

Test 5 POO

```
#include <iostream>

#define pi 3.1415

using namespace std;

class Cerc;

class Dreptunghi
{
    int lungime, latime;
public:
    Dreptunghi()
    {
        cout<<"\nS-a apelat constructorul vid.\n";
        cout<<"\nIntroduceti latimea dreptunghiului: ";
        cin>>latime;
        while(latime < 0){
            cout<<" Latura nu poate fi negativa\n"<<"Va rog sa introduceti un numar pozitiv corespunzand latimii dreptunghiului: ";
            cin>>latime;
        }
        cout<<"\nIntroduceti lungimea dreptunghiului: ";
        cin>>lungime;
        while(lungime < 0){
            cout<<" Latura nu poate fi negativa\n"<<"Va rog sa introduceti un numar pozitiv corespunzand lungimii dreptunghiului: ";
            cin>>lungime;
        }
    }
}
```

```
Dreptunghi(int lungime1, int latime1)
{
    cout<<"S-a apelat constructorul cu parametrii.\n";
    lungime = lungime1;
    latime = latime1;
}

Dreptunghi(Dreptunghi&x)
{
    cout<<"S-a apelat constructorul de copiere.\n";
    lungime = x.lungime;
    latime = x.latime;
}

~Dreptunghi()
{
    cout<<"\nS-a apelat destructorul.\n";
}

friend float aria(Dreptunghi);

friend class Cerc;

};

class Cerc
{
    float raza;

public:
    void transforma(Dreptunghi);
    friend float aria(Cerc);
};

float aria(Dreptunghi drept)
```

```
{
    float arie;

    arie = drept.lungime*drept.latime;

    return arie;
}
```

```
void Cerc::transforma(Dreptunghi drept)
{
    raza=(drept.lungime+drept.latime)/2;
}
```

```
float aria(Cerc cerc)
{
    float arie;

    arie=pi*cerc.raza*cerc.raza;

    return arie;
}
```

```
int main(void)
{
    float lungime,latime,arie_drept1,arie_drept2,arie_drept3, arie_c1,arie_c2,arie_c3;
    Cerc c1,c2,c3;
    char tasta_oprire;

    cout<<"\nAriile unor dreptunghiuri si a unor cercuri obtinute din dreptunghiuri\n";

    do
    {
        Dreptunghi drept1;

        cout<<"\nIntroduceti lungimea dreptunghiului 2: ";

        cin>>lungime;

        cout<<"\nIntroduceti lungimea dreptunghiului 2: ";
```

```

cin>>latime;

Dreptunghi drept2(lungime,latime);

Dreptunghi drept3(drept2);


arie_drept1=aria(drept1);

cout<<"\nDreptunghiul drept1 are aria: "<<arie_drept1<<endl;

c1.transforma(drept1);

arie_c1=aria(c1);

cout<<"\nCercul c1 transformat din dreptunghiul drept1 are aria: "<<arie_c1<<endl;

arie_drept2=aria(drept2);

cout<<"\nDreptunghiul drept2 are aria="<<arie_drept2<<endl;

c2.transforma(drept2);

arie_c2=aria(c2);

cout<<"\nCercul c2 transformat din dreptunghiul drept2 are aria: "<<arie_c2<<endl;

arie_drept3=aria(drept3);

cout<<"\nDreptunghiul drept3 are aria="<<arie_drept3<<endl;

c3.transforma(drept3);

arie_c3=aria(c3);

cout<<"\nCercul c3 transformat din dreptunghiul drept3 are aria: "<<arie_c3<<endl;


cout<<"\nDoriti sa continuati:('d' = DA/ 'n' = NU) ";

cin>>tasta_oprire;

while(tasta_oprire != 'd' && tasta_oprire != 'n'){

cout<<"\nAi tastat o optiune invalida\n"<<"Tasteaza 'd' pentru DA si 'n' pentru NU: ";

cin>>tasta_oprire;

}

}while(tasta_oprire != 'n');

return 0;

}

```