

Laborator 5

Implementati clasa Fractie avand urmatoarii operatori:

```
class Fractie
{
    int a; //numarator
    int b; //numitor
public:
    Fractie(int aa=0,int bb=1); // constructor cu parametrii impliciti

    double getValoare(); //cat face a/b
    Fractie getInv(); //b/a

    void setdata(int,int);//modifica valorile numaratorului si numitorului

    float getA();//returneaza numaratorul
    float getB();//returneaza numitorul

    friend Fractie operator +(const Fractie&, const Fractie&); //supradefinire operator adunare
    friend Fractie operator *( const Fractie&, const Fractie&);
    friend Fractie operator -(const Fractie&); //transforma numerele in inversul lor. ex: 8 -> -8

    Fractie& operator +=( const Fractie& a);

    bool operator ==(const Fractie&);//supradefinire operator de egalitate
    bool operator !=(const Fractie& x);// supradefinire operator diferit
    // pot sa folosesc in implementare operatorul == implementat anterior {return(!(*this==x));}

    bool operator <(const Fractie&);// supradefinire operator <
    bool operator <=(const Fractie&);// supradefinire operator <=

    Fractie& operator++(); // forma prefixata
    Fractie operator++(int); // forma postfixata

    friend std::istream& operator>>(std::istream& in, Fractie& z); // operator pentru citire
    friend std::ostream& operator<<(std::ostream& out, const Fractie& z); //operator pentru afisare
};
```

TOTAL: 9 puncte + 1 punct din oficiu

- Implementare 5 puncte
- Testare in main: 4 puncte