using System;

public class Fraction

{

public Fraction(int numerator, int denominator)

{

Console.WriteLine("In Fraction Constructor( int, int) ");

this.numerator = numerator;

this.denominator = denominator;

}

public Fraction(int wholeNumber)

{

Console.WriteLine("In Fraction Constructor( int )");

numerator = wholeNumber;

denominator = 1;

}

public static implicit operator Fraction(int theInt)

{

Console.WriteLine(" In implicit conversion to Fraction");

return new Fraction(theInt);

}

public static explicit operator int(Fraction theFraction)

{

Console.WriteLine("In explicit conversion to int");

return theFraction.numerator / theFraction.denominator;

}

public static bool operator ==(Fraction lhs, Fraction rhs)

{

Console.WriteLine("In operator ==");

if (lhs.numerator == rhs.numerator &&

lhs.denominator == rhs.denominator)

{

return true;

}

// thực hiện khi hai phân số không bằng nhau

return false;

}

public static bool operator !=(Fraction lhs, Fraction rhs)

{

Console.WriteLine("In operator !=");

return !(lhs == rhs);

}

public override bool Equals(object o)

{

Console.WriteLine("In method Equals");

if (!(o is Fraction))

{

return false;

}

return this == (Fraction)o;

}

// public static Fraction operator +(Fraction lhs, Fraction rhs)

// {

// Console.WriteLine("In operator +");

// if (lhs.denominator == rhs.denominator)

// {

// return new Fraction(lhs.numerator + rhs.numerator, lhs.denominator);

// }

// //thực hiện khi hai mẫu số khộng bằng nhau

// int firstProduct = lhs.numerator \* rhs.denominator;

// int secondProduct = rhs.numerator \* lhs.denominator;

// return new Fraction(firstProduct + secondProduct,

// lhs.denominator \* rhs.denominator);

// }

public static Fraction operator + (Fraction lhs, Fraction rhs)

{

Console.WriteLine("In operator + ");

if(lhs.denominator == rhs.denominator)

{

return new Fraction(lhs.numerator + rhs.numerator, lhs.denominator);

}

// thực hiện khi hai mẫu só không bằng nhau

int firstProduct = lhs.numerator \* rhs.denominator;

int secondProduct = lhs.denominator + rhs.numerator;

return new Fraction(firstProduct + secondProduct,

lhs.denominator \* rhs.denominator);

}

public override string ToString()

{

string s = numerator.ToString() + "/" + denominator.ToString();

return s;

}

//biến thành viên lưu tử số và mẫu số

private int numerator;

private int denominator;

}

public class Tester

{

static void Main()

{

Fraction f1 = new Fraction(3, 4);

Console.WriteLine("f1:{0}", f1.ToString());

Fraction f2 = new Fraction(2, 4);

Console.WriteLine("f2:{0}", f2.ToString());

Fraction f3 = f1 + f2;

Console.WriteLine("f1 + f2 = f3:{0}", f3.ToString());

Fraction f4 = f3 + 5;

Console.WriteLine("f4 = f3 + 5:{0}", f4.ToString());

Fraction f5 = new Fraction(2, 4);

if (f5 == f2)

{

Console.WriteLine("f5:{0}==f2:{1}",

f5.ToString(), f2.ToString());

}

}

}