

PROGRAMARE ORIENTATĂ OBIECT

[Titlul proiectului]

Elev: [nume si prenume elev], [clasa]

Profesor: [nume si prenume prof]

Cuprins

generat automat

Toate titlurile se vor formata cu stilul Heading 1 iar subtitlurile cu Heading 2

Generare cuprins automat - Meniul Reference – Table of Contents

Table of Contents

Cuprins	2
generat automat	2
Programarea orientată obiect. Noțiuni teoretice.....	3
CLASA [titlul clasei]	4
Descrierea clasei: fișierul "[nume_clasa.h]"	4

Programarea orientată obiect. Noțiuni teoretice.

Programarea orientată obiect (POO sau OOP – eng. Project Oriented Programming) este ...

Noțiuni fundamentale:

- Clasă
- Constructor
- Destructor
- Specificatori de acces (semnificație, declarare, mod de utilizare)
 - Private:
 - Public:
- Funcții *friend*¹

¹ Funcție prietena cu clasa, fara sa fie membru

CLASA [titlul clasei]

Descrierea clasei: fișierul "[nume_clasa.h]"

Clasa Polinom implementează polinoame în format algebric

P(x) =

```
#include<iostream>
#include<fstream>
using namespace std;
struct nod
{
    int coef,exp;
    nod *urm;
};
class polinom
{
    int n;
    nod *p,*u;
public:
    polinom() {n=0; p=NULL; u=NULL;};
    polinom(polinom&);
    ~polinom();
    friend polinom& operator+(polinom&, polinom&);
    friend istream& operator>>(istream &, polinom &);
    friend ostream& operator<<(ostream &, polinom &);
    int val(int);
};
```

Specificatii de formatare:

1. Fontul folosit in descriere (nu in secventele de cod) : Calibri (Body), 11
2. Fontul folosit in secventele de cod : Courier New, 11
3. Paragrafele spatiate la 6px inainte si dupa (cu exceptia codului)
4. Codul indentat cu Indent – Special first line – 1cm
5. Fiecare capitol incepe pe o pagina noua (insert - page-break la sfarsit de capitol)
6. Pagina formata cu 2.5 cm la interior, 2cm in rest
7. Paginare fata-verso cu numerotare in exterior (fara prima pagina)
8. Cuprins generat automat
9. Bibliografie pe ultima pagina