

I. zárthelyi dolgozat
Programozás módszertan II.
2010/2011. I.félév
2010. október 13.

1. feladat:

Használja az alábbi osztály-deklarációt!

```
class Teglatest
{
private:
    double a, b, c, felsz, terf; //Az "a,b,c" élhosszúságú Teglatest, annak felszíne és térfogata
    void TeglatestFelszin(); //a felszint (felsz adattag értékét) meghatározó tagfüggvény
    void TeglatestTerfogat(); //a térfogatot (terf adattag értékét) meghatározó tagfüggvény

public:
    Teglatest( ); //alapértelmezett konstruktor: a=b=c=1, a többi ebből számolandó
    Teglatest(double a0, double b0, double c0); //paraméteres konstruktor
    Teglatest(const Teglatest& T); //másoló konstruktor
    double GetFelszin();
    double GetTerfogat();
    double testatlo(); //a Teglatest testátlójának hosszát adja vissza
    Teglatest operator=(const Teglatest& T); //értékadás operátor
    bool operator==(const Teglatest& T) const; //egyenlőség-vizsgálat
    bool operator!=(const Teglatest& T) const;
    friend Teglatest operator*(const Teglatest& T, const int& szam); //Olyan Teglatesttel tér
    //vissza, amelynek éleinek hosszai a T Teglatest élei hosszainak szam-szorosai
    friend ostream& operator<<(ostream& os, const Teglatest& T) //a cout<<T<< endl;
    //működéséhez
};
```

Írja meg a tagfüggvényeket! A main.cpp állományban írja meg a main() főfüggvényt, mutassa be a fenti tagfüggvények hívását!

2. feladat: Egészítse ki az alábbi programot úgy, hogy helyesen működjön.

```
#include <iostream>
using namespace std;

class Racionalis{
private:
    int szamlalo, nevezo;
    void egyszerusit(){
        if(nevezo<0){szamlalo*=-1;nevezo*=-1;}
        int nagyobb=szamlalo, kisebb=nevezo, tmp;
        if(nagyobb<0) nagyobb *=-1;
```

```

        if(kisebb<0)kisebb *=-1;

        while(nagyobb!=0){
            if(nagyobb<kisebb){
                tmp=nagyobb;
                nagyobb=kisebb;
                kisebb=tmp;
            }
            nagyobb-=kisebb;
        }
        szamlalo /= kisebb;
        nevezo /= kisebb;
    }
public:
    Racionalis(){};
    Racionalis(int sz0, int n0){
        szamlalo=sz0;
        nevezo=n0;
        egyszerusit();
    }

    Racionalis operator+(const Racionalis &j) const {
        return Racionalis(szamlalo*j.nevezo+j.szamlalo*nevezo,nevezo*j.nevezo);
    }

    bool operator==(const Racionalis &j) const {
        return(nevezo==j.nevezo && szamlalo==j.szamlalo);
    }

    friend ostream& operator<<(ostream &os, const Racionalis &r);

};

ostream& operator<<(ostream &os, const Racionalis &r){
    os<<r.szamlalo<<" / "<<r.nevezo;
    return os;}

int main(){
    Racionalis x(-2,4), y(1,5), z;
    z=x+y;
    cout<<z<<endl;
    if(x==y)cout<<"egyenlo!"<<endl;
    else cout<<"nem egyenlo!"<<endl;
    Racionalis a(4,5), b(3,4);
    a*=b; //tagfüggvénnyel oldja meg!!
    cout<<a<<endl;
    cout<<a-3<<endl; //tagfüggvénnyel oldja meg!!

```

```
cout<<b/2<<endl; //tagfüggvénnnyel oldja meg!!  
cout<<5/b<<endl; //friend (nem tag)függvénnnyel oldja meg!!  
a*=4; //friend (nem tag)függvénnnyel oldja meg!!  
cout<<a<<endl;  
system("pause");  
return 0;  
}
```