

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
імені ТАРАСА ШЕВЧЕНКА**



ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра прикладних інформаційних систем

Звіт до практичної роботи №7

з курсу

«Об'єктно-орієнтоване програмування»

студента 2 курсу

групи ПП-22

спеціальності 122 «Комп'ютерні науки»

ОП «Прикладне програмування»

Шевлюк Вікторії Віталіївни

Викладач:

к.ф.-м.н., доц. Шолохов О.В.

Київ – 2022

Тема: Використання інтерфейсів у C#.

Мета роботи: Набуття навичок розробки класів з використанням інтерфейсів та базового принципу об'єктно-орієнтованого програмування – успадкування.

Завдання:

Створити клас, що містить методи для знаходження відсотку від числа, збільшення, зменшення числа на певну кількість відсотків (число може бути десятковим або раціональним дробом). При створенні класу використати відповідні інтерфейси. (Варіант 7)

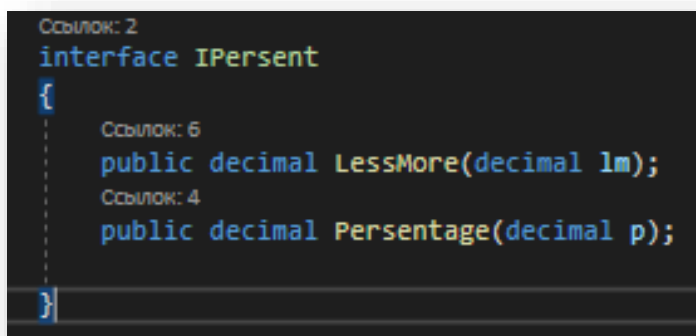
Хід роботи:

Почнемо роботу з того, що створимо інтерфейс, в якому будуть визначені методи для роботи з відсотками.

Інтерфейс - це «полегшений» клас, де всі функції є віртуальними і відсутні поля та конструктори (але можуть бути властивості). Кількість функцій, визначених у конкретному інтерфейсі, залежить від того, яку поведінку ми намагаємося змодельювати за допомогою цього

інтерфейсу. Інтерфейс реалізують класи, і обов'язковою умовою є те, що в кожному класі, що реалізує інтерфейс, повинні бути прописані усі методи, що є в інтерфейсі. Інакше робота з кодом буде неможливою.

Отже, мій інтерфейс:



```
Ссылка: 2
interface IPersent
{
    Ссылка: 6
    public decimal LessMore(decimal lm);
    Ссылка: 4
    public decimal Percentage(decimal p);
}
```

В інтерфейсі я визначила методи для додавання/віднімання до числа певного його відсотку і знаходження відсотку від числа.

Далі створимо класи, у яких реалізуємо відповідні методи.

Клас для раціонального дробу:

```
namespace Lab3_Shevliuk_OOP
{
    Ссылка: 2
    class RationalFraction : IPersent
    {
        Ссылка: 6
        public int numerator { get; set; }
        Ссылка: 6
        public int denominator { get; set; }

        Ссылка: 1
        public RationalFraction(int numerator, int denominator) {...}

        Ссылка: 4
        public decimal Percentage(decimal p)
        {
            decimal lm = ((decimal)numerator) / ((decimal)denominator) * (p / 100);
            return lm;
        }

        Ссылка: 6
        public decimal LessMore(decimal lm)
        {
            return (((decimal)numerator) / ((decimal)denominator) + lm);
        }
    }
}
```

Тепер клас для десяткового дробу:

```
Ссылка: 2
class Decimal : IPersent
{
    Ссылка: 6
    public double number { get; set; }
    Ссылка: 1
    public Decimal(double number)
    {
        this.number = number;
    }
    Ссылка: 4
    public decimal Percentage(decimal p)
    {
        return ((p / 100) * (decimal)number);
    }
    Ссылка: 6
    public decimal LessMore(decimal lm)
    {
        return (decimal)number + lm;
    }
}
```

У ході роботи самої програми користувач зможе ввести число з яким хочу працювати і відсоток, який бажає знайти.

Код основного тіла програми:

```
while (true)
{
    int s = select_menu();
    switch (s)
    {
        case 1:

            Console.WriteLine("Введіть чисельник: ");
            int number = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine();
            Console.WriteLine("Введіть знаменник: ");
            int denom = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine();
            Console.WriteLine("Введіть відсоток: ");
            double p = Convert.ToDouble(Console.ReadLine());
            Console.WriteLine();
            RationalFraction n = new RationalFraction(number, denom);
            decimal res = n.Persentage((decimal)p);
            Console.WriteLine("{0}% від {1}/{2} = {3}", p, n.numerator, n.denominator, res);
            Console.WriteLine();
            decimal more = n.LessMore(res);
            Console.WriteLine("{0}/{1} + {2}%= {3}", n.numerator, n.denominator, p, more);
            Console.WriteLine();
            decimal less = n.LessMore(-res);
            Console.WriteLine("{0}/{1} + {2}%= {3}", n.numerator, n.denominator, p, less);
            Console.WriteLine();
            Console.WriteLine();
            break;

        case 2:

            Console.WriteLine("Введіть число: ");
            double numb = Convert.ToDouble(Console.ReadLine());
            Console.WriteLine();
            Console.WriteLine("Введіть відсоток: ");
            double pc = Convert.ToDouble(Console.ReadLine());
            Console.WriteLine();
            Decimal d = new Decimal(numb);
            decimal d_find_res = d.Persentage((decimal)pc);
            Console.WriteLine("{0}% від {1} = {2}", pc, d.number, d_find_res);
            Console.WriteLine();
            decimal d_plus_res = d.LessMore(d_find_res);
            Console.WriteLine("{0} + {1}%= {2}", d.number, pc, d_plus_res);
            Console.WriteLine();
            decimal d_minus_res = d.LessMore(-d_find_res);
            Console.WriteLine("{0} - {1}%= {2}", d.number, pc, d_minus_res);
            Console.WriteLine();
            Console.WriteLine();
            break;

        case 0:

            Environment.Exit(9);
            break;
    }
}
```

I, зрештою, результат роботи програми:

```
3 яким числом ви хочете працювати?
1 -> Раціональний дріб
2 -> Десятковий дріб
0 -> Завершити програму
1
Введіть чисельник: 4
Введіть знаменник: 6
Введіть відсоток: 33
33% від 4/6 = 0.220000000000000000000000000000
4/6 + 33%= 0.88666666666666666666666666667
4/6 + 33%= 0.44666666666666666666666666667
3 яким числом ви хочете працювати?
1 -> Раціональний дріб
2 -> Десятковий дріб
0 -> Завершити програму
2
Введіть число: 56.56
Введіть відсоток: 76
76% від 56.56 = 42.9856
56.56 + 76%= 99.5456
56.56 - 76%= 13.5744
```

Висновок: у ході цієї лабораторної роботи я набула навичок розробки класів з використанням інтерфейсів та базового принципу об'єктно-орієнтованого програмування – успадкування.

Контрольні питання:

1. Що таке інтерфейс?

Інтерфейс - це «полегшений» клас, де всі функції є віртуальними і відсутні поля (але можуть бути властивості).

2. Що спільного між інтерфейсами та абстрактними класами?

Для них неможливо створити об'єкт, усі абстрактні методи повинні бути замінені підкласом, вони не мають конструктора за замовчуванням,

Дозволяють реалізовувати динамічний поліморфізму.

3. Чим відрізняються інтерфейси від абстрактних класів?

► В інтерфейсі заборонено прописувати реалізацію його членів. В абстрактному класі частина членів може мати реалізацію. Іншими словами, інтерфейс це той самий абстрактний клас, у якого всі методи абстрактні.

► В інтерфейсі заборонено описувати поля (змінні, об'єкти), в абстрактному класі можна.

► Інтерфейс не може мати конструктор. В абстрактному класі може бути оголошений конструктор.

► Будь-який клас може бути успадкований від декількох інтерфейсів. При цьому клас може бути успадкований тільки від одного абстрактного класу (і не більше).

4. Який інтерфейс є базовим для всіх інтерфейсів?

Базовими інтерфейсами інтерфейсу є явні базові інтерфейси та їхні базові інтерфейси. Іншими словами, набір базових інтерфейсів є повним транзитивним закриттям явних базових інтерфейсів, їх явних базових інтерфейсів тощо. Інтерфейс успадковує всі члени своїх базових інтерфейсів.

5. Що повинен містити клас, який реалізує інтерфейс?

Клас, що реалізує інтерфейс, повинен визначити усі методи, описані в цьому інтерфейсі, або має бути абстрактним класом.

6. Чи обов'язково клас, який реалізує інтерфейс має містити реалізації всіх його методів?

Так

7. Чи можна змінній типу інтерфейс надавати значення об'єкта типу класу, в якому реалізовано цей інтерфейс?

Так

8. Чи може інтерфейс мати багато реалізацій?

Так