

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 № 3 з дисципліни
«Основи програмування - 2»
«Перевантаження операторів»

Варіант 4

Виконав: Бондаренко М.В. ПП-13

Перевірила: Вечерковська А.С.

Київ 2022

Лабораторна робота 2

Тема: перевантаження операторів.

Постановка задачі

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.

Код C#:







Program.cs:

```
1      using System;
2
3      namespace lab3
4      {
5          internal class Program
6          {
7              public static void Main(string[] args)
8              {
9
10                 BooleanVector B1 = Funcs.EnterVector();
11                 BooleanVector B2 = new BooleanVector();
12
13                 Funcs.ConvertSize( B_reload: B1, B_default: B2);
14
15                 B1.Output( vector_name: "B1 vector");
16                 B2.Output( vector_name: "B2 vector");
17
18                 B1 = ~B1;
19                 B1.Output( vector_name: "~B1 vector");
20
21                 BooleanVector B3 = B1 & B2;
22                 B3.Output( vector_name: "B3 vector");
23                 Console.WriteLine($"B3 weight is {B3.Weight()}");
24
25             }
26         }
27     }
28
29
```

Funcs.cs:

```
1  using System;
2  using System.Diagnostics.SymbolStore;
3  using System.Linq;
4  using System.Security.Policy;
5
6  namespace lab3
7  {
8      [2 usages]
9      public class Funcs
10     {
11         [1 usage]
12         public static BooleanVector EnterVector()
13         {
14             string vector = "";
15             Console.Write("Enter the size of boolean vector: ");
16             int size = Convert.ToInt32(Console.ReadLine());
17             Random rand = new Random();
18             for (int i = 0; i < size; i++)
19             {
20                 vector += rand.Next(0, 2);
21             }
22
23             BooleanVector B = new BooleanVector(vector);
24
25             return B;
26         }
27
28         [1 usage]
29         public static void ConvertSize(BooleanVector B_reload, BooleanVector B_default)
30         {
31             if (B_reload.GetLength() <= B_default.GetLength())
32             {
33                 B_default.Vector = B_default.Vector.Substring(B_default.GetLength() - B_reload.GetLength());
34             }
35             else
36             {
37                 B_default.Vector = string.Concat(Enumerable.Repeat("0", B_reload.GetLength() - B_default.GetLength())) + B_default.Vector;
38             }
39         }
40     }
```

BooleanVector.cs:

```
1  using System;
2  using System.Collections.Generic;
3  using System.Xml.Schema;
4
5  namespace lab3
6  {
7       18 usages  3 exposing APIs
8      public class BooleanVector
9      {
10         private string vector;
11
12          3 usages
13         public BooleanVector(string vector)
14         {
15             this.vector = vector;
16         }
17
18          1 usage
19         public BooleanVector()
20         {
21             vector = "1010010";
22         }
23
24          8 usages
25         public int GetLength()
26         {
27             return vector.Length;
28         }
29
30          4 usages
31         public string Vector
32         {
33             get => vector;
34             set => vector = value;
35         }
36     }
37 }
```

```

33  public int Weight()
34  {
35      int weight = 0;
36      foreach (var ch:char in vector)
37      {
38          if (ch == '1') weight++;
39      }
40
41      return weight;
42  }
43
44
45  public static BooleanVector operator &(BooleanVector B1, BooleanVector B2)
46  {
47      string conjunction = "";
48      for (int i = 0; i < B1.GetLength(); i++)
49      {
50          if (B1.vector[i] == B2.vector[i] && B1.vector[i] == '1') conjunction += 1;
51          else conjunction += 0;
52      }
53
54      BooleanVector B3 = new BooleanVector(conjunction);
55      return B3;
56  }

```

```

58  public static BooleanVector operator ~(BooleanVector B1)
59  {
60
61      string inversion = "";
62      for (int i = 0; i < B1.GetLength(); i++)
63      {
64          if (B1.vector[i] == '0')
65          {
66              inversion += 1;
67          }
68          else
69          {
70              inversion += 0;
71          }
72      }
73
74      BooleanVector B2 = new BooleanVector(inversion);
75      return B2;
76  }
77
78  public void Output(string vector_name)
79  {
80      Console.WriteLine($"{vector_name}:\t{vector}");
81  }
82
83
84  }
85  }

```

Console:

```
Enter the size of boolean vector: 10
B1 vector:      0101100000
B2 vector:      0001010010
~B1 vector:     1010011111
B3 vector:      0000010010
B3 weight is 2
```