



Перегрузка методов и операторов

Последнее обновление: 30.10.2015



Иногда возникает потребность создать один и тот же метод, но с разным набором параметров. И в зависимости от имеющихся параметров и использовать определенную версию метода. Допустим, в нашем классе `State` мы хотим определить метод для нападения на другое государство - метод **Attack**. Первая реализация метода будет принимать в качестве параметра объект `State` - то есть государство, на которое мы нападаем:

```
1 Public Sub Attack(enemy As State)
2
3 End Sub
```

Но предположим, что мы хотим определить версию данного метода, где мы будем указывать не только государство, но количество войск, с помощью которых мы нападаем на врага. Тогда мы можем просто добавить вторую версию данного метода:

```
1 Public Class State
2
3     Public Sub Attack(enemy As State)
4         'Здесь код метода
5     End Sub
6
7     Public Sub Attack(enemy As State, army As Integer)
8         'Здесь код метода
9     End Sub
10
11 End Class
```

Либо в определении каждой версии метода мы можем использовать ключевое слово **Overloads** (тогда все версии этого метода должны быть помечены этим словом):

```
1 Public Class State
2
3     Public Overloads Sub Attack(enemy As State)
4         'Здесь код метода
5     End Sub
```

```
6
7     Public Overloads Sub Attack(enemy As State, army As Integer)
8         'Здесь код метода
9     End Sub
10
11 End Class
```

Перегрузка операторов

Кроме методов мы можем также перегружать операторы. Поскольку перегружаемый оператор будет использоваться для всех объектов данного класса, то он объявляется с модификаторами **Public Shared**, после которых указывается ключевое слово **Operator** и далее знак оператора:

```
1 Public Shared Operator +(a As Object, b As Object) As Object
2
3 End Operator
```

Посмотрим на примере. У нас есть класс State или государство. Определим для этого класса оператор "+", который будет обозначать объединение двух государств, кроме того, предусмотрим операторы сравнения - ">" и "<", с помощью которых мы будем сравнивать два государства:

```
1 Public Class State
2
3     Private name_ As String = ""
4
5     Public Property President() As String
6     Public Property Area() As Double
7     Public Property Population() As Integer
8
9     Public Property Name As String
10         Get
11             Return name_
12         End Get
13         Set(value As String)
14             name_ = value
15         End Set
16     End Property
17
18     Public Shared Operator +(a As State, b As State) As State
19
20         Dim c As State = New State()
21         'Площадь нового государства
22         c.Area = a.Area + b.Area
23         'Население нового государства
24         c.Population = a.Population + b.Population
25         c.President = a.President
26         c.Name = a.Name
```

```
27
28     Return c
29 End Operator
30
31 Public Shared Operator >(a As State, b As State) As Boolean
32
33     If a.Area > b.Area Then
34         Return True
35     Else
36         Return False
37     End If
38
39 End Operator
40
41 Public Shared Operator <(a As State, b As State) As Boolean
42
43     If a.Area < b.Area Then
44         Return True
45     Else
46         Return False
47     End If
48
49 End Operator
50
51 End Class
```

Обратите внимание, что поскольку мы применяем операцию сложения и сравнения над двумя объектами State, то мы должны передать эти объекты в качестве параметров. Тогда мы можем использовать операторы следующим образом:

```
1 Sub Main()
2
3     Dim s1 As State = New State()
4     s1.Name = "State1"
5     s1.Area = 1000
6     s1.President = "John"
7     s1.Population = 1000
8
9     Dim s2 As State = New State()
10    s2.Name = "State2"
11    s2.Area = 300
12    s2.President = "Dick"
13    s2.Population = 100
14
15    If s1 > s2 Then
16        Console.WriteLine("Государство {0} больше государства {1}", s1, s2)
17    Else
18        Console.WriteLine("Государство {0} меньше государства {1}", s1, s2)
19    End If
```

```
20
21     Dim s3 As State = s1 + s2
22
23     Console.WriteLine("Название государства : {0}", s3.Name)
24     Console.WriteLine("Президент государства : {0}", s3.President)
25     Console.WriteLine("Площадь государства : {0}", s3.Area)
26     Console.WriteLine("Население государства : {0}", s3.Population)
27
28     Console.ReadLine()
29 End Sub
```

[Назад](#) [Содержание](#) [Вперед](#)



ТАКЖЕ НА METANIT.COM

ListView

2 месяца назад • 1 коммент...

ListView в JavaFX,
создание списков,
получение выбранных в ...

Параметры строки запроса

5 месяцев назад • 1 коммен...

Параметры строки
запроса query string в
приложении Blazor на ...

Взаимодействие с кодом Python

5 месяцев назад • 4 коммен...

Взаимодействие с кодом
Python в программе на
языке Си, установка Qt, ...

Отпра серве

5 меся

Отпра
серве
помо

G

Присоединиться к обсуждению...

войти с помощью

или через DISQUS ?

Имя



Поделиться

Лучшие Новые Старые

M

Михаил

2 года назад edited

— ▾

Мне кажется, нужно ещё в классе State перегрузить оператор implicit string (я раньше си шарп учил, не до конца знаю, как это должно выглядеть здесь), для того чтобы Console.WriteLine("Государство {0} больше государства {1}", s1, s2) выводила не название типа объекта, а данные об объекте.
Думаю я понятно объяснил проблему

1

0

Ответить • Поделиться ›

Подписаться

О защите персональных данных

Помощь сайту

YooMoney:
41001174743222

Qiwi:
qiwi.com/n/METANIT

Перевод на карту
Номер карты:
4048415020898850

[Вконтакте](#) | [Телеграм](#) | [Twitter](#) | [Помощь сайту](#)

Контакты для связи: metanit22@mail.ru

Copyright © metanit.com, 2023. Все права защищены.