

Московский Авиационный Институт
(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Лабораторная работа 0 по курсу ООП: основы программирования на языке C#

0.РЕАЛИЗАЦИЯ КЛАССА

Работу выполнил:

М8О-205Б-21 Титеев Р.М.

Руководитель: _____/Кузнецова С.В.

(подпись)

Дата: ____ октября 2022

Реализация класса

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace lab_0
8 {
9     class Fraction
10    {
11        private int numerator;
12        private int denominator;
13        public Fraction(){
14            this.numerator = 1;
15            this.denominator = 1;
16        }
17        private static int gcd(int a, int b)
18        {
19            while (b != 0)
20            {
21                int temp = b;
22                b = a % b;
23                a = temp;
24            }
25            return a;
26        }
27
28        private static int lcm(int a, int b)
29        {
30            return (a / gcd(a, b)) * b;
31        }
32
33        public Fraction(int n, int d){
34            this.numerator = n;
35            if (d!=0){
36                this.denominator = d;
37            }
38            else{
39                throw new NullReferenceException("Denominator can't be 0");
40            }
41        }
42        public string view(){
43            return $"({this.numerator}/{this.denominator})";
44        }
45        public void reverse(){
```

```

46         int a = this.numerator;
47         int b = this.denominator;
48         this.numerator = b;
49         this.denominator = a;
50     }
51     public void negative(){
52         this.numerator = this.numerator*(-1);
53     }
54
55     public int get_integer_part(){
56         return (int)(this.numerator/this.denominator);
57     }
58
59     // Multiply
60     public static Fraction operator *(Fraction f1, int num)
61     {
62         return (new Fraction(f1.numerator*num, f1.denominator));
63     }
64     public static Fraction operator *(int num, Fraction f1)
65     {
66         return (new Fraction(f1.numerator*num, f1.denominator));
67     }
68     public static Fraction operator *(Fraction f1, Fraction f2)
69     {
70         return (new Fraction(f1.numerator*f2.numerator, f1.denominator*f2.
71             denominator));
72     }
73
74     // Sum
75     public static Fraction operator +(Fraction f1, int num)
76     {
77         return (new Fraction(f1.numerator + num*f1.denominator, f1.
78             denominator));
79     }
80     public static Fraction operator +(int num, Fraction f1)
81     {
82         return (new Fraction(f1.numerator + num*f1.denominator, f1.
83             denominator));
84     }
85     public static Fraction operator +(Fraction f1, Fraction f2)
86     {
87         int l = lcm(f1.denominator, f2.denominator);
88         int m1 = l/f1.denominator;
89         int m2 = l/f2.denominator;
90         return (new Fraction(f1.numerator*m2 + f2.numerator*m1, l));
91     }
92
93     public static Fraction operator -(Fraction f1, int num)

```

```

91     {
92         return (new Fraction(f1.numerator - num*f1.denominator, f1.
            denominator));
93     }
94     public static Fraction operator -(int num, Fraction f1)
95     {
96         return (new Fraction(f1.numerator - num*f1.denominator, f1.
            denominator));
97     }
98     public static Fraction operator -(Fraction f1, Fraction f2)
99     {
100         int l = lcm(f1.denominator, f2.denominator);
101         int m1 = l/f1.denominator;
102         int m2 = l/f2.denominator;
103         return (new Fraction(f1.numerator*m2 - f2.numerator*m1, l));
104     }
105
106     // Power
107     public static Fraction operator ^(Fraction f1, int num)
108     {
109         int n = (int)System.Math.Pow(f1.numerator, num);
110         int d = (int)System.Math.Pow(f1.denominator, num);
111         return (new Fraction(n, d));
112     }
113 }
114 internal class Program
115 {
116     static void Main(string[] args)
117     {
118         Fraction f1 = new Fraction(1,2);
119
120         Fraction f2 = new Fraction(4,5);
121
122         Fraction f3 = new Fraction(4,0);
123
124         Console.WriteLine($"f1 = {f1.view()}, f2 = {f2.view()}");
125
126         Console.WriteLine($"{f1.view()} + {f2.view()} = {(f1+f2).view()}")
            ;
127         Console.WriteLine($"{f1.view()} + {3} = {(f1+3).view()}");
128         Console.WriteLine($"{f1.view()} * {f2.view()} = {(f1*f2).view()}")
            ;
129         Console.WriteLine($"{-3} * {f2.view()} = {((-3)*f2).view()}");
130         Console.WriteLine($"{f2.view()} - {f1.view()} = {(f2-f1).view()}")
            ;
131         Console.WriteLine($"{f1.view()} ^ {2} = {(f1^2).view()}");
132         Console.WriteLine($"The integer part of {(f1+f2).view()}: {(f1+f2)
            .get_integer_part()}");

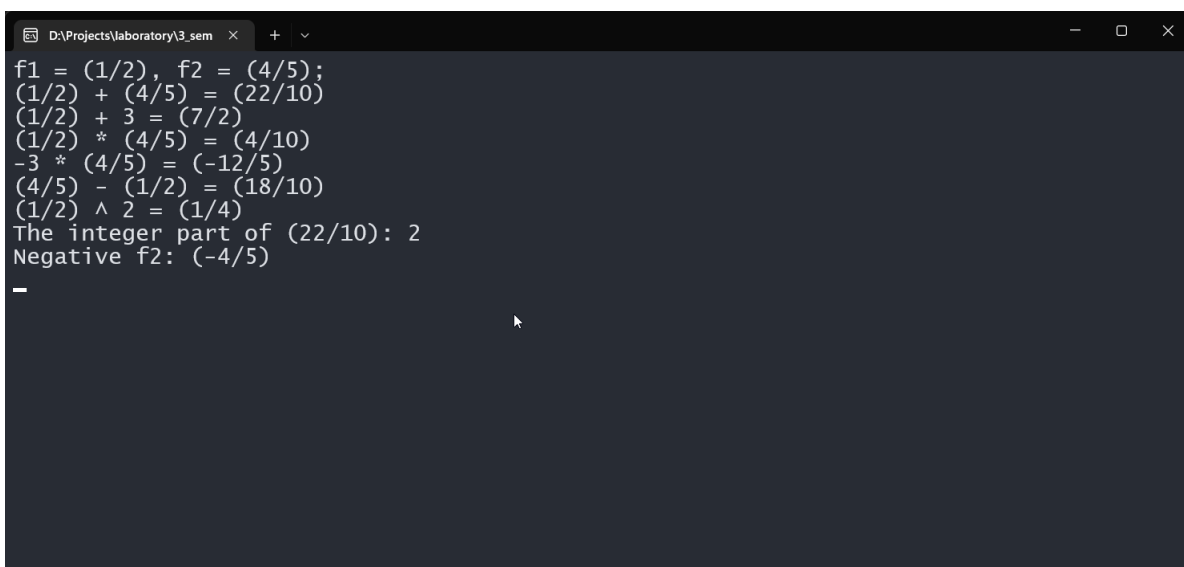
```

```

133         f2.negative();
134         Console.WriteLine($"Negative f2: {f2.view()}");
135
136         Console.ReadKey();
137
138     }
139 }
140 }

```

Результат работы



```

D:\Projects\laboratory\3_sem x + v
f1 = (1/2), f2 = (4/5);
(1/2) + (4/5) = (22/10)
(1/2) + 3 = (7/2)
(1/2) * (4/5) = (4/10)
-3 * (4/5) = (-12/5)
(4/5) - (1/2) = (18/10)
(1/2) ^ 2 = (1/4)
The integer part of (22/10): 2
Negative f2: (-4/5)
-

```