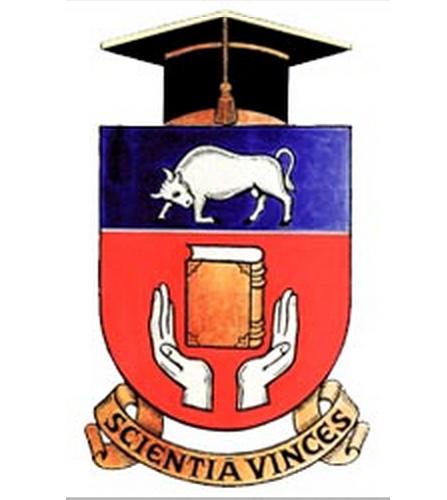
Universitatea de Stat din Tiraspol

Facultatea de Fizica Matematica si Informatica



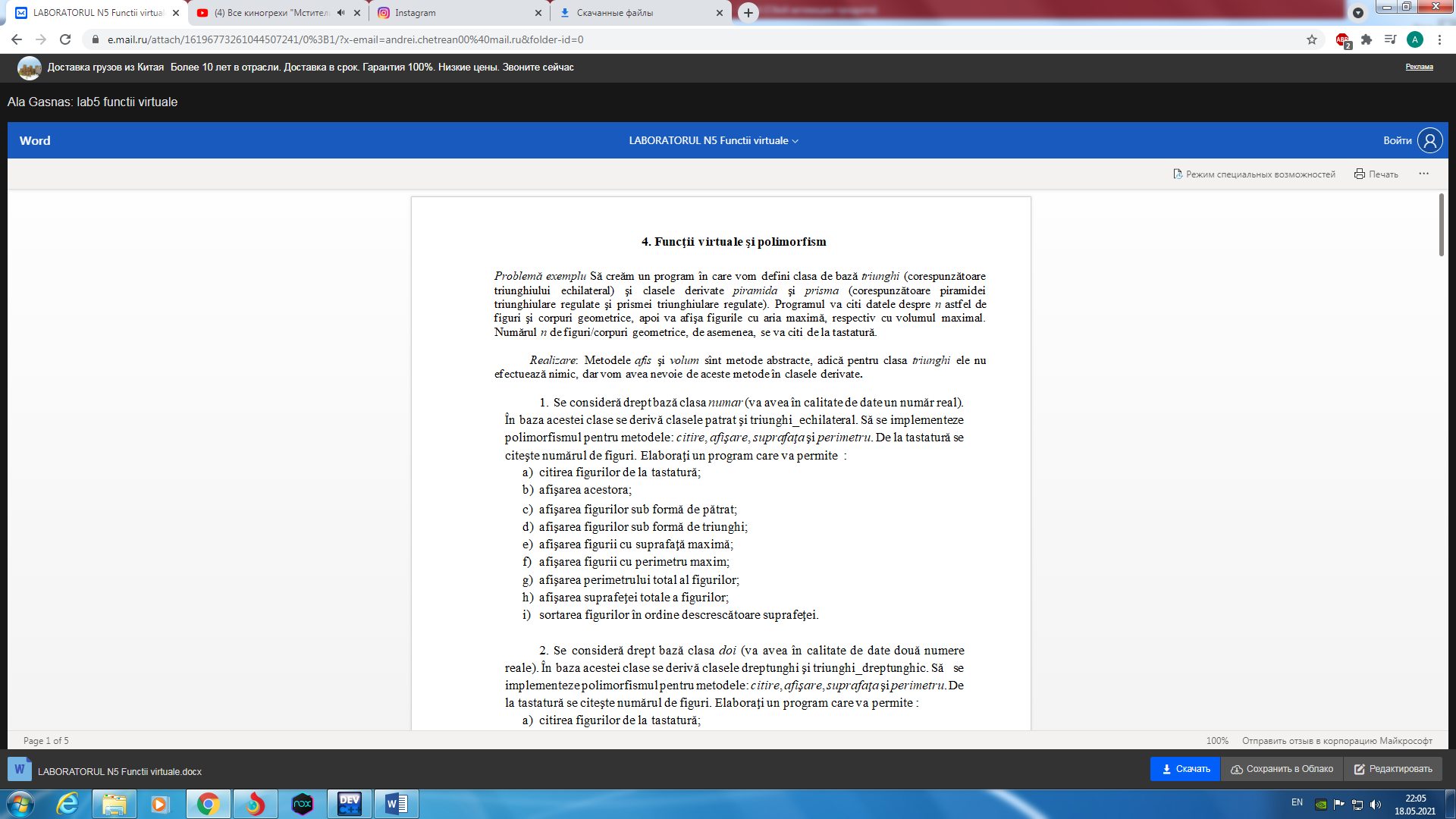
Laboratorul #4

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

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

**Chișinău, 2021**

**Probleme propuse spre rezolvare:**



#include<bits/stdc++.h>

using namespace std;

class triunghi{

protected:

int latura;

public:

virtual void afis() = 0;

virtual void citire();

virtual float aria();

virtual float volum() = 0;

};

class piramida: public triunghi{

protected:

int height;

int apotema;

public:

virtual void afis();

virtual void citire();

virtual float aria();

virtual float volum();

};

class prisma: public triunghi{

protected:

int height;

public:

virtual void afis();

virtual void citire();

virtual float aria();

virtual float volum();

};

void triunghi::citire(){

cout << "\nIntroduceti latura triunghiului echilateral: " ;

cin >> latura;

}

float triunghi::aria(){

return latura \* latura \* pow(3,1/2) / 4;

}

void piramida::afis(){

cout << "Datele piramidei: \n";

cout << "Latura bazei piramidei " << latura << " cm\n";

cout << "Inaltimea piramidei " << height << " cm\n";

cout << "Apotema piramidei " << apotema << " cm\n";

cout << "Aria piramidei " << fixed << setprecision(2) << aria() << " centimetri patrati\n";

cout << "Volumul piramidei " << fixed << setprecision(2) << volum() << " centimetri cubi\n";

}

void piramida::citire(){

triunghi::citire();

cout << "Introduceti inaltimea piramidei: ";

cin >> height;

cout << "Introduceti apotema piramidei: ";

cin >> apotema;

}

float piramida::aria(){

float s = triunghi::aria();

return apotema / 2.0 \* latura \* 3 + s;

}

float piramida::volum(){

float s = triunghi::aria();

return s \* height / 3.0;

}

float prisma::volum(){

float s = triunghi::aria();

return s \* height;

}

float prisma::aria(){

float s = triunghi::aria();

return latura \* height \* 3 + s \* 2;

}

void prisma::citire(){

triunghi::citire();

cout << "Introduceti inaltimea prismei: ";

cin >> height;

}

void prisma::afis(){

cout << "Datele prismei: \n";

cout << "Latura bazei prismei " << latura << " cm\n";

cout << "Inaltimea prismei " << height << " cm\n";

cout << "Aria prismei " << fixed << setprecision(2) << aria() << " centimetri patrati\n";

cout << "Volumul prismei " << fixed << setprecision(2) << volum() << " centimetri cubi\n";

}

int main(){

triunghi \*ob[100];

int n;

cout << "n=";

cin >> n;

int choice;

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

cout << "\n\nIntroduceti alegerea dumneavoastra (1 sau 2): ";

cin >> choice;

if (choice == 1) {

ob[i] = new piramida;

ob[i] -> citire();

} else if(choice == 2) {

ob[i] = new prisma;

ob[i] -> citire();

} else {

cout << "Ati facut alegerea gresit";

i--;

}

}

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

cout << "\n\nDatele figurii numarul " << i + 1 << " :\n";

ob[i] -> afis();

}

float sMax = ob[0] -> aria();

int sMaxIndex = 0;

float vMax = ob[0] -> volum();

int vMaxIndex = 0;

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

if (ob[i] -> aria() > sMax) {

sMax = ob[i] -> aria();

sMaxIndex = i;

}

if (ob[i] -> volum() > vMax) {

vMax = ob[i] -> volum();

vMaxIndex = i;

}

}

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

cout << "\nAria maxima: " << sMax << " centimetri patrati";

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

ob[sMaxIndex] -> afis();

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

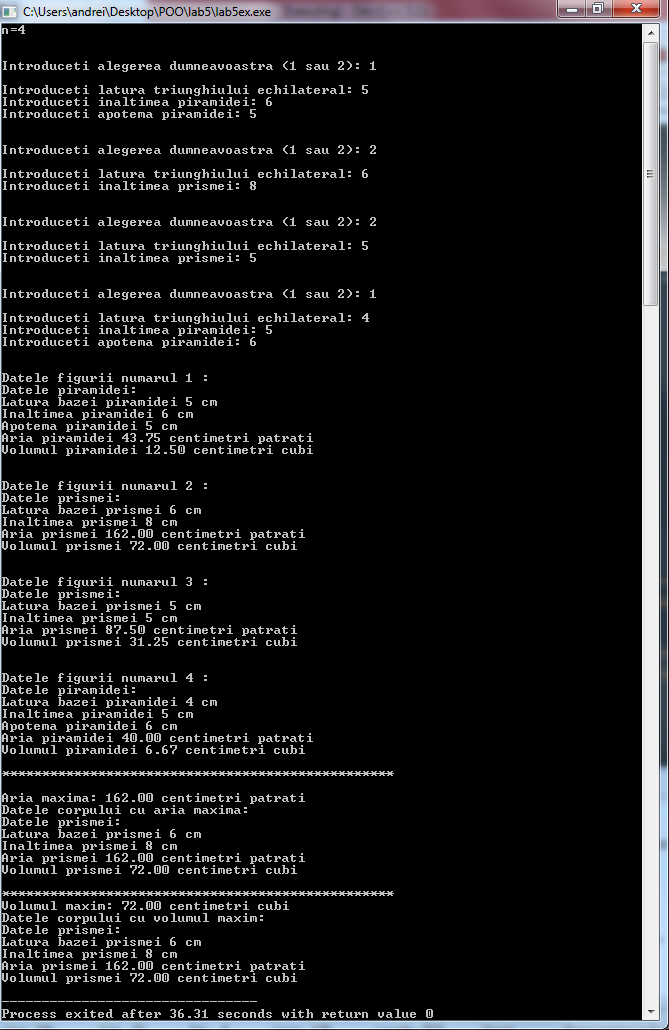
cout << "Volumul maxim: " << vMax << " centimetri cubi";

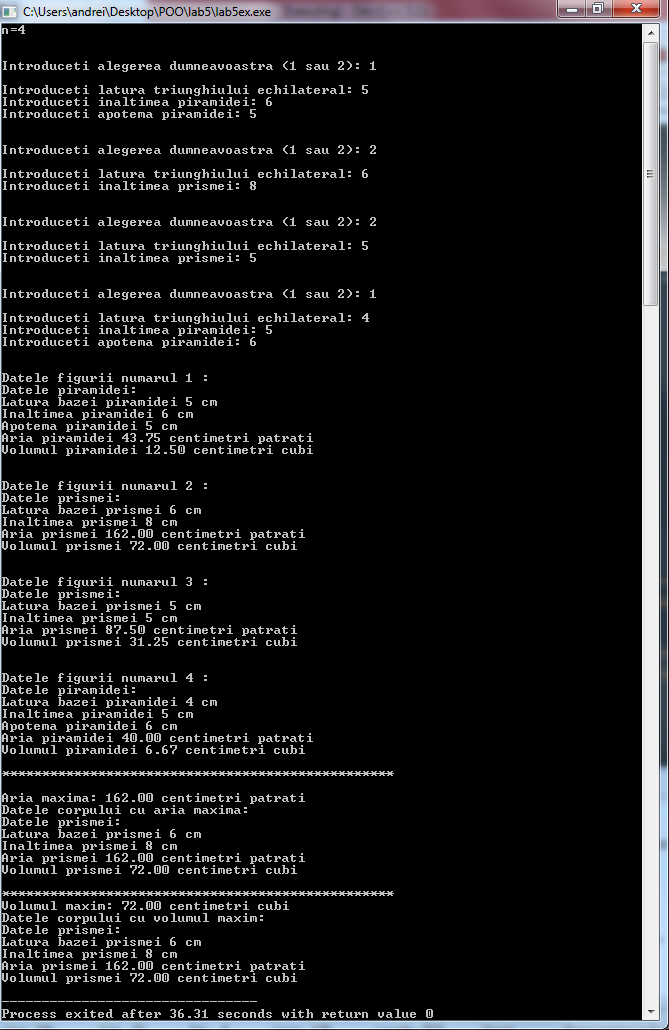
cout << "\nDatele corpului cu volumul maxim:\n";

ob[vMaxIndex] -> afis();

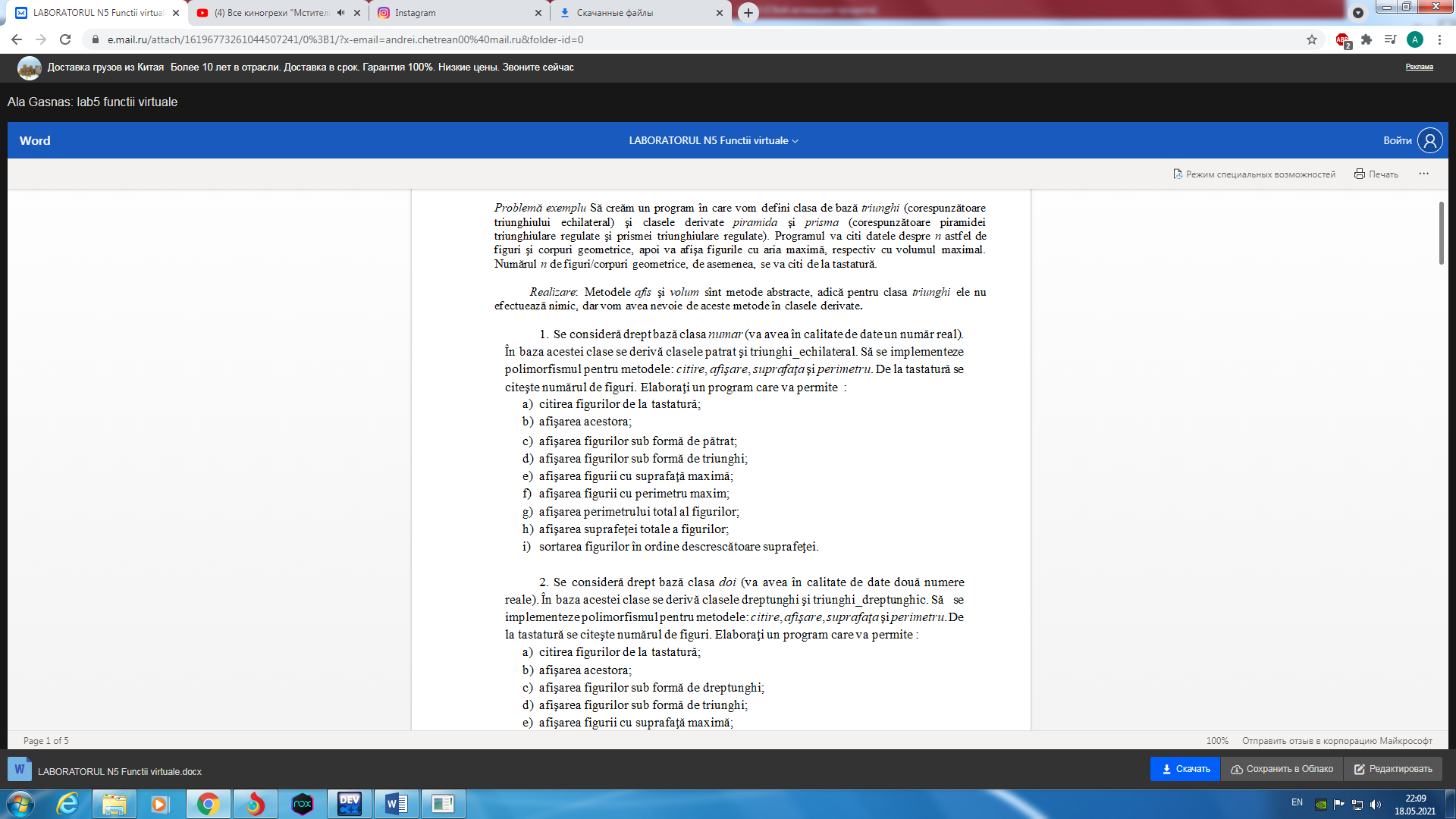
return 0;

}





**Problema 1.**



#include<bits/stdc++.h>

using namespace std;

class numar{

protected:

float num;

public:

virtual void afisare() = 0;

virtual void citire() = 0;

virtual float aria() = 0;

virtual float perimetru() = 0;

};

class patrat: public numar{

public:

virtual void afisare();

virtual void citire();

virtual float aria();

virtual float perimetru();

};

class triunghi\_echilateral: public numar{

public:

virtual void afisare();

virtual void citire();

virtual float aria();

virtual float perimetru();

};

void patrat::citire(){

cout << "Introduceti latura patratului: ";

cin >> num;

}

void triunghi\_echilateral::citire(){

cout << "Introduceti latura triunghiului echilateral: ";

cin >> num;

}

void patrat::afisare(){

cout << "\nLatura patratului: " << num << " cm\n";

cout << "Aria patratului: " << fixed << setprecision(2) << aria() << " centimetri patrati\n";

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

}

void triunghi\_echilateral::afisare(){

cout << "\nLatura triunghiului echilateral: " << num << " cm\n";

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

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

}

float patrat::aria(){

return num \* num;

}

float patrat::perimetru(){

return 4 \* num;

}

float triunghi\_echilateral::aria(){

return num \* num \* sqrt(3) / 4.0;

}

float triunghi\_echilateral::perimetru(){

return 3 \* num;

}

int main(){

numar \*ob[100];

int choices[100];

int n;

cout << "n=";

cin >> n;

int choice;

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

cout << "\n\nIntroduceti alegerea dumneavoastra (1 sau 2): ";

cin >> choice;

if (choice == 1) {

ob[i] = new patrat;

ob[i] -> citire();

choices[i] = 1;

} else if(choice == 2) {

ob[i] = new triunghi\_echilateral;

ob[i] -> citire();

choices[i] = 2;

} else {

cout << "Ati facut alegerea gresit";

i--;

}

}

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

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

cout << "\nDatele figurii numarul " << i + 1 << " :\n";

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

ob[i] -> afisare();

}

cout << "\n\n--------------------------------------------------------";

cout << "\nDatele patratelor create:\n";

cout << "--------------------------------------------------------\n";

int k = 1;

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

if (choices[i] == 1) {

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

cout << "\nDatele patratului numarul " << k << " :\n";

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

ob[i] -> afisare();

k++;

}

}

k = 1;

cout << "\n\n--------------------------------------------------------";

cout << "\nDatele triunghiurilor echilaterale create:\n";

cout << "--------------------------------------------------------\n";

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

if (choices[i] == 2) {

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

cout << "\nDatele triunghiului numarul " << k << " :\n";

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

ob[i] -> afisare();

k++;

}

}

float sMax = ob[0] -> aria();

int sMaxIndex = 0;

float pMax = ob[0] -> perimetru();

int pMaxIndex = 0;

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

if (ob[i] -> aria() > sMax) {

sMax = ob[i] -> aria();

sMaxIndex = i;

}

if (ob[i] -> perimetru() > pMax) {

pMax = ob[i] -> perimetru();

pMaxIndex = i;

}

}

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

cout << "\nAria maxima: " << sMax << " centimetri patrati";

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

ob[sMaxIndex] -> afisare();

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

cout << "Perimetrul maxim: " << pMax << " cm";

cout << "\nDatele figurii cu perimetrul maxim:\n";

ob[pMaxIndex] -> afisare();

float sTotal = 0;

float pTotal = 0;

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

sTotal += ob[i] -> aria();

pTotal += ob[i] -> perimetru();

}

cout << "\n---------------------------------------------------------\n";

cout << "Perimetrul total a figurilor create: " << pTotal << " cm\n";

cout << "\n---------------------------------------------------------\n";

cout << "Aria totala a figurilor create: " << sTotal << " centimetri patrati\n";

numar \*auxObj;

for (int i = 0;i < n; i++)

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

if (ob[i] -> aria() > ob[j] -> aria()) {

auxObj = ob[i];

ob[i] = ob[j];

ob[j] = auxObj;

}

cout << "\n---------------------------------------------------------\n";

cout << "Figurile create sortate dupa suprafata:\n";

cout << "----------------------------------------------------------\n";

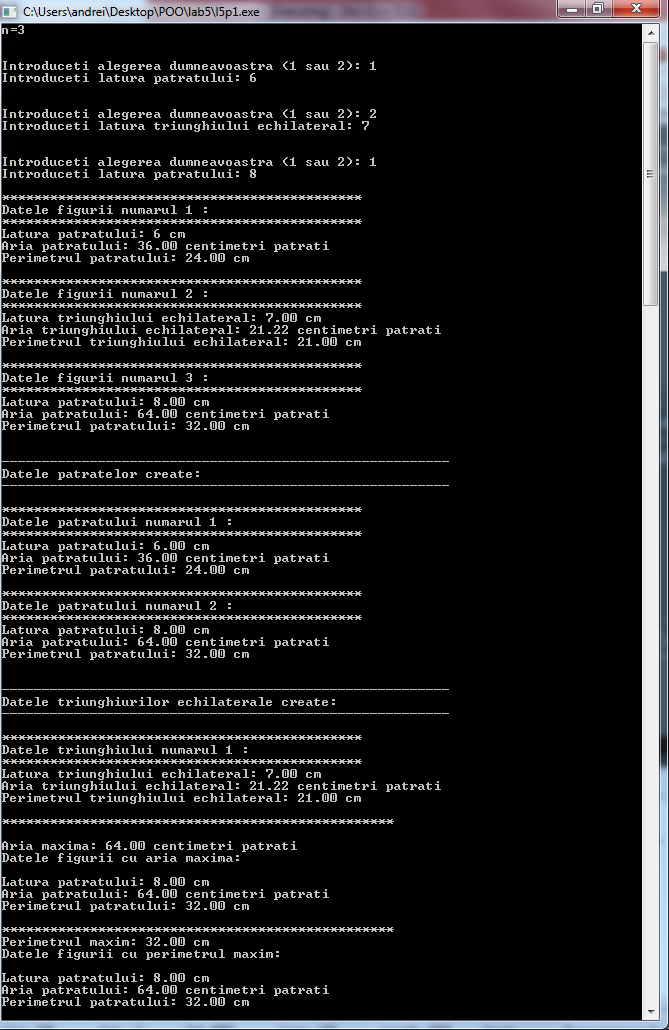
for (int i = 0;i < n; i++) {

ob[i] -> afisare();

}

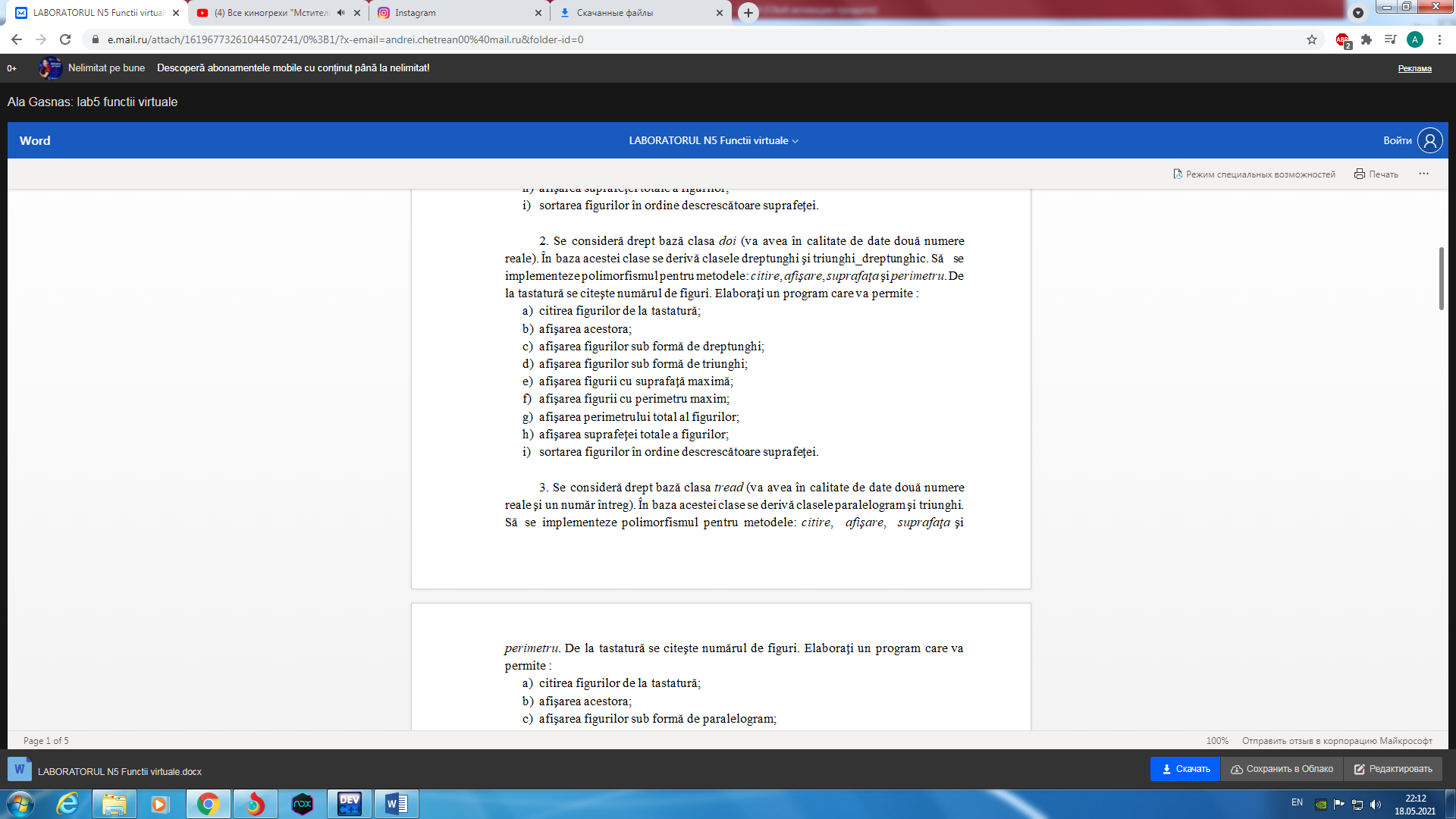
return 0;

}





**Problema 2.**



#include<bits/stdc++.h>

using namespace std;

class doi{

protected:

float num1;

float num2;

public:

virtual void afisare() = 0;

virtual void citire() = 0;

virtual float aria() = 0;

virtual float perimetru() = 0;

};

class dreptunghi: public doi{

public:

virtual void afisare();

virtual void citire();

virtual float aria();

virtual float perimetru();

};

class triunghi\_dreptunghic: public doi{

public:

virtual void afisare();

virtual void citire();

virtual float aria();

virtual float perimetru();

};

void dreptunghi::citire(){

cout << "Introduceti lungimea dreptungiului: ";

cin >> num1;

cout << "Introduceti latimea dreptunghiului: ";

cin >> num2;

}

void triunghi\_dreptunghic::citire(){

cout << "Introduceti prima cateta a triunghiului dreptunghic: ";

cin >> num1;

cout << "Introduceti a doua cateta a triunghiului dreptunghic: ";

cin >> num2;

}

void dreptunghi::afisare(){

cout << "\nLungimea dreptunghiului: " << num1 << " cm\n";

cout << "Latimea dreptunghiului: " << num2 << " cm\n";

cout << "Aria dreptunghiului: " << fixed << setprecision(2) << aria() << " centimetri patrati\n";

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

}

void triunghi\_dreptunghic::afisare(){

cout << "\nPrima cateta a triunghiului dreptunghic: " << num1 << " cm\n";

cout << "A doua cateta a triunghiului dreptunghic: " << num2 << " cm\n";

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

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

}

float dreptunghi::aria(){

return num1 \* num2;

}

float dreptunghi::perimetru(){

return (num1 + num2) \* 2;

}

float triunghi\_dreptunghic::aria(){

return num1 \* num2 / 2.0;

}

float triunghi\_dreptunghic::perimetru(){

float ipotenuza = num1 \* num1 + num2 \* num2;

ipotenuza = sqrt(ipotenuza);

return ipotenuza + num1 + num2;

}

int main(){

doi \*ob[100];

int choices[100];

int n;

cout << "n=";

cin >> n;

int choice;

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

cout << "\n\nIntroduceti alegerea dumneavoastra (1 sau 2): ";

cin >> choice;

if (choice == 1) {

ob[i] = new dreptunghi;

ob[i] -> citire();

choices[i] = 1;

} else if(choice == 2) {

ob[i] = new triunghi\_dreptunghic;

ob[i] -> citire();

choices[i] = 2;

} else {

cout << "Ati facut alegerea gresit";

i--;

}

}

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

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

cout << "\nDatele figurii numarul " << i + 1 << " :\n";

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

ob[i] -> afisare();

}

cout << "\n\n--------------------------------------------------------";

cout << "\nDatele dreptunghiurilor create:\n";

cout << "--------------------------------------------------------\n";

int k = 1;

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

if (choices[i] == 1) {

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

cout << "\nDatele dreptunghiului numarul " << k << " :\n";

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

ob[i] -> afisare();

k++;

}

}

k = 1;

cout << "\n\n--------------------------------------------------------";

cout << "\nDatele triunghiurilor dreptunghice create:\n";

cout << "--------------------------------------------------------\n";

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

if (choices[i] == 2) {

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

cout << "\nDatele triunghiului numarul " << k << " :\n";

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

ob[i] -> afisare();

k++;

}

}

float sMax = ob[0] -> aria();

int sMaxIndex = 0;

float pMax = ob[0] -> perimetru();

int pMaxIndex = 0;

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

if (ob[i] -> aria() > sMax) {

sMax = ob[i] -> aria();

sMaxIndex = i;

}

if (ob[i] -> perimetru() > pMax) {

pMax = ob[i] -> perimetru();

pMaxIndex = i;

}

}

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

cout << "\nAria maxima: " << sMax << " centimetri patrati";

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

ob[sMaxIndex] -> afisare();

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

cout << "Perimetrul maxim: " << pMax << " cm";

cout << "\nDatele figurii cu perimetrul maxim:\n";

ob[pMaxIndex] -> afisare();

float sTotal = 0;

float pTotal = 0;

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

sTotal += ob[i] -> aria();

pTotal += ob[i] -> perimetru();

}

cout << "\n---------------------------------------------------------\n";

cout << "Perimetrul total a figurilor create: " << pTotal << " cm\n";

cout << "\n---------------------------------------------------------\n";

cout << "Aria totala a figurilor create: " << sTotal << " centimetri patrati\n";

doi \*auxObj;

for (int i = 0;i < n; i++)

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

if (ob[i] -> aria() > ob[j] -> aria()) {

auxObj = ob[i];

ob[i] = ob[j];

ob[j] = auxObj;

}

cout << "\n---------------------------------------------------------\n";

cout << "Figurile create sortate dupa suprafata:\n";

cout << "----------------------------------------------------------\n";

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

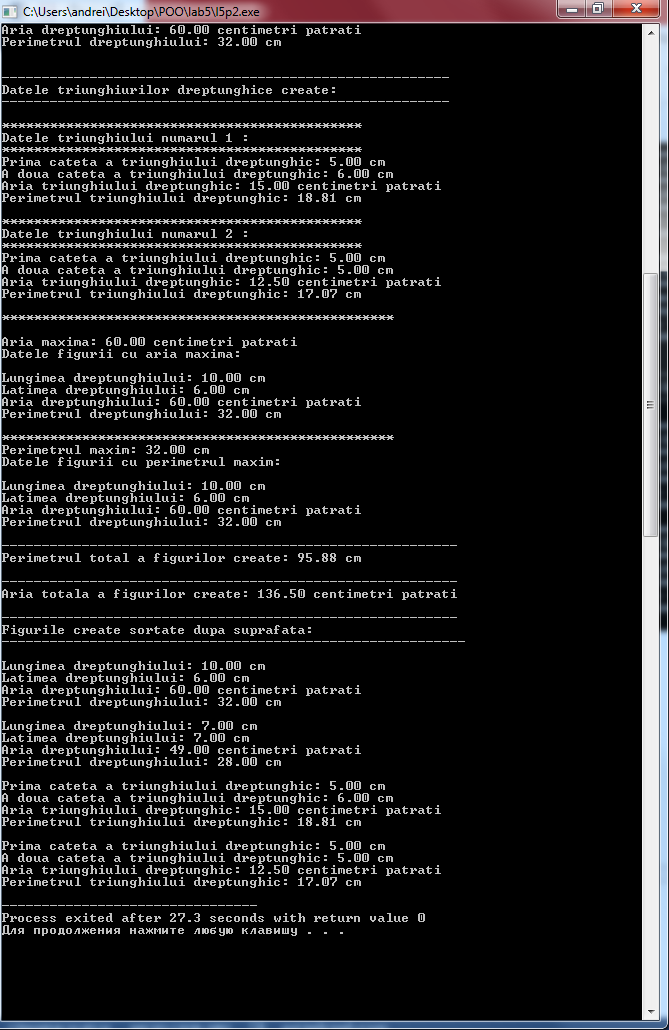
ob[i] -> afisare();

}

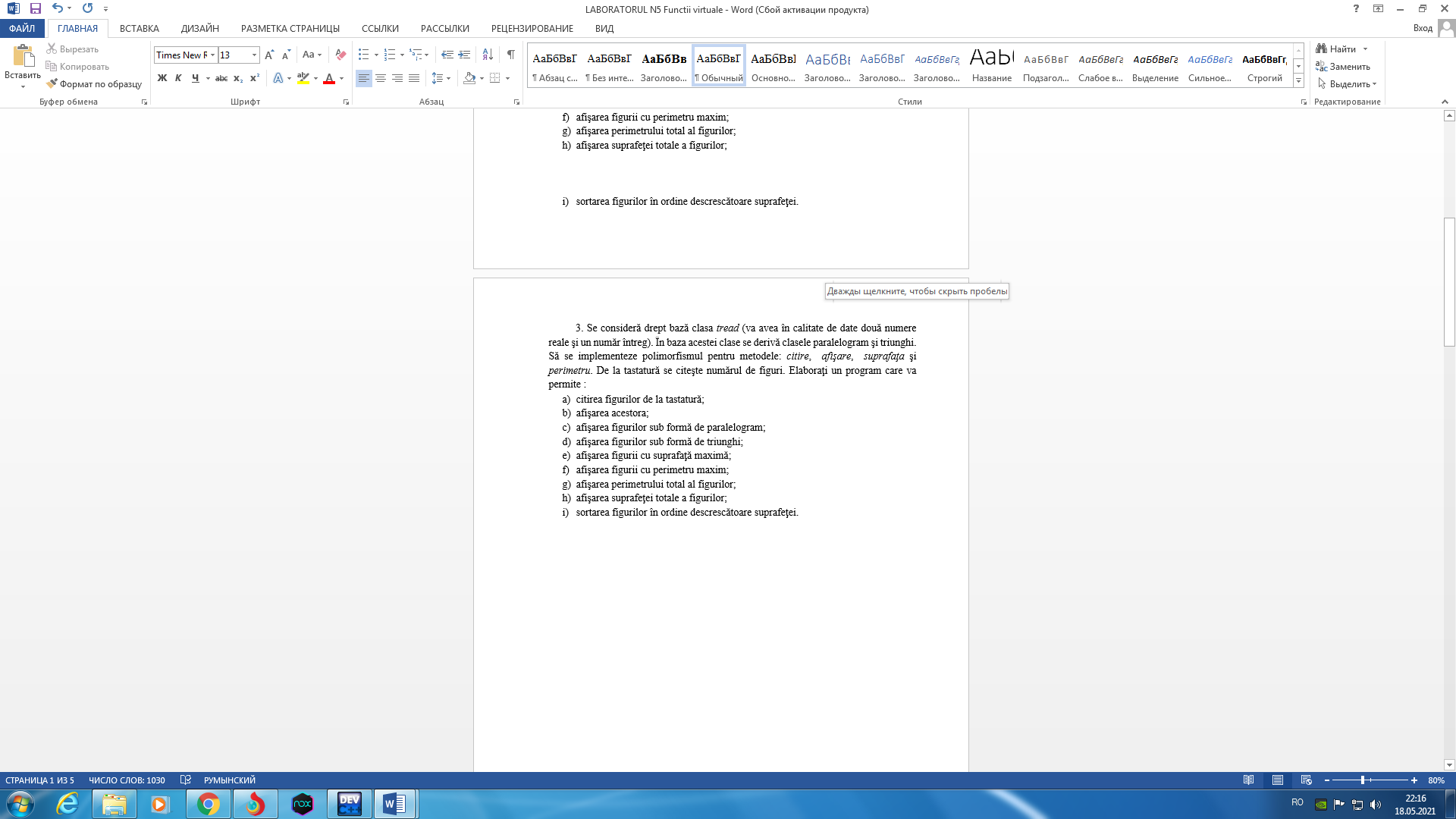
return 0;

}





**Problema 3.**



#include<bits/stdc++.h>

using namespace std;

class tread{

protected:

float num1;

float num2;

int num3;

public:

virtual void afisare() = 0;

virtual void citire() = 0;

virtual float aria() = 0;

virtual float perimetru() = 0;

};

class paralelogram: public tread{

public:

virtual void afisare();

virtual void citire();

virtual float aria();

virtual float perimetru();

};

class triunghi: public tread{

public:

virtual void afisare();

virtual void citire();

virtual float aria();

virtual float perimetru();

};

void paralelogram::citire(){

cout << "Introduceti baza paralelogramului: ";

cin >> num1;

cout << "Introduceti latura paralelogramului: ";

cin >> num2;

cout << "Introduceti inaltimea paralelogramului: ";

cin >> num3;

}

void triunghi::citire(){

cout << "Introduceti prima latura a triunghiului: ";

cin >> num1;

cout << "Introduceti a doua latura a triunghiului: ";

cin >> num2;

cout << "Introduceti a treia latura a triunghiului: ";

cin >> num3;

}

void paralelogram::afisare(){

cout << "\nBaza parelelogramului: " << num1 << " cm\n";

cout << "Latura paralelogramului: " << num2 << " cm\n";

cout << "Inaltimea paralelogramului: " << num3 << " cm\n";

cout << "Aria paralelogramului: " << fixed << setprecision(2) << aria() << " centimetri patrati\n";

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

}

void triunghi::afisare(){

cout << "\nPrima latura a triunghiului : " << num1 << " cm\n";

cout << "A doua latura a triunghiului : " << num2 << " cm\n";

cout << "A treia latura a triunghiului : " << num3 << " cm\n";

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

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

}

float paralelogram::aria(){

return num1 \* num3;

}

float paralelogram::perimetru(){

return num1 \* 2 + num2 \* 2;

}

float triunghi::aria(){

float p = (num1 + num2 + num3) / 2.0;

float s = p \* (p - num1) \* (p - num2) \* (p - num3);

s = sqrt(s);

return s;

}

float triunghi::perimetru(){

return num1 + num2 + num3;

}

int main(){

tread \*ob[100];

int choices[100];

int n;

cout << "n=";

cin >> n;

int choice;

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

cout << "\n\nIntroduceti alegerea dumneavoastra (1 sau 2): ";

cin >> choice;

if (choice == 1) {

ob[i] = new paralelogram;

ob[i] -> citire();

choices[i] = 1;

} else if(choice == 2) {

ob[i] = new triunghi;

ob[i] -> citire();

choices[i] = 2;

} else {

cout << "Ati facut alegerea gresit";

i--;

}

}

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

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

cout << "\nDatele figurii numarul " << i + 1 << " :\n";

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

ob[i] -> afisare();

}

cout << "\n\n--------------------------------------------------------";

cout << "\nDatele paralelogramelor create:\n";

cout << "--------------------------------------------------------\n";

int k = 1;

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

if (choices[i] == 1) {

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

cout << "\nDatele paralelogramului numarul " << k << " :\n";

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

ob[i] -> afisare();

k++;

}

}

k = 1;

cout << "\n\n--------------------------------------------------------";

cout << "\nDatele triunghiurilor create:\n";

cout << "--------------------------------------------------------\n";

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

if (choices[i] == 2) {

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

cout << "\nDatele triunghiului numarul " << k << " :\n";

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

ob[i] -> afisare();

k++;

}

}

float sMax = ob[0] -> aria();

int sMaxIndex = 0;

float pMax = ob[0] -> perimetru();

int pMaxIndex = 0;

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

if (ob[i] -> aria() > sMax) {

sMax = ob[i] -> aria();

sMaxIndex = i;

}

if (ob[i] -> perimetru() > pMax) {

pMax = ob[i] -> perimetru();

pMaxIndex = i;

}

}

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

cout << "\nAria maxima: " << sMax << " centimetri patrati";

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

ob[sMaxIndex] -> afisare();

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

cout << "Perimetrul maxim: " << pMax << " cm";

cout << "\nDatele figurii cu perimetrul maxim:\n";

ob[pMaxIndex] -> afisare();

float sTotal = 0;

float pTotal = 0;

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

sTotal += ob[i] -> aria();

pTotal += ob[i] -> perimetru();

}

cout << "\n---------------------------------------------------------\n";

cout << "Perimetrul total a figurilor create: " << pTotal << " cm\n";

cout << "\n---------------------------------------------------------\n";

cout << "Aria totala a figurilor create: " << sTotal << " centimetri patrati\n";

tread \*auxObj;

for (int i = 0;i < n; i++)

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

if (ob[i] -> aria() > ob[j] -> aria()) {

auxObj = ob[i];

ob[i] = ob[j];

ob[j] = auxObj;

}

cout << "\n---------------------------------------------------------\n";

cout << "Figurile create sortate dupa suprafata:\n";

cout << "----------------------------------------------------------\n";

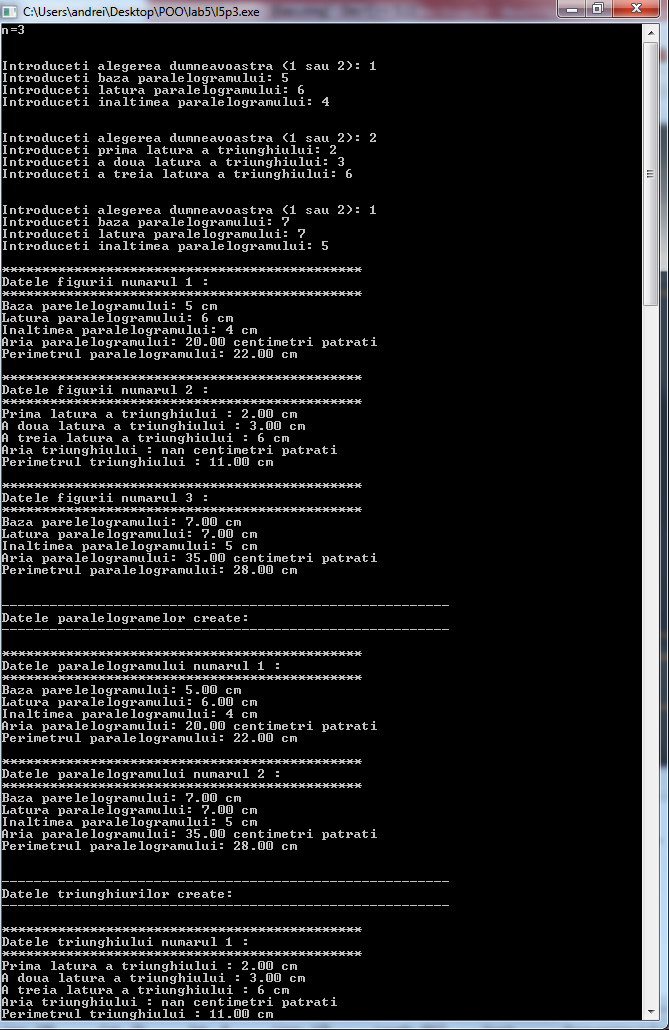
for (int i = 0;i < n; i++) {

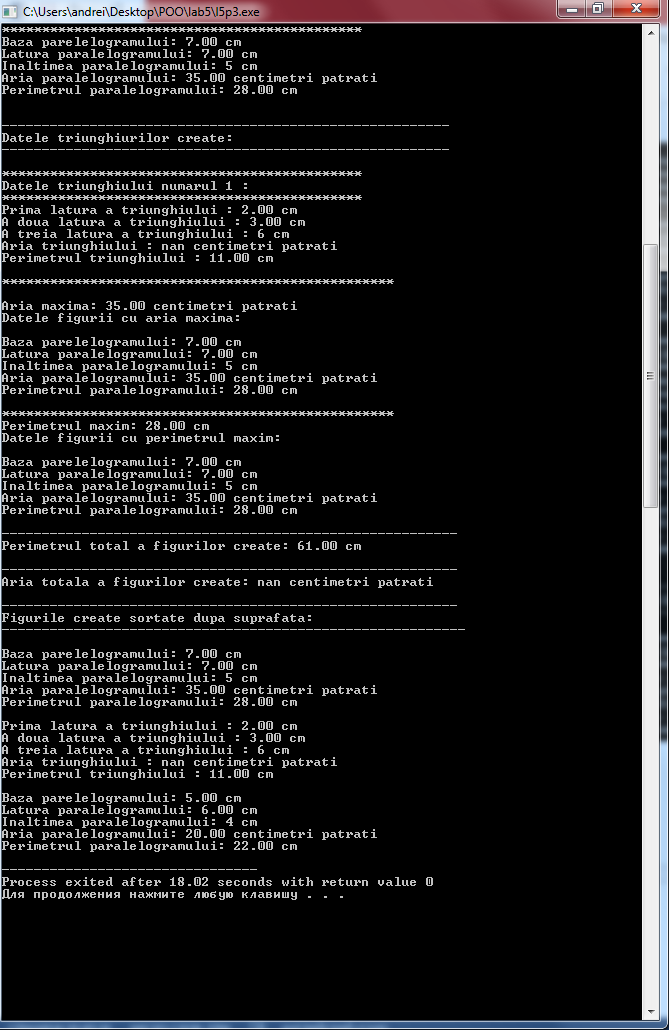
ob[i] -> afisare();

}

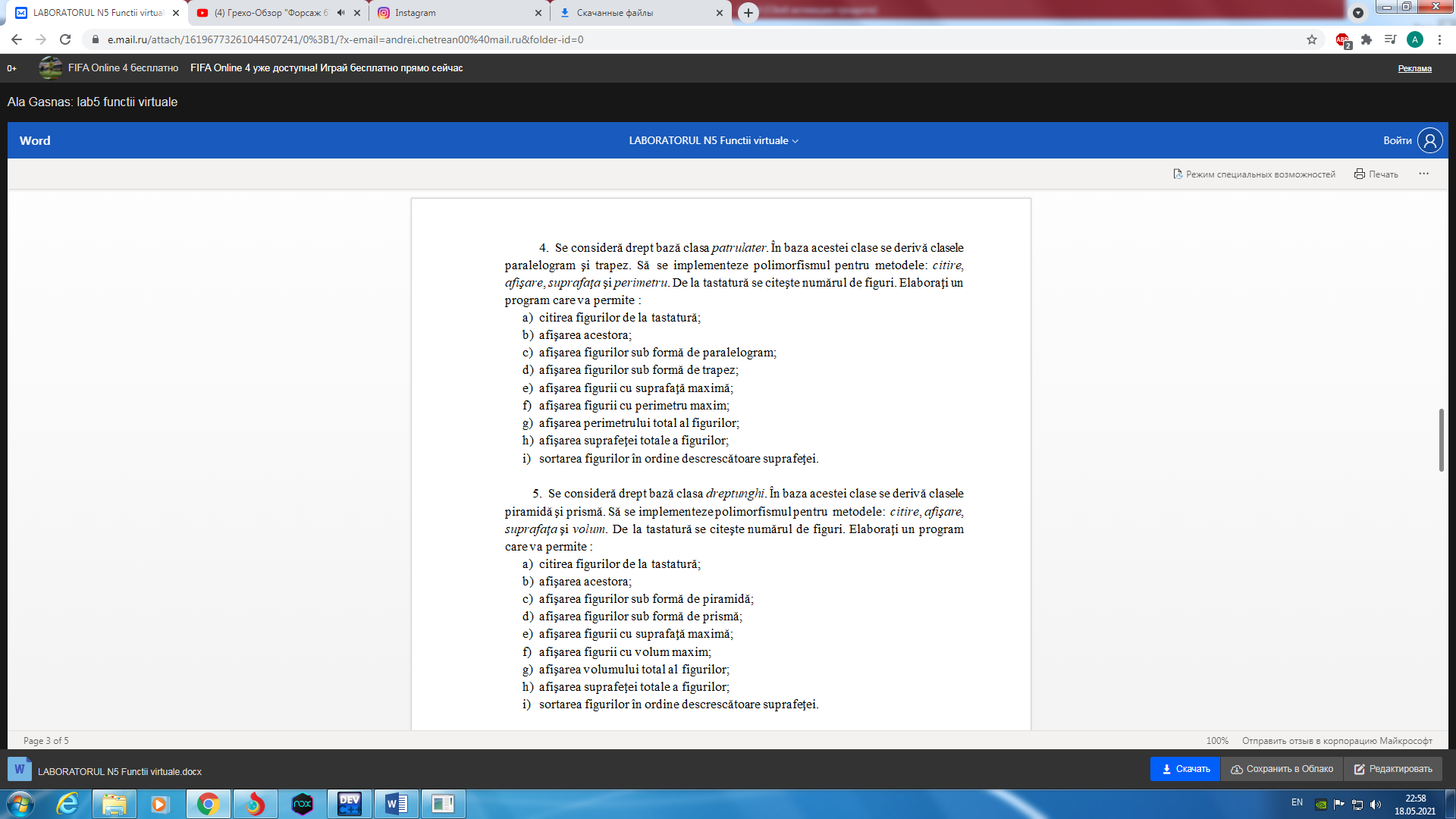
return 0;

}





**Problema 4.**



#include<bits/stdc++.h>

using namespace std;

class tread{

protected:

float num1;

float num2;

int num3;

public:

virtual void afisare() = 0;

virtual void citire() = 0;

virtual float aria() = 0;

virtual float perimetru() = 0;

};

class paralelogram: public tread{

public:

virtual void afisare();

virtual void citire();

virtual float aria();

virtual float perimetru();

};

class triunghi: public tread{

public:

virtual void afisare();

virtual void citire();

virtual float aria();

virtual float perimetru();

};

void paralelogram::citire(){

cout << "Introduceti baza paralelogramului: ";

cin >> num1;

cout << "Introduceti latura paralelogramului: ";

cin >> num2;

cout << "Introduceti inaltimea paralelogramului: ";

cin >> num3;

}

void triunghi::citire(){

cout << "Introduceti prima latura a triunghiului: ";

cin >> num1;

cout << "Introduceti a doua latura a triunghiului: ";

cin >> num2;

cout << "Introduceti a treia latura a triunghiului: ";

cin >> num3;

}

void paralelogram::afisare(){

cout << "\nBaza parelelogramului: " << num1 << " cm\n";

cout << "Latura paralelogramului: " << num2 << " cm\n";

cout << "Inaltimea paralelogramului: " << num3 << " cm\n";

cout << "Aria paralelogramului: " << fixed << setprecision(2) << aria() << " centimetri patrati\n";

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

}

void triunghi::afisare(){

cout << "\nPrima latura a triunghiului : " << num1 << " cm\n";

cout << "A doua latura a triunghiului : " << num2 << " cm\n";

cout << "A treia latura a triunghiului : " << num3 << " cm\n";

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

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

}

float paralelogram::aria(){

return num1 \* num3;

}

float paralelogram::perimetru(){

return num1 \* 2 + num2 \* 2;

}

float triunghi::aria(){

float p = (num1 + num2 + num3) / 2.0;

float s = p \* (p - num1) \* (p - num2) \* (p - num3);

s = sqrt(s);

return s;

}

float triunghi::perimetru(){

return num1 + num2 + num3;

}

int main(){

tread \*ob[100];

int choices[100];

int n;

cout << "n=";

cin >> n;

int choice;

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

cout << "\n\nIntroduceti alegerea dumneavoastra (1 sau 2): ";

cin >> choice;

if (choice == 1) {

ob[i] = new paralelogram;

ob[i] -> citire();

choices[i] = 1;

} else if(choice == 2) {

ob[i] = new triunghi;

ob[i] -> citire();

choices[i] = 2;

} else {

cout << "Ati facut alegerea gresit";

i--;

}

}

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

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

cout << "\nDatele figurii numarul " << i + 1 << " :\n";

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

ob[i] -> afisare();

}

cout << "\n\n--------------------------------------------------------";

cout << "\nDatele paralelogramelor create:\n";

cout << "--------------------------------------------------------\n";

int k = 1;

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

if (choices[i] == 1) {

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

cout << "\nDatele paralelogramului numarul " << k << " :\n";

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

ob[i] -> afisare();

k++;

}

}

k = 1;

cout << "\n\n--------------------------------------------------------";

cout << "\nDatele triunghiurilor create:\n";

cout << "--------------------------------------------------------\n";

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

if (choices[i] == 2) {

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

cout << "\nDatele triunghiului numarul " << k << " :\n";

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

ob[i] -> afisare();

k++;

}

}

float sMax = ob[0] -> aria();

int sMaxIndex = 0;

float pMax = ob[0] -> perimetru();

int pMaxIndex = 0;

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

if (ob[i] -> aria() > sMax) {

sMax = ob[i] -> aria();

sMaxIndex = i;

}

if (ob[i] -> perimetru() > pMax) {

pMax = ob[i] -> perimetru();

pMaxIndex = i;

}

}

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

cout << "\nAria maxima: " << sMax << " centimetri patrati";

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

ob[sMaxIndex] -> afisare();

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

cout << "Perimetrul maxim: " << pMax << " cm";

cout << "\nDatele figurii cu perimetrul maxim:\n";

ob[pMaxIndex] -> afisare();

float sTotal = 0;

float pTotal = 0;

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

sTotal += ob[i] -> aria();

pTotal += ob[i] -> perimetru();

}

cout << "\n---------------------------------------------------------\n";

cout << "Perimetrul total a figurilor create: " << pTotal << " cm\n";

cout << "\n---------------------------------------------------------\n";

cout << "Aria totala a figurilor create: " << sTotal << " centimetri patrati\n";

tread \*auxObj;

for (int i = 0;i < n; i++)

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

if (ob[i] -> aria() > ob[j] -> aria()) {

auxObj = ob[i];

ob[i] = ob[j];

ob[j] = auxObj;

}

cout << "\n---------------------------------------------------------\n";

cout << "Figurile create sortate dupa suprafata:\n";

cout << "----------------------------------------------------------\n";

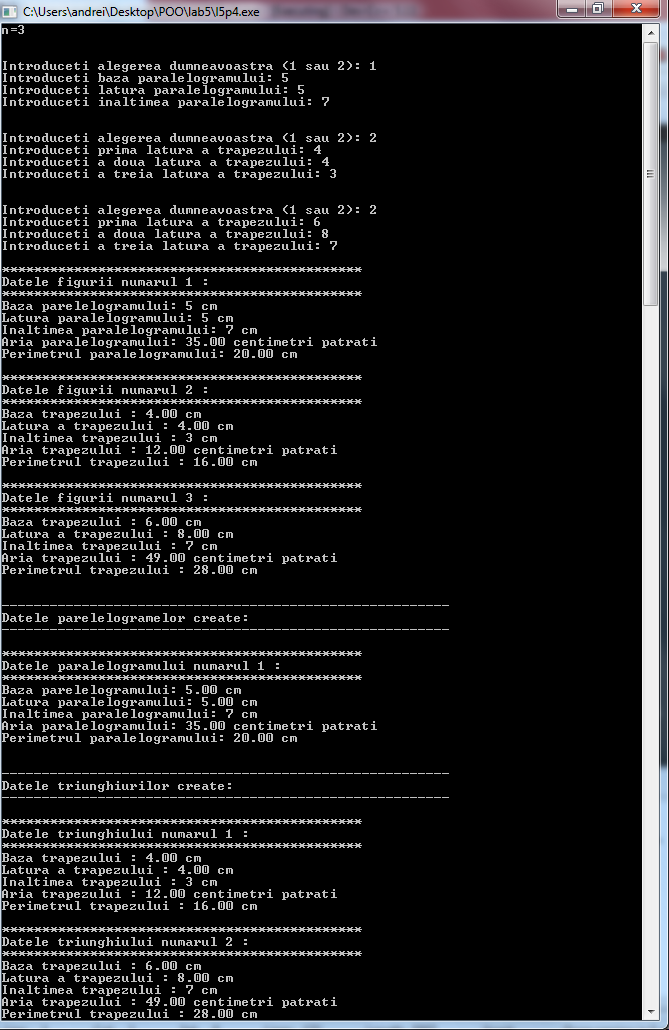
for (int i = 0;i < n; i++) {

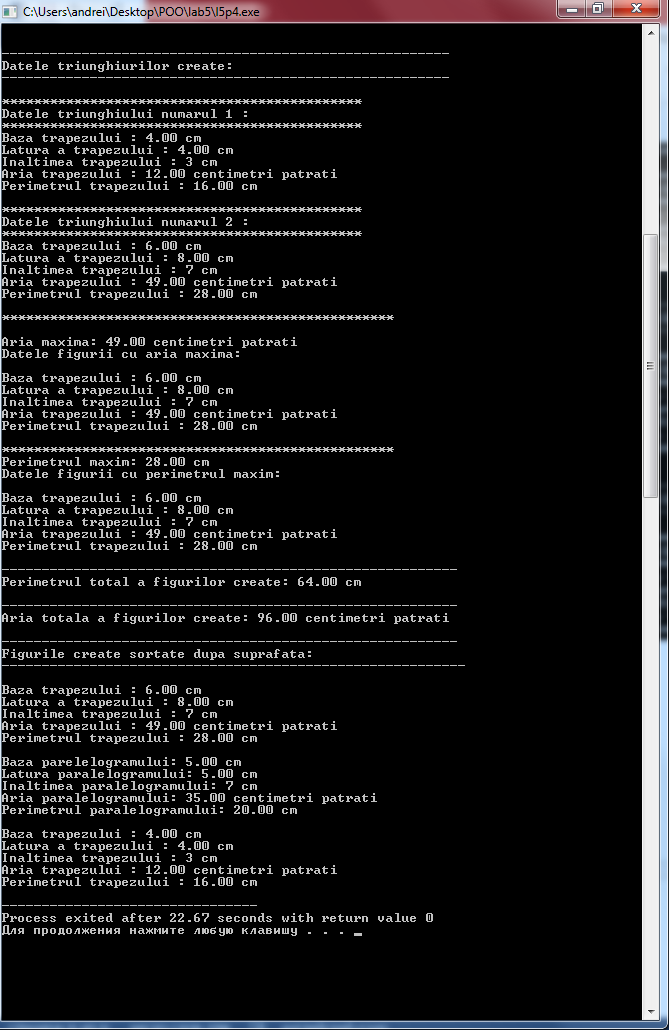
ob[i] -> afisare();

}

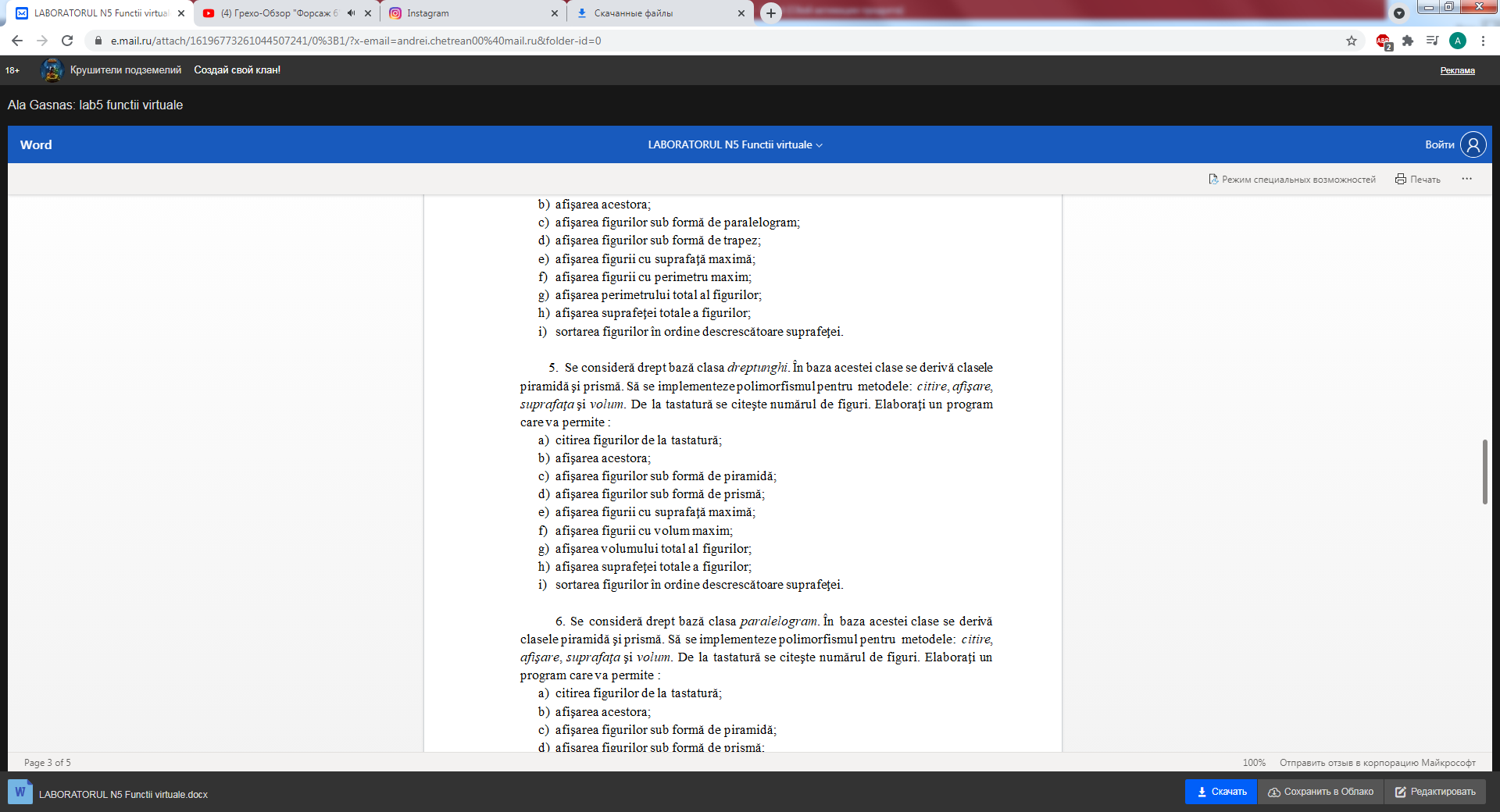
return 0;

}





**Problema 5.**



#include<bits/stdc++.h>

using namespace std;

class dreptunghi{

protected:

float num1;

float num2;

int num3;

public:

virtual void afisare() = 0;

virtual void citire() = 0;

virtual float aria() = 0;

virtual float volum() = 0;

};

class piramida: public dreptunghi{

public:

virtual void afisare();

virtual void citire();

virtual float aria();

virtual float volum();

};

class prisma: public dreptunghi{

public:

virtual void afisare();

virtual void citire();

virtual float aria();

virtual float volum();

};

void piramida::citire(){

cout << "Introduceti lungimea bazei piramidei: ";

cin >> num1;

cout << "Introduceti latime bazei piramidei: ";

cin >> num2;

cout << "Introduceti inaltimea piramidei: ";

cin >> num3;

}

void prisma::citire(){

cout << "Introduceti lungimea bazei prismei: ";

cin >> num1;

cout << "Introduceti latime bazei prismei: ";

cin >> num2;

cout << "Introduceti inaltimea prismei: ";

cin >> num3;

}

void piramida::afisare(){

cout << "\nLungimea bazei piramidei: " << num1 << " cm\n";

cout << "Latimea bazei piramidei: " << num2 << " cm\n";

cout << "Inaltimea piramidei: " << num3 << " cm\n";

cout << "Aria piramidei: " << fixed << setprecision(2) << aria() << " centimetri patrati\n";

cout << "Volumul piramidei: " << fixed << setprecision(2) << volum() << " centimetri cubi\n";

}

void prisma::afisare(){

cout << "\nLungimea bazei prismei : " << num1 << " cm\n";

cout << "Latimea bazei prismei : " << num2 << " cm\n";

cout << "Inaltimea prismei: " << num3 << " cm\n";

cout << "Aria prismei : " << fixed << setprecision(2) << aria() << " centimetri patrati\n";

cout << "Volumul prismei : " << fixed << setprecision(2) << volum() << " centimetri cubi\n";

}

float piramida::aria(){

float diagonala = num1 \* num1 + num2 \* num2;

diagonala = pow(diagonala,0.5);

float semiDiag = diagonala / 2.0;

float apot = num3 \* num3 + semiDiag \* semiDiag;

apot = pow(apot,0.5);

float perimBaza = (num1 + num2) \* 2;

float ariaLat = perimBaza \* apot / 2.0;

float ariaBaza = num1 \* num2;

return ariaLat + ariaBaza;

}

float piramida::volum(){

return num1 \* num2 \* num3 / 3.0;

}

float prisma::aria(){

return num1 \* num2 \* 2 + num2 \* num3 \* 2 + num1 \* num3 \* 2;

}

float prisma::volum(){

return num1 \* num2 \* num3;

}

int main(){

dreptunghi \*ob[100];

int choices[100];

int n;

cout << "n=";

cin >> n;

int choice;

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

cout << "\n\nIntroduceti alegerea dumneavoastra (1 sau 2): ";

cin >> choice;

if (choice == 1) {

ob[i] = new piramida;

ob[i] -> citire();

choices[i] = 1;

} else if(choice == 2) {

ob[i] = new prisma;

ob[i] -> citire();

choices[i] = 2;

} else {

cout << "Ati facut alegerea gresit";

i--;

}

}

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

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

cout << "\nDatele corpului numarul " << i + 1 << " :\n";

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

ob[i] -> afisare();

}

cout << "\n\n--------------------------------------------------------";

cout << "\nDatele piramidei create:\n";

cout << "--------------------------------------------------------\n";

int k = 1;

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

if (choices[i] == 1) {

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

cout << "\nDatele piramidei numarul " << k << " :\n";

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

ob[i] -> afisare();

k++;

}

}

k = 1;

cout << "\n\n--------------------------------------------------------";

cout << "\nDatele prismei create:\n";

cout << "--------------------------------------------------------\n";

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

if (choices[i] == 2) {

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

cout << "\nDatele prismei numarul " << k << " :\n";

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

ob[i] -> afisare();

k++;

}

}

float sMax = ob[0] -> aria();

int sMaxIndex = 0;

float vMax = ob[0] -> volum();

int vMaxIndex = 0;

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

if (ob[i] -> aria() > sMax) {

sMax = ob[i] -> aria();

sMaxIndex = i;

}

if (ob[i] -> volum() > vMax) {

vMax = ob[i] -> volum();

vMaxIndex = i;

}

}

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

cout << "\nAria maxima: " << sMax << " centimetri patrati";

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

ob[sMaxIndex] -> afisare();

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

cout << "Volumul maxim: " << vMax << " centimetri cubi";

cout << "\nDatele corpului cu volumul maxim:\n";

ob[vMaxIndex] -> afisare();

float sTotal = 0;

float vTotal = 0;

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

sTotal += ob[i] -> aria();

vTotal += ob[i] -> volum();

}

cout << "\n---------------------------------------------------------\n";

cout << "Volumul total a corpurilor create: " << vTotal << " centimetri cubi\n";

cout << "\n---------------------------------------------------------\n";

cout << "Aria totala a corpurilor create: " << sTotal << " centimetri patrati\n";

dreptunghi \*auxObj;

for (int i = 0;i < n; i++)

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

if (ob[i] -> aria() > ob[j] -> aria()) {

auxObj = ob[i];

ob[i] = ob[j];

ob[j] = auxObj;

}

cout << "\n---------------------------------------------------------\n";

cout << "Corpurile create sortate dupa suprafata:\n";

cout << "----------------------------------------------------------\n";

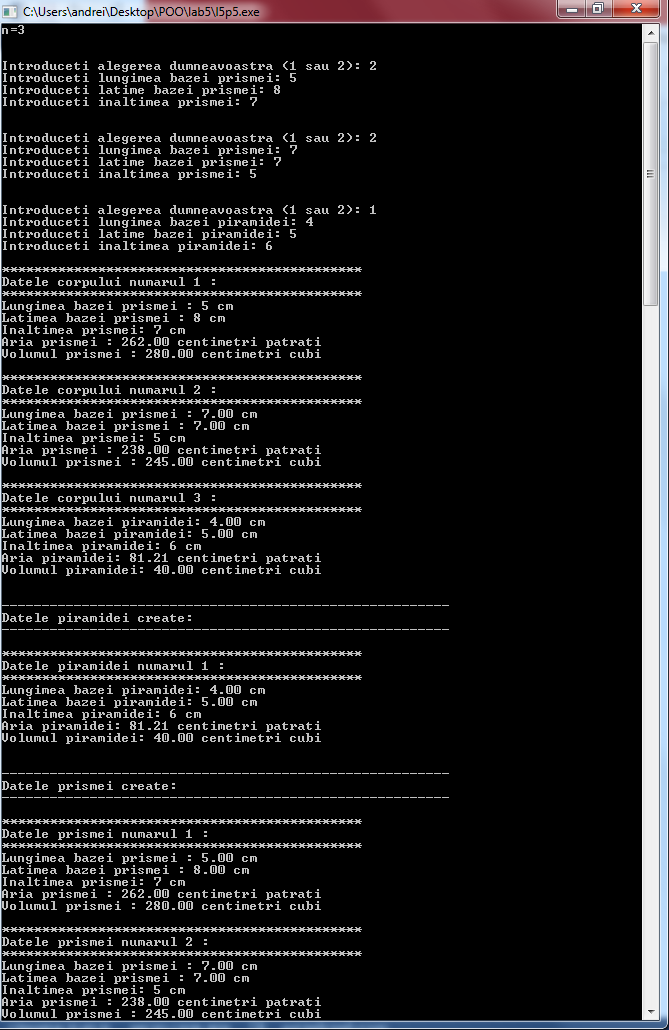
for (int i = 0;i < n; i++) {

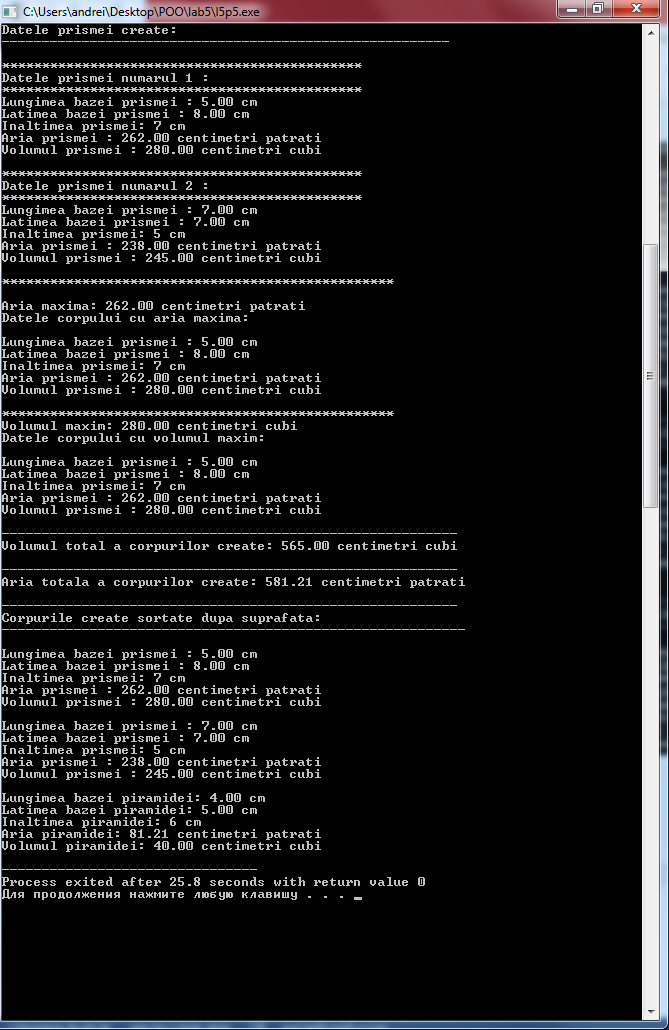
ob[i] -> afisare();

}

return 0;

}





**Problema 6.**

#include<bits/stdc++.h>

using namespace std;

class paralelogram{

protected:

float num1;

float num2;

int num3;

int num4;

public:

virtual void afisare() = 0;

virtual void citire() = 0;

virtual float aria() = 0;

virtual float volum() = 0;

};

class piramida: public paralelogram{

public:

virtual void afisare();

virtual void citire();

virtual float aria();

virtual float volum();

};

class prisma: public paralelogram{

public:

virtual void afisare();

virtual void citire();

virtual float aria();

virtual float volum();

};

void piramida::citire(){

cout << "Introduceti lungimea bazei piramidei: ";

cin >> num1;

cout << "Introduceti latime bazei piramidei: ";

cin >> num2;

cout << "Introduceti inaltimea bazei piramidei: ";

cin >> num3;

cout << "Introduceti inaltimea piramidei: ";

cin >> num4;

}

void prisma::citire(){

cout << "Introduceti lungimea bazei prismei: ";

cin >> num1;

cout << "Introduceti latime bazei prismei: ";

cin >> num2;

cout << "Introduceti inaltimea bazei prismei: ";

cin >> num3;

cout << "Introduceti inaltimea prismei: ";

cin >> num4;

}

void piramida::afisare(){

cout << "\nLungimea bazei piramidei: " << num1 << " cm\n";

cout << "Latimea bazei piramidei: " << num2 << " cm\n";

cout << "Inaltimea bazei piramidei: " << num3 << " cm\n";

cout << "Inaltimea piramidei: " << num4 << " cm\n";

cout << "Aria piramidei: " << fixed << setprecision(2) << aria() << " centimetri patrati\n";

cout << "Volumul piramidei: " << fixed << setprecision(2) << volum() << " centimetri cubi\n";

}

void prisma::afisare(){

cout << "\nLungimea bazei prismei : " << num1 << " cm\n";

cout << "Latimea bazei prismei : " << num2 << " cm\n";

cout << "Inaltimea bazei prismei: " << num3 << " cm\n";

cout << "Inaltimea prismei: " << num4 << " cm\n";

cout << "Aria prismei : " << fixed << setprecision(2) << aria() << " centimetri patrati\n";

cout << "Volumul prismei : " << fixed << setprecision(2) << volum() << " centimetri cubi\n";

}

float piramida::aria(){

float diagonala = num1 \* num1 + num2 \* num2;

diagonala = pow(diagonala,0.5);

float semiDiag = diagonala / 2.0;

float apot = num3 \* num3 + semiDiag \* semiDiag;

apot = pow(apot,0.5);

float perimBaza = (num1 + num2) \* 2;

float ariaLat = perimBaza \* apot / 2.0;

float ariaBaza = num1 \* num2;

return ariaLat + ariaBaza;

}

float piramida::volum(){

return num1 \* num3 \* num4 / 3.0;

}

float prisma::aria(){

return num1 \* num2 \* 2 + num2 \* num3 \* 2 + num1 \* num3 \* 2;

}

float prisma::volum(){

return num1 \* num3 \* num4;

}

int main(){

paralelogram \*ob[100];

int choices[100];

int n;

cout << "n=";

cin >> n;

int choice;

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

cout << "\n\nIntroduceti alegerea dumneavoastra (1 sau 2): ";

cin >> choice;

if (choice == 1) {

ob[i] = new piramida;

ob[i] -> citire();

choices[i] = 1;

} else if(choice == 2) {

ob[i] = new prisma;

ob[i] -> citire();

choices[i] = 2;

} else {

cout << "Ati facut alegerea gresit";

i--;

}

}

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

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

cout << "\nDatele corpului numarul " << i + 1 << " :\n";

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

ob[i] -> afisare();

}

cout << "\n\n--------------------------------------------------------";

cout << "\nDatele piramidei create:\n";

cout << "--------------------------------------------------------\n";

int k = 1;

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

if (choices[i] == 1) {

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

cout << "\nDatele piramidei numarul " << k << " :\n";

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

ob[i] -> afisare();

k++;

}

}

k = 1;

cout << "\n\n--------------------------------------------------------";

cout << "\nDatele prismei create:\n";

cout << "--------------------------------------------------------\n";

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

if (choices[i] == 2) {

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

cout << "\nDatele prismei numarul " << k << " :\n";

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

ob[i] -> afisare();

k++;

}

}

float sMax = ob[0] -> aria();

int sMaxIndex = 0;

float vMax = ob[0] -> volum();

int vMaxIndex = 0;

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

if (ob[i] -> aria() > sMax) {

sMax = ob[i] -> aria();

sMaxIndex = i;

}

if (ob[i] -> volum() > vMax) {

vMax = ob[i] -> volum();

vMaxIndex = i;

}

}

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

cout << "\nAria maxima: " << sMax << " centimetri patrati";

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

ob[sMaxIndex] -> afisare();

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

cout << "Volumul maxim: " << vMax << " centimetri cubi";

cout << "\nDatele corpului cu volumul maxim:\n";

ob[vMaxIndex] -> afisare();

float sTotal = 0;

float vTotal = 0;

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

sTotal += ob[i] -> aria();

vTotal += ob[i] -> volum();

}

cout << "\n---------------------------------------------------------\n";

cout << "Volumul total a corpurilor create: " << vTotal << " centimetri cubi\n";

cout << "\n---------------------------------------------------------\n";

cout << "Aria totala a corpurilor create: " << sTotal << " centimetri patrati\n";

paralelogram \*auxObj;

for (int i = 0;i < n; i++)

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

if (ob[i] -> aria() > ob[j] -> aria()) {

auxObj = ob[i];

ob[i] = ob[j];

ob[j] = auxObj;

}

cout << "\n---------------------------------------------------------\n";

cout << "Corpurile create sortate dupa suprafata:\n";

cout << "----------------------------------------------------------\n";

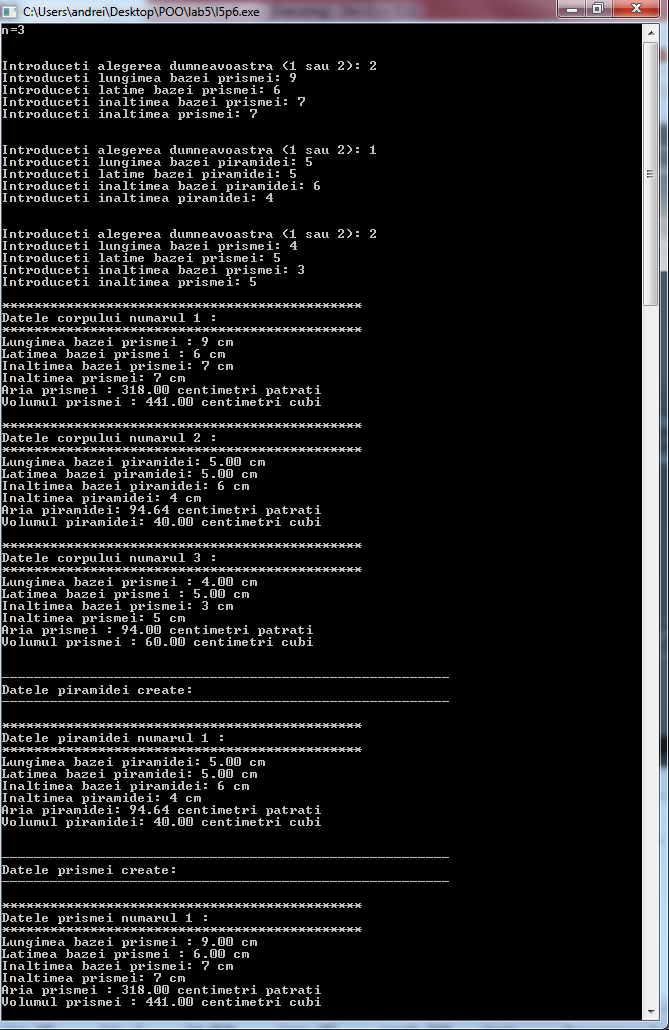
for (int i = 0;i < n; i++) {

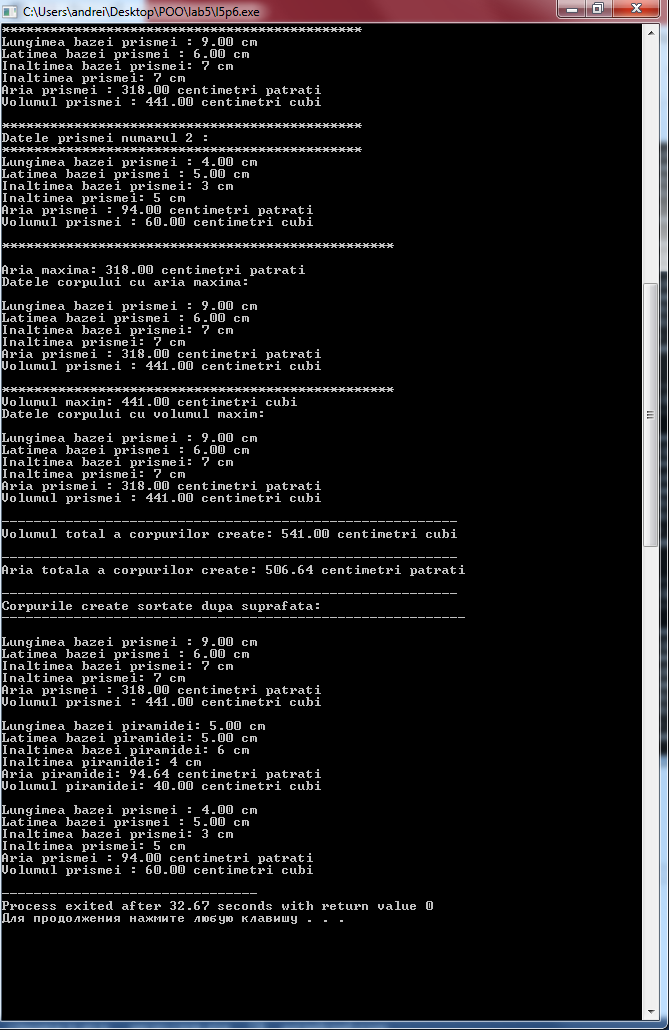
ob[i] -> afisare();

}

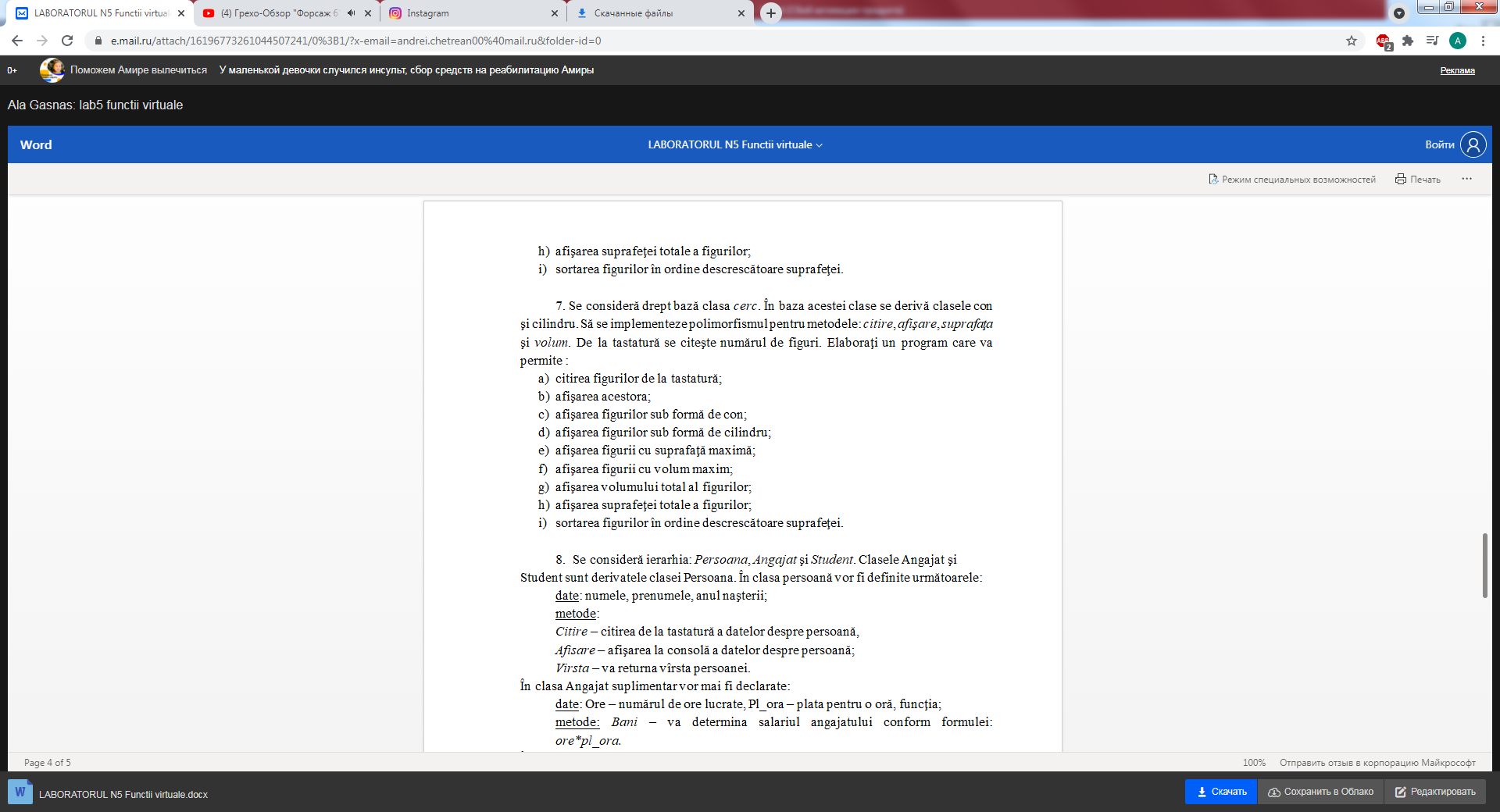
return 0;

}





**Problema 7.**



#include<bits/stdc++.h>

using namespace std;

class cerc{

protected:

float num1;

float num2;

public:

virtual void afisare() = 0;

virtual void citire() = 0;

virtual float aria() = 0;

virtual float volum() = 0;

};

class con: public cerc{

public:

virtual void afisare();

virtual void citire();

virtual float aria();

virtual float volum();

};

class cilindru: public cerc{

public:

virtual void afisare();

virtual void citire();

virtual float aria();

virtual float volum();

};

void con::citire(){

cout << "Introduceti raza bazei conului: ";

cin >> num1;

cout << "Introduceti inaltimea conului: ";

cin >> num2;

}

void cilindru::citire(){

cout << "Introduceti raza bazei cilindrului: ";

cin >> num1;

cout << "Introduceti raza bazei cilindrului: ";

cin >> num2;

}

void con::afisare(){

cout << "\nRaza bazei conului: " << num1 << " cm\n";

cout << "Inaltimea conului: " << num2 << " cm\n";

cout << "Aria conului: " << fixed << setprecision(2) << aria() << " centimetri patrati\n";

cout << "Volumul conului: " << fixed << setprecision(2) << volum() << " centimetri cubi\n";

}

void cilindru::afisare(){

cout << "\nRaza bazei cilindrului: " << num1 << " cm\n";

cout << "Inaltimea cilindrului: " << num2 << " cm\n";

cout << "Aria cilindrului: " << fixed << setprecision(2) << aria() << " centimetri patrati\n";

cout << "Volumul cilindrului: " << fixed << setprecision(2) << volum() << " centimetri cubi\n";

}

float con::aria(){

float generatoare = num1 \* num1 + num2 \* num2;

generatoare = sqrt(generatoare);

float ariaBaza = 3.14 \* num1 \* num1;

float ariaLat = 3.14 \* num1 \* generatoare;

return ariaBaza + ariaLat;

}

float con::volum(){

return 3.14 \* num1 \* num1 \* num2 / 3.0;

}

float cilindru::aria(){

float ariaBaza = 3.14 \* num1 \* num1;

float ariaLat = 2 \* 3.14 \* num1 \* num2;

return ariaBaza \* 2 + ariaLat;

}

float cilindru::volum(){

return 3.14 \* num1 \* num1 \* num2;

}

int main(){

cerc \*ob[100];

int choices[100];

int n;

cout << "n=";

cin >> n;

int choice;

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

cout << "\n\nIntroduceti alegerea dumneavoastra (1 sau 2): ";

cin >> choice;

if (choice == 1) {

ob[i] = new con;

ob[i] -> citire();

choices[i] = 1;

} else if(choice == 2) {

ob[i] = new cilindru;

ob[i] -> citire();

choices[i] = 2;

} else {

cout << "Ati facut alegerea gresit";

i--;

}

}

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

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

cout << "\nDatele corpului numarul " << i + 1 << " :\n";

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

ob[i] -> afisare();

}

cout << "\n\n--------------------------------------------------------";

cout << "\nDatele conurilor create:\n";

cout << "--------------------------------------------------------\n";

int k = 1;

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

if (choices[i] == 1) {

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

cout << "\nDatele conului numarul " << k << " :\n";

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

ob[i] -> afisare();

k++;

}

}

k = 1;

cout << "\n\n--------------------------------------------------------";

cout << "\nDatele cilindrurilor create:\n";

cout << "--------------------------------------------------------\n";

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

if (choices[i] == 2) {

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

cout << "\nDatele cilindrului numarul " << k << " :\n";

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

ob[i] -> afisare();

k++;

}

}

float sMax = ob[0] -> aria();

int sMaxIndex = 0;

float vMax = ob[0] -> volum();

int vMaxIndex = 0;

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

if (ob[i] -> aria() > sMax) {

sMax = ob[i] -> aria();

sMaxIndex = i;

}

if (ob[i] -> volum() > vMax) {

vMax = ob[i] -> volum();

vMaxIndex = i;

}

}

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

cout << "\nAria maxima: " << sMax << " centimetri patrati";

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

ob[sMaxIndex] -> afisare();

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

cout << "Volumul maxim: " << vMax << " centimetri cubi";

cout << "\nDatele corpului cu volumul maxim:\n";

ob[vMaxIndex] -> afisare();

float sTotal = 0;

float vTotal = 0;

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

sTotal += ob[i] -> aria();

vTotal += ob[i] -> volum();

}

cout << "\n---------------------------------------------------------\n";

cout << "Volumul total a corpurilor create: " << vTotal << " centimetri cubi\n";

cout << "\n---------------------------------------------------------\n";

cout << "Aria totala a corpurilor create: " << sTotal << " centimetri patrati\n";

cerc \*auxObj;

for (int i = 0;i < n; i++)

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

if (ob[i] -> aria() > ob[j] -> aria()) {

auxObj = ob[i];

ob[i] = ob[j];

ob[j] = auxObj;

}

cout << "\n---------------------------------------------------------\n";

cout << "Corpurile create sortate dupa suprafata:\n";

cout << "----------------------------------------------------------\n";

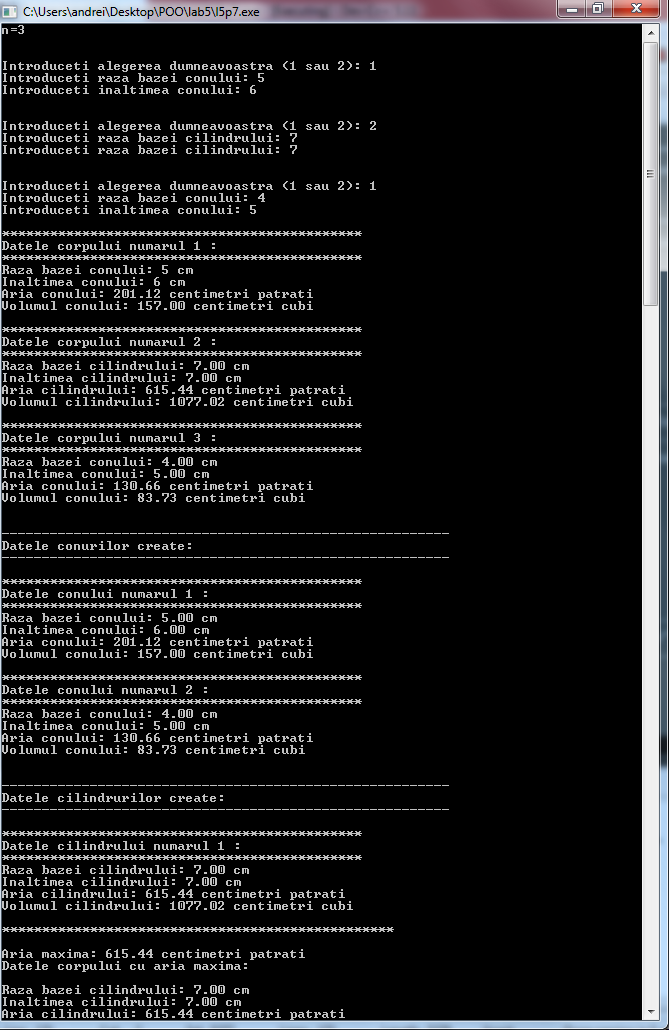
for (int i = 0;i < n; i++) {

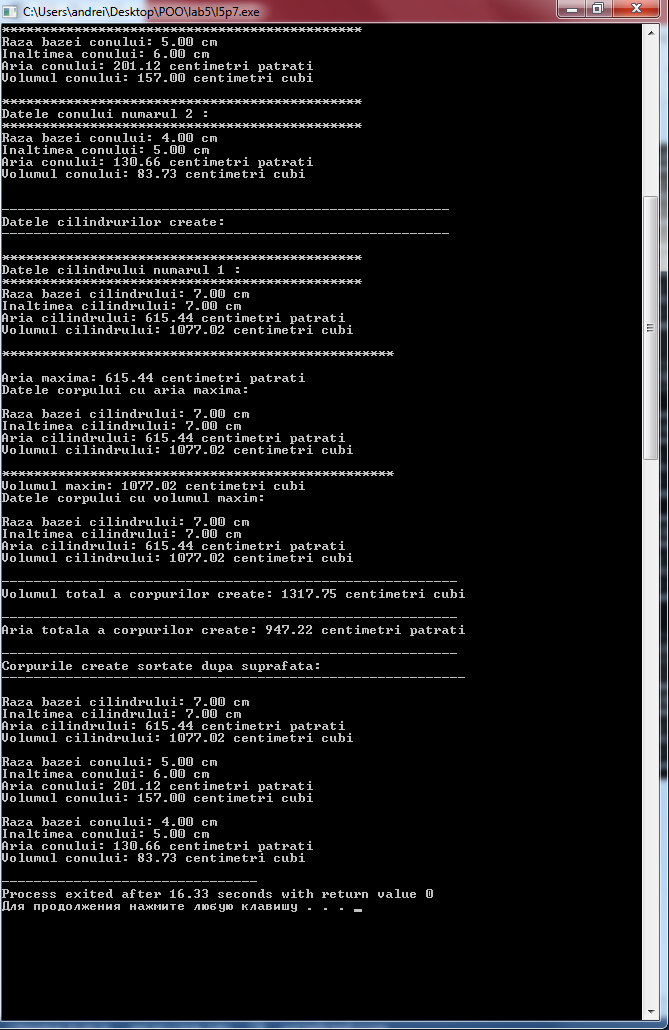
ob[i] -> afisare();

}

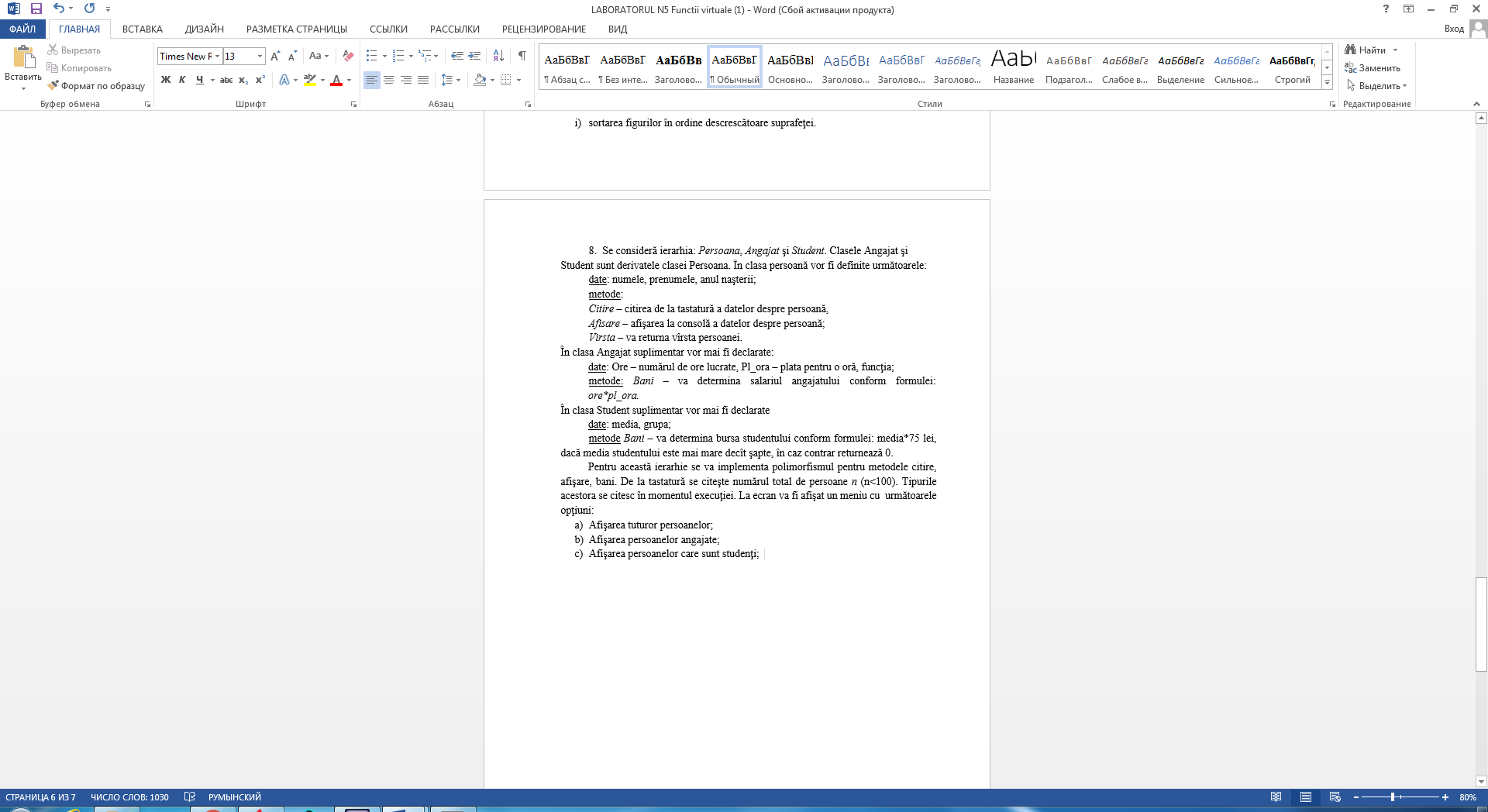
return 0;

}





**Problema 8.**



#include<bits/stdc++.h>

using namespace std;

class Persoana{

protected:

string nume;

string prenume;

int an\_nastere;

public:

virtual void afisare();

virtual void citire();

virtual int varsta();

virtual float bani() = 0;

};

class Angajat: public Persoana{

private:

int nr\_ore;

float plata;

public:

virtual void citire();

virtual void afisare();

virtual float bani();

};

class Student: public Persoana{

private:

float media;

string grupa;

public:

virtual void afisare();

virtual void citire();

virtual float bani();

};

void Persoana::citire(){

cout << "Introduceti numele persoanei: ";

cin >> nume;

cout << "Introduceti prenumele persoanei: ";

cin >> prenume;

cout << "Introduceti anul nasterii a angajatului: ";

cin >> an\_nastere;

}

void Persoana::afisare(){

cout << "\nNumele persoanei: " << nume << endl;

cout << "Prenumele persoanei: " << prenume << endl;

cout << "Anul nasterii: " << an\_nastere << endl;

cout << "Varsta: " << varsta() << " ani\n";

}

int Persoana::varsta(){

return 2021 - an\_nastere;

}

void Angajat::citire(){

Persoana::citire();

cout << "Introduceti numarul de ore: ";

cin >> nr\_ore;

cout << "Introduceti plata pentru o ora: ";

cin >> plata;

}

void Angajat::afisare(){

Persoana::afisare();

cout << "\nNumarul de ore lucrate: " << nr\_ore << endl;

cout << "Plata pentru o ora: " << plata << endl;

cout << "Salariu: " << bani() << " lei\n";

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

}

float Angajat::bani(){

return nr\_ore \* plata;

}

void Student::citire(){

Persoana::citire();

cout << "Introduceti media studentului: ";

cin >> media;

cout << "Introduceti grupa studentului: ";

cin >> grupa;

}

void Student::afisare(){

Persoana::afisare();

cout << "\nMedia studentului: " << media << endl;

cout << "Grupa studentului: " << grupa << endl;

cout << "Bursa: " << bani() << " lei\n";

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

}

float Student::bani(){

if (media > 7.5) {

return media \* 75;

}

return 0;

}

int main(){

Persoana \*ob[100];

int choices[100];

int n;

cout << "n=";

cin >> n;

int choice;

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

cout << "\n\nIntroduceti alegerea dumneavoastra (1 sau 2): ";

cin >> choice;

if (choice == 1) {

ob[i] = new Angajat;

ob[i] -> citire();

choices[i] = 1;

} else if(choice == 2) {

ob[i] = new Student;

ob[i] -> citire();

choices[i] = 2;

} else {

cout << "Ati facut alegerea gresit";

i--;

}

}

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

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

cout << "\nDatele persoanei numarul " << i + 1 << " :\n";

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

ob[i] -> afisare();

}

cout << "\n\n--------------------------------------------------------";

cout << "\nDatele angajatior:\n";

cout << "--------------------------------------------------------\n";

int k = 1;

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

if (choices[i] == 1) {

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

cout << "\nDatele angajatului numarul " << k << " :\n";

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

ob[i] -> afisare();

k++;

}

}

k = 1;

cout << "\n\n--------------------------------------------------------";

cout << "\nDatele studentilor:\n";

cout << "--------------------------------------------------------\n";

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

if (choices[i] == 2) {

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

cout << "\nDatele studentului numarul " << k << " :\n";

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

ob[i] -> afisare();

k++;

}

}

float bMax = ob[0] -> bani();

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

if (ob[i] -> bani() > bMax) {

bMax = ob[i] -> bani();

}

}

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

cout << "\nSuma cea mai mare pe care o primeste o persoana: " << bMax << " lei";

cout << "\nDatele persoanelor/persoanei ce primesc aceasta suma:\n";

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

if (ob[i] -> bani () == bMax) {

ob[i] ->afisare();

}

}

float bMin = ob[0] -> bani();

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

if (ob[i] -> bani() < bMin) {

bMin = ob[i] -> bani();

}

}

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

cout << "\nSuma cea mai mica pe care o primeste o persoana: " << bMin << " lei";

cout << "\nDatele persoanelor/persoanei ce primesc aceasta suma:\n";

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

if (ob[i] -> bani() == bMin) {

ob[i] ->afisare();

}

}

int anMin = ob[0] -> varsta();

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

if (ob[i] -> varsta() < anMin){

anMin = ob[i] -> varsta();

}

}

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

cout << "\nCea mai tanara varsta: " << anMin << " ani";

cout << "\nDatele persoanelor/persoanei cu aceasta varsta:\n";

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

if (ob[i] -> varsta() == anMin) {

ob[i] -> afisare();

}

}

Persoana \*auxObj;

for (int i = 0;i < n; i++)

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

if (ob[i] -> bani() > ob[j] -> bani()) {

auxObj = ob[i];

ob[i] = ob[j];

ob[j] = auxObj;

}

cout << "\n---------------------------------------------------------\n";

cout << "Persoanele sortate dupa banii primiti:\n";

cout << "----------------------------------------------------------\n";

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

ob[i] -> afisare();

}

for (int i = 0;i < n; i++)

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

if (ob[i] -> varsta() < ob[j] -> varsta()) {

auxObj = ob[i];

ob[i] = ob[j];

ob[j] = auxObj;

}

cout << "\n---------------------------------------------------------\n";

cout << "Persoanele sortate dupa varsta:\n";

cout << "----------------------------------------------------------\n";

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

ob[i] -> afisare();

}

return 0;

}

