

Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний
інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 4 з дисципліни
«Основи програмування 2»

«Перевантаження операторів»

Варіант_№4

Виконав студент ІП-14 Берковський Максим Юрійович
(шифр, прізвище, ім'я, по батькові)

Перевірив _____
(прізвище, ім'я, по батькові)

Київ 2021_

Лабораторна робота №4

Тема: Перевантаження операторів

Мета – вивчити механізми створення класів з використанням перевантажених операторів(операцій).

Хід роботи

Задача.

4. Побудувати клас "Булевий вектор" розмірності n (складається з булевих констант – 0 і 1). Реалізувати для нього декілька конструкторів, геттери, метод знаходження його ваги (кількості компонент, рівних 1). Перевантажити оператори кон'юнкції ("&") та інверсії ("~") компонент булевих векторів. Створити три булеві вектори (B1, B2, B3), використовуючи різні конструктори. Знайти доповнення вектора B1. Визначити вектор B3 як кон'юнкцію зміненого вектора B1 та вектора B2 ($B3 = B1 \wedge B2$). Обчислити вагу отриманого вектора B3.

Постановка задачі.

Для виконання цього завдання спочатку потрібно створити клас BoolVector. Це ми зробимо в окремому файлі. Далі створимо 3 конструктори цього класу для одного параметра, трьох та чотирьох. Потім пропишемо геттери для отримання значень вектора у вигляді масиву, його розміру та ваги. Також створимо методи які будуть виводити в консоль вектор, його вагу та розмір. Далі залишається перегрузити оператори "~" та "&" для нашого класу. Потім прописуємо дії в головній програмі і перевіряємо у відладчику.

Випробування коду на C#:

Код:

```
BoolVector.cs Program.cs
C# oplib4_Csharp oplib4_Csharp.BoolVector

1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace oplib4_Csharp
8 {
9     COMPILE: 12
10    class BoolVector
11    {
12
13        COMPILE: 0
14        public int[] GetBoolVectorArr()
15        {
16            return BoolVectorArr;
17        }
18
19        COMPILE: 2
20        public BoolVector(int n)
21        {
22            BoolVectorArr = new int[n];
23            for (int i = 0; i < n; i++)
24            {
25                BoolVectorArr[i] = GetRandom();
26            }
27
28        COMPILE: 1
29        public BoolVector(int n, int a, int b)
30        {
31            if (a == 0 && b == 1 || a == 1 && b == 0)
32            {
33                BoolVectorArr = new int[n];
34                for (int i = 0; i < n; i++)
35                {
36                    if (i % 2 == 0) BoolVectorArr[i] = a;
37                    else BoolVectorArr[i] = b;
38                }
39            }
40            else Console.WriteLine("Error with initializing Boolean Vector!");
41        }
42
43        COMPILE: 1
44        public BoolVector(int a, int na, int b, int nb)
45        {
46            if (a == 0 && b == 1 || a == 1 && b == 0)
47            {
48                int n = na + nb;
49                BoolVectorArr = new int[n];
50                for (int i = 0; i < n; i++)
51                {
52                    if (i < na) BoolVectorArr[i] = a;
53                    else BoolVectorArr[i] = b;
54                }
55            }
56            else Console.WriteLine("Error with initializing Boolean Vector!");
57        }
58    }
59 }
```

```
BoolVector.cs  Program.cs
C# oplab4_Csharp  oplab4_Csharp.BoolVector

57
58
59 COMMON: 6 public void Print()
60 {
61     Console.WriteLine("[{}]", string.Join(" ", BoolVectorArr));
62 }
63 COMMON: 1 private int GetRandom()
64 {
65     Random rnd = new Random();
66
67     int value = rnd.Next(0, 2);
68
69     return value;
70 }
71
72 COMMON: 4 public int GetLen()
73 {
74     int len = 0;
75     foreach (int v in BoolVectorArr)
76     {
77         len++;
78     }
79     return len;
80 }
81
82 COMMON: 0 public void PrintLen()
83 {
84     Console.WriteLine(GetLen());
85 }
86 COMMON: 1 public int GetWeight()
87 {
88     int sum = 0;
89     foreach(int v in BoolVectorArr)
90     {
91         if (v == 1) sum ++;
92     }
93     return sum;
94 }
95 COMMON: 0 public void PrintWeight()
96 {
97     Console.WriteLine(GetWeight());
98 }
99
100
101 COMMON: 1 public static BoolVector operator ~(BoolVector vector)
102 {
103     for (int i = 0; i < vector.GetLen(); i++)
104     {
105         if (vector.BoolVectorArr[i] == 0) vector.BoolVectorArr[i] = 1;
106         else if (vector.BoolVectorArr[i] == 1) vector.BoolVectorArr[i] = 0;
107     }
108     return vector;
109 }
110
```

```
COMMON: 1 public static BoolVector operator &(BoolVector vector1, BoolVector vector2)
{
    var vector3 = new BoolVector(vector1.GetLen());

    for (int i = 0; i < vector3.GetLen(); i++)
    {
        if(vector1.BoolVectorArr[i] == 1 && vector2.BoolVectorArr[i] == 1) vector3.BoolVectorArr[i] = 1;
        else vector3.BoolVectorArr[i] = 0;
    }

    return vector3;
}
121
122
123
124
125
```

```
BoolVector.cs  Program.cs  x
C# oplib4_Csharp  oplib4_Csharp.Program

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace oplib4_Csharp
8  {
9      class Program
10     {
11         static void Main(string[] args)
12         {
13             var B1 = new BoolVector(7);
14             var B2 = new BoolVector(7, 1, 0);
15             var B3 = new BoolVector(0, 4, 1, 10);
16
17             B1.Print();
18             B2.Print();
19             B3.Print();
20
21             Console.WriteLine("~B1:");
22             B1 = ~B1;
23             B1.Print();
24             Console.WriteLine("B2:");
25             B2.Print();
26             Console.WriteLine("B3 = ~B1 & B2:");
27             B3 = B1 & B2;
28             B3.Print();
29         }
30     }
31 }
```

Результат:

```
Консоль отладки Microsoft Visual Studio

[1, 1, 1, 0, 1, 0, 1]
[1, 0, 1, 0, 1, 0, 1]
[0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
~B1:
[0, 0, 0, 1, 0, 1, 0]
B2:
[1, 0, 1, 0, 1, 0, 1]
B3 = ~B1 & B2:
[0, 0, 0, 0, 0, 0, 0]
```

Висновок:

- ознайомився з технологією перевантаження операторів і навчився розробляти алгоритми та програми з їх застосуванням;
- перевінив роботу програми, випробувавши її у відладчику середовища розробки Visual Studio і отримав задовільний результат.