Основы программирования

Классы. Подробности

Свойства, статические методы, перегрузка операций

Каждый класс -- отдельный файл

Каждый новый класс необходимо сохранять в файле с именем этого класса.

Класс `Vector` должен быть сохранен в отдельном файле (в обозревателе решений выделить «`ConsoleApplication1`» далее или правой кнопкой или в меню «`Проект`» выбрать «`Добавить класс`» или «`Добавить`»+«`Класс`» и новому классу дать нужное имя)

Класс Вектор

Рассмотрим такой объект -- вектор на плоскости.

Чем характеризуется этот объект? Как его можно задать?

Какие операции мы можем выполнять с векторами?

Поля класса Vector

```
//public double x, y;

double x, y;
```

Конструкторы класса Vector

```
// Конструктор с параметрами
public Vector(double x, double y)
{
      this.x = x;
      this.y = y;
}

// Конструктор без параметров
public Vector()
{
}
```

Свойства класса Vector

```
// Свойство доступа к x
public double X
{
      get { return x;}
      set { x = value;}
}

// Свойство доступа к y
public double Y
{
      get { return y;}
      set { y = value;}
}
```

Метод вычисления длины класса Vector

```
// Метод вычисления длины(модуля) вектора
public double Dlina()
{
      return Math.Sqrt(x * x + y * y);
}
```

Свойство класса Vector, имитирующее поле

```
// Свойство - модуль вектора
public double Modul
{
      get
      {
            return Math.Sqrt(x * x + y * y);
      }
}
```

Метод сложения векторов класса Vector

Как сложить вектора?

Наверное, для получения координаты необходимо сложить соответствующие координаты двух векторов.

Например так:

```
x3 =  x1 + x2;
y3 =  y1 + y2;
```

А где нам взять «недостающие» координаты?

Ведь у объекта вектор только две координаты `x` и `y`?

Метод сложения векторов 1

```
// Метод сложения векторов
public void Summa(Vector v)
{
      x = x + v.x;
      y = y + v.y;
}     а this здесь нужно?


//Vector v1 = new Vector(1,1);
//Vector v2 = new Vector(2,2);
//v1.Summa(v2);
```

Метод сложения векторов 2

```
// Метод сложения векторов
public Vector Summa(Vector v)
{
     Vector v0 = new Vector();
     v0.x = x + v.x;
     v0.y = y + v.y;
     return v0;
}

//Vector v1 = new Vector(1,1);
//Vector v2 = new Vector(2,2);
//Vector v3 = v1.Summa(v2);
```

Статический метод сложения векторов 2

```
//Статический метод сложения векторов

static public Vector SummaSt(Vector v1, Vector v2)
{
      /*Vector v0 = new Vector();
      v0.x = v1.x + v2.x;
      v0.y = v1.y + v2.y;
      return v0; */
      // или можно одной строкой:
      return new Vector(v1.x + v2.x, v1.y + v2.y);
}
```

Класс Program

```
// 1 - Создает два экземпляра Vector

Vector v1 = new Vector(1, 1);
Vector v2 = new Vector(2, 1);
```

Класс Program. Обращение к полям

```
// 2 - если поля открыты, то так:
//v1.x = 2;
//Console.WriteLine("Координаты первого вектора
                          ({0}, {1})", v1.x, v1.y);

// 3 - если полям закрыты, то так:
v2.X = 3;
v2.Y = 2;
Console.WriteLine("Координаты второго вектора
                          ({0}, {1})", v2.X, v2.Y);
```

Класс Program. Получение длины вектора

```
// 4 - вычисляет длину вектора
Console.WriteLine("Длина первого вектора:
      {0,7:#0.00}", v1.Dlina());

// 5 - вычисляет длину вектора используя свойство
Console.WriteLine("Длина второго вектора:
      {0,7:#0.00}", v2.Modul);
```

Класс Program. Сложение векторов 1

```
// 6 - проверяем работу метода сложения векторов

v1.Summa(v2);

Console.Write("Координаты нового вектора
      ({0}, {1})", v1.X, v1.Y);
Console.WriteLine(", а его длина равна
      {0,7:#0.00}", v1.Modul);
```

Класс Program. Сложение векторов 2

```
// 7 -  проверяем работу метода сложения векторов,
        возвращающего экземпляр класса Vector

Vector v3 = new Vector();
v3 = v1.Summa(v2);

Console.Write("Координаты нового вектора
        ({0}, {1})", v3.X, v3.Y);
Console.WriteLine(", а его длина равна
        {0,7:#0.00}", v3.Modul);
```

Класс Program. Сложение векторов 3

```
// 8 -  проверяем работу статического метода
        сложения векторов

Vector v3 = new Vector();
v3=Vector.SummaSt(v1, v2);

Console.Write("Координаты нового вектора
        ({0}, {1})", v3.X, v3.Y);
Console.WriteLine(", а его длина равна
        {0,7:#0.00}", v3.Modul);
```

Операции класса Vector

Какие операции можно выполнять с векторами?

- сложения;
- вычитания;
- умножение на число (число * вектор);
- умножение на число (вектор * число);
- умножение векторов.

Операции сложения и вычитания

```
public static Vector operator
                              +(Vector v1, Vector v2)
{
     return new Vector(v1.x + v2.x, v1.y + v2.y);
}

public static Vector operator
                              -(Vector v1, Vector v2)
{
     return new Vector(v1.x - v2.x, v1.y - v2.y);
}
```

Операция умножения

```
public static double operator
                              *(Vector v1, Vector v2)
{
                    return v1.x*v2.x + v1.y*v2.y;
}
```

Проверка операции умножения

```
// 9 - проверяем работу операции умножения
векторов

double v4 = 0;
v4 = v1 * v2;
Console.Write("Произведение векторов {0}", v4);
```

Операция умножения вектора на число

```
public static Vector operator *(Vector v1, int z)
{
                        return new Vector(v1.x*z,
                                          v1.y*z);
}

public static Vector operator *(int z, Vector v2)
{
                        return new Vector(v2.x*z,
                                          v2.y*z);
}
```

Проверка операции умножения на число

```
// 10 -  проверяем работу операции умножения
         векторов

Vector v5 = new Vector();

v5 = v1 * 3;
Console.WriteLine("Координаты нового вектора
         ({0}, {1})", v5.X, v5.Y);

v5 = 3 * v1;
Console.WriteLine("Координаты нового вектора
         ({0}, {1})", v5.X, v5.Y);
```

Задание

Написать класс комплексных чисел. Комплексное число состоит из двух частей операции с которыми производятся независимо.

Например: `x1 = a1 + i*b1; x2 = a2 + i*b2,`

тогда `x = (a1 + a2) + i*(b1 + b2).`

Описать члены класса:

- два поля;
- конструктов без параметров;
- конструктов с параметрами;
- свойства;
- метод сложения комплексных чисел;
- статический метод сложения комплексных чисел;
- операция сложения комплексных чисел.