;

cout << "Tipul triunghiului : ";

tip();

cout << endl;

cout << "Perimetrul triunghiului : " << fixed << setprecision(2) << perimetru() << "cm \n";

cout << "Aria triunghiului : " << fixed << setprecision(2) << arie() << " centimetri patrati \n";

}

int main(){

Triunghi t1,t2;

t1.init(8,6,5);

t1.check();

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

cout << "\nDatele primului triunghi: \n";

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

t1.afisare();

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

t2.citire();

t2.check();

cout << "\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

cout << "\nDatele celui de al doilea triunghi \n";

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

t2.afisare();

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

cout << "\nDatele triunghiului cu aria maxima: \n";

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

if (t1.arie() > t2.arie()) {

t1.afisare();

} else {

t2.afisare();

}

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

cout << "\nDatele triunghiului cu perimetrul minim:\n";

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

if (t2.perimetru() > t1.perimetru()) {

t1.afisare();

} else {

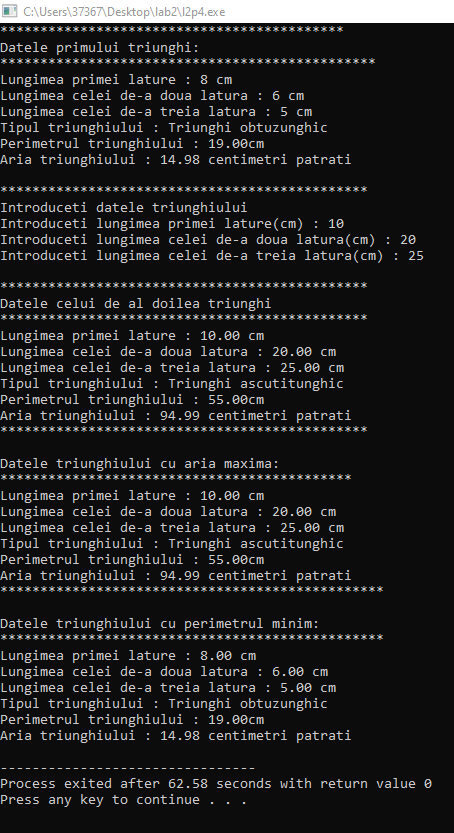
t2.afisare();

}

return 0;

}

**Screenshot-uri:**



1. Creaţi obiectul *paralelipiped\_dreptunghic*

date: *lungimile bazelor şi înălţimea*;

metode: *citire, afişare, iniţializare, determinare a suprafeţei  laterale,  suprafeţei totale, şi a volumului*.

Se vor introduce datele despre două paralelipipede dreptunghice şi se vor afişa datele despre paralelipipedul cu volumul maxim şi paralelipipedul cu suprafaţa totală minimă.  
**Codul C++ :**

#include <iostream>

using namespace std;

class paral\_drept{

int lung;

int lat;

int inaltime;

public:

void citire();

void afisare();

void init(float L, float l, float inalt);

int volum();

int aria();

};

void paral\_drept::citire(){

cout << "Lungimea= " ;cin >> lung;

cout <<"Latimea= " ; cin >> lat;

cout <<"Inaltime= " ; cin >> inaltime;

}

void paral\_drept::afisare(){

cout <<"Lungimea: " <<lung <<endl;

cout <<"Latimea: " <<lat <<endl;

cout << "Inaltime: " <<inaltime <<endl;

cout << "Volumul paralelipipedului este: " <<volum() <<endl;

cout << "Aria paralelipipedului este : " <<aria() <<endl;

}

void paral\_drept :: init(float L, float l, float inalt){

lung = L;

lat = l;

inaltime = inalt;

}

int paral\_drept::volum(){

return lung\*lat\*inaltime;

}

int paral\_drept::aria(){

return 2\*(lung\*lat+lat\*inaltime+lung\*inaltime);

}

int main(){

paral\_drept pd1,pd2;

pd1.init(2,4,7);

cout << "\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

cout << "Datele primului paralelipiped \n";

pd1.afisare();

cout << "\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

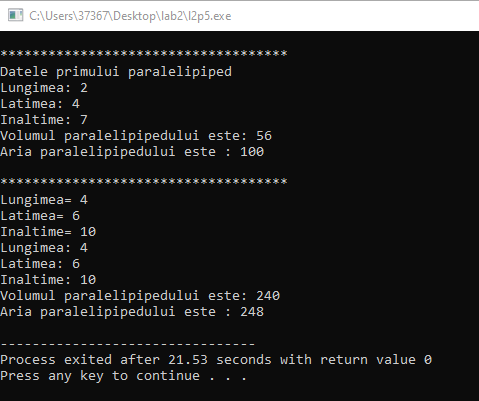
pd2.citire();

pd2.afisare();

return 0 ;

}

**Screenshot-uri:**



1. Creaţi obiectul *ecuaţie1 -* va reprezenta o ecuaţie de gradul 1 *ax+b=0*. date: *a, b*;

metode: *citire, afişare, iniţializare, soluţie*.

Se vor introduce datele despre două ecuaţii şi se vor afişa ecuaţiile introduse cît

şi soluţiile acestora.

**Codul C++ :**

#include <iostream>

#include <iomanip>

using namespace std;

class ecgr1 {

private:

float a;

float b;

public:

void citire();

void afisare();

void init(float a\_2, float b\_2);

float solutie();

};

void ecgr1 :: citire() {

cout << "a=";

cin >> a;

cout << "b=";

cin >> b;

}

void ecgr1 :: afisare(){

if ( a == 0) {

cout <<"Imposibil! A nu trebuie sa fie egal cu 0." <<endl;

}

if (a == 1) {

cout << "x + " << b << " = 0" << endl;

} else if (a > 1){

cout << a << "x + " << b << " = 0" << endl;

}

cout << "Solutia: x = " <<solutie() <<endl;

}

float ecgr1 :: solutie() {

return -b / a;

}

int main(){

ecgr1 ec1, ec2;

ec1.citire();

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

cout << "\nSolutia primei ecuatii: \n";

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

ec1.afisare();

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

ec2.citire();

cout << "\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

cout << "\nSolutia celei de-a doua ecuatie: \n";

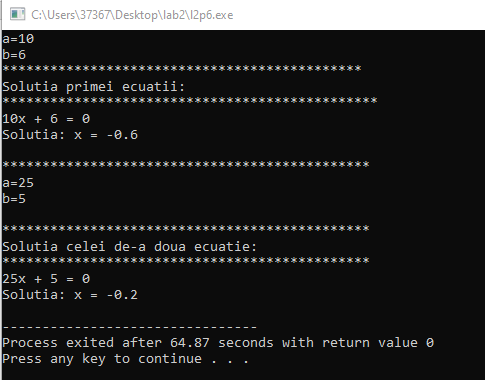
cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

ec2.afisare();

return 0;

}

**Screenshot-uri:**



1. Creaţi obiectul *ecuaţie2 -* va reprezenta o ecuaţie pătratică de ordinul doi

*ax2+bx+c=0*.

date: *a, b, c*;

metode: *citire, afişare, discriminant, soluţie*.

Se vor introduce datele despre două ecuaţii şi se vor afişa ecuaţiile introduse, discriminantul fiecărei ecuaţii cît şi soluţiile acestora.

**Codul C++ :**

#include <iostream>

#include <cmath>

using namespace std;

class ecuatie2{

float a;

float b;

float c;

public:

void citire();

void afisare();

float solutie();

float discrim();

};

void ecuatie2::citire(){

cout << "Dati a: ";

cin >> a ;

cout << "Dati b: ";

cin >> b ;

cout << "Dati c: ";

cin >> c;

}

void ecuatie2::afisare(){

cout << "Ecuatia afisata este: "<<a<<"x^2+"<<b<<"x+"<<c<<"=0"<<endl;

cout << "Solutie: " <<solutie() <<endl;

}

float ecuatie2::solutie(){

float x1,x2;

float delta = b \* b - 4 \* a \*c;

cout << "Delta este:" <<delta<<endl;

if(delta>=0){

x1=(-b+sqrt(delta))/(2\*a);

x2=(-b-sqrt(delta))/(2\*a);

cout<<"x1="<<x1<<endl;

cout<<"x2="<<x2<<endl;

}

else

if(delta<0){

cout<<"Ecuatia nu are solutii in R" <<endl;

}

}

int main () {

ecuatie2 ecgr2[100];

int i,n;

cout << "Introduceti n :";

cin >>n ;

for (i=1;i<=n;i++){

ecgr2[i].citire();

cout << "\n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

}

for (i=1;i<=n;i++){

ecgr2[i].afisare();

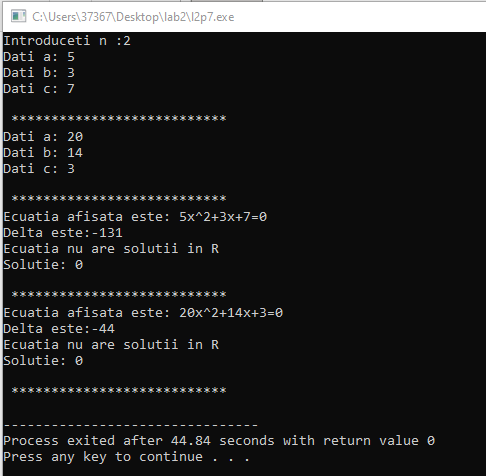
cout << "\n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

}

return 0;

}

**Screenshot-uri:**



1. Creaţi obiectul *persoana*,

date: *nume, anul naşterii, genul, înălţimea*;

metode: *citire, afişare, virsta, greutate* (se va determina conform formulelor: *Gmasculin=50+0.75\*(înălţime-150) + (vîrsta-20)/4 – pentru genul masculin*; *Gfeminin=Gmasculin–10* – *pentru genul feminin*.

De la tastatură se citesc datele despre *n* persoane. Să se afişeze:

1. Lista turor persoanelor;
2. Lista persoanelor cu o vîrstă mai mare decît 25 ani;
3. Lista persoanelor cu o greutate cuprinsă între 55 şi 80 kg.

**Codul C++ :**

#include <iostream>

#include <string>

using namespace std;

class persoana{

public:

string nume;

int anul\_nasterii;

string genul;

int inaltime;

void citire();

void afisare();

int varsta();

float greutate();

};

void persoana :: citire(){

cout<<"Dati Numele : ";

cin>>nume;

cout<<"Dati Anul nasterii : ";

cin>>anul\_nasterii;

cout<<"Dati Genul: ";

cin>>genul;

cout<<"Dati Inaltimea: ";

cin>>inaltime;

}

void persoana :: afisare(){

cout << "Numele : " << nume <<endl;

cout<<"Anul nasterii : " << anul\_nasterii<<endl;

cout<<"Genul: " << genul <<endl;

cout<<"Inaltimea: " << inaltime <<endl;

cout << "Greutatea persoanei: " << greutate() << " kg" << endl;

cout << "Varsta:" <<varsta() <<endl;

}

int persoana :: varsta(){

return 2021-anul\_nasterii;

}

float persoana :: greutate(){

float result = 50 + 0.75 \* (inaltime-150) + (varsta()-20) / 4.0;

if (genul == "m" || genul == "M"){

return result;

}

else if (genul == "f" || genul == "F"){

return result - 10;

}

}

int main() {

persoana persoana1[100];

int i,n ;

cout << "Introduceti n: " <<endl;

cin >> n;

for (i=1;i<=n ;i++){

persoana1[i].citire();

cout << "\n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

}

cout << "Datele despre persoanele introduse:" <<endl;

cout << "\n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

for (i=1;i<=n;i++){

persoana1[i].afisare();

cout << "\n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

}

cout << "\n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

for (i=1;i<=n;i++){

if (persoana1[i].varsta() > 25){

cout <<"Lista persoanelor cu o varsta mai mare decat 25 ani:\n" ;

persoana1[i].afisare();

cout << "\n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

}

}

for (i=1;i<=n;i++){

if (persoana1[i].varsta() < 25){

cout <<"Nu sunt persoane mai mari de varsta de 25 \n";

cout << "\n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

}

}

cout << "\n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

for (i=1;i<=n;i++){

if (persoana1[i].greutate() >= 50 && persoana1[i].greutate() <=80){

cout<<"Persoanele cu greutatea cuprinsa intre 50 si 80 kg: \n";

cout << "\n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

persoana1[i].afisare();

}

else{

cout << "\n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

cout << "Nu sunt persoane cu o greutate cuprinsa intre 50 si 80 kg \n";

cout << "\n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

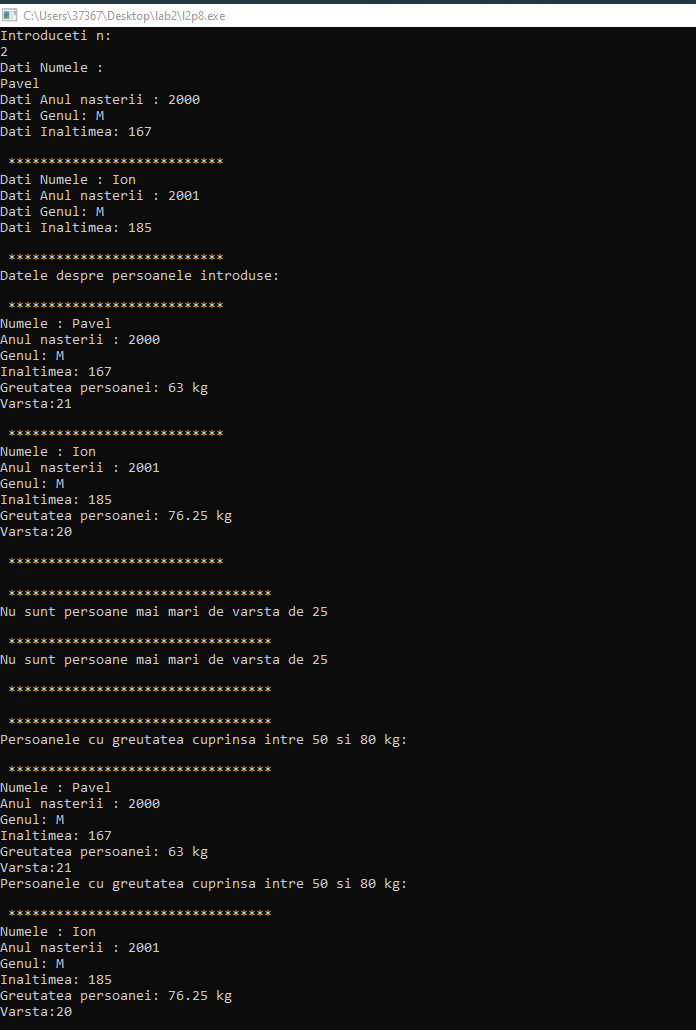
}

}

return 0;

}

**Screenshot-uri:**



1. Creaţi obiectul *student*

date: *nume, anul naşterii, notele la trei discipline*;

metode: *citire, afişare, virsta, media, restanţa* (va returna 1 doar dacă studentul are cel puţin o notă mai mică decît 5), *bursa* (va fi egală cu 500 lei dacă studentul nu are restanţă şi media este mai mare decît 7.5).

De la tastatură se citesc datele despre *n* studenţi. Să se afişeze:

1. Lista tuturor studenţilor;
2. Lista studenţilor restanţieri;
3. Lista studenţilor care au bursă;
4. Lista studenţilor cu o vîrstă mai mică decît 18 ani.

**Codul C++ :**

#include <iostream>

#include <string>

using namespace std;

class student{

string nume;

float anul\_nasterii;

int disci1;

int disci2;

int disci3;

public:

void citire();

void afisare();

float varsta();

float media();

int restanta();

float bursa();

};

void student::citire(){

cout << "Nume student: ";

cin >> nume;

cout << "Anul nasterii: ";

cin >> anul\_nasterii;

cout << "Notele obtinute la trei discipline: ";

cin >> disci1 >> disci2 >> disci3;

}

void student::afisare(){

cout << "Nume student: " << nume <<endl;

cout << "Anul nasterii: " << anul\_nasterii <<endl;

cout << " Notele obtinute la trei discipline: " << disci1 << "," << disci2 << "," << disci3 <<endl;

cout << "Varsta studentului: " << varsta() << " ani" <<endl;

cout << "Media studentului: " << media() <<endl;

cout << "Restante: " << restanta() <<endl;

bursa() ;

}

float student::varsta(){

return 2021-anul\_nasterii;

}

float student::media(){

return (disci1+disci2+disci3)/ 3;

}

int student::restanta(){

if (disci1 < 5 || disci2 < 5 || disci3 < 5 )

return 1;

return 0;

}

float student::bursa(){

if (restanta() > 0 && media() < 7.5) {

cout << "Fara bursa" <<endl;

}

else{

cout << "Bursa 500 lei"<<endl;

}

}

int main(){

student student1[100];

int i,n;

cout<<"Introduceti n: ";

cin >> n;

for(i=1;i<=n;i++){

student1[i].citire();

cout << "\n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

}

cout << "Lista tuturor studentilor: "<<endl;

cout << "\n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

for (i=1;i<=n;i++){

student1[i].afisare();

cout << "\n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

}

cout << "Studentii cu o varsta mai mica de 18 ani:\n";

cout << "\n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

for (i=1;i<=n;i++){

if (student1[i].varsta() < 18){

student1[i].afisare();

cout << "\n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

}

else{

cout << "Nu sunt stundeti mai mici de 18 ani";

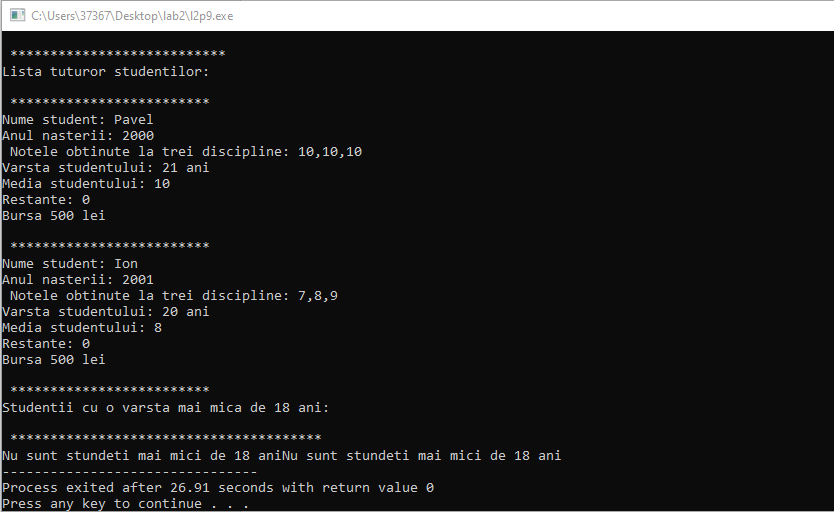
}

}

return 0 ;

}

**Screenshot-uri:**



1. Creaţi obiectul *angajat*

date: *nume, funcţia, anul angajării, ore lucrate*, *plata pentru o oră*;

metode: *citire, afişare, stagiu* (diferenţa dintre anul curent şi anul angajării)*, salariu calculat* (se va determina conform formulei: *ore lucrate\* plata pentru o oră*)*, spor* (va constitui 15% din salariul calculat, dacă stagiul este mai mare decît 5 ani şi

mai mic sau egal decît 8 ani şi 25%, dacă stagiu este mai mare decît 8 ani), *Salariu primit* (se va determina conform formulei: *salariu calculat+spor*).

De la tastatură se citesc datele despre *n* angajaţi. Să se afişeze:

1. Lista tuturor angajaţilor;
2. Lista angajaţilor cu un stagiu mai mare decît 10 ani;
3. Lista angajaţilor cu un salariu mai mare decît 1000;

**Codul C++ :**

#include <iostream>

#include <string>

using namespace std;

class angajat{

string nume;

string functia;

int anul\_angajarii;

int ore\_lucrate;

float plata\_pe\_ora;

public:

void citire();

void afisare();

int stagiu();

float salariu\_calc();

float spor();

float salariu\_primit();

};

void angajat::citire(){

cout << "Nume: ";

cin >> nume;

cout << "Functia: ";

cin >> functia;

cout << "Anul angajarii: ";

cin >>anul\_angajarii;

cout<<"Ore lucrate:";

cin >> ore\_lucrate;

cout <<"Plata pentru o ora: ";

cin >>plata\_pe\_ora;

}

void angajat::afisare(){

cout <<"Numele angajatului: " << nume <<endl;

cout <<"Functia pe care o detine: " <<functia <<endl;

cout <<"Anul in care a fost angajat: " <<anul\_angajarii <<endl;

cout <<"Ore lucrate: " <<ore\_lucrate <<endl;

cout <<"Plata pentru o ora: " <<plata\_pe\_ora <<endl;

cout <<"Stagiul angajatului: " <<stagiu() <<endl;

cout <<"Spor: " <<spor() <<endl;

cout <<"Salariu primit:" <<salariu\_primit() <<endl;

}

int angajat::stagiu(){

return 2021-anul\_angajarii;

}

float angajat::salariu\_calc(){

return ore\_lucrate\*plata\_pe\_ora;

}

float angajat::spor(){

if (stagiu() > 5 && stagiu() <= 8){

return salariu\_calc() \* 0.15;

}

if (stagiu() > 8){

return salariu\_calc() \* 0.25;

}

else{

return 0;

}

}

float angajat::salariu\_primit(){

return salariu\_calc() + spor();

}

int main(){

angajat angajat1[100];

int i,n;

cout<<"Introduceti n: ";

cin >> n;

for(i=1;i<=n;i++){

angajat1[i].citire();

cout << "\n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

}

cout << "Lista tuturor angajatilor: "<<endl;

cout << "\n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

for (i=1;i<=n;i++){

angajat1[i].afisare();

cout << "\n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

}

cout << "Lista angajatilor cu stagiu mai mare decat 10 ani:" <<endl;

cout << "\n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

for (i=1;i<=n;i++){

if (angajat1[i].stagiu() > 10){

angajat1[i].afisare();

}

}

cout << "Lista angajatilor cu salariu mai mare de 1000:" <<endl;

cout << "\n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

for (i=1;i<=n;i++){

if (angajat1[i].salariu\_primit() > 1000){

angajat1[i].afisare();

cout << "\n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

}

}

return 0;

}

**Screenshot-uri:**

