

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.  
Методології програмування»

«Перевантаження операторів»

Варіант 26

Виконав студент ІП-11 Рябов Юрій Ігорович  
(шифр, прізвище, ім'я, по батькові)

Перевірів Вітковська Ірина Іванівна  
(прізвище, ім'я, по батькові)

Київ 2022

## **Лабораторна робота №4**

### **Перевантаження операторів**

#### **Мета**

Вивчити механізми створення класів з використанням перевантаження операторів.

#### **Індивідуальне завдання**

Варіант 26

26. Визначити клас "Час" для роботи із часом в межах доби. Членами класу є години, хвилини та секунди. Реалізувати для нього декілька конструкторів, геттери, метод обчислення часу, що залишився до вказаного моменту часу. Перевантажити оператори: префіксний "++" – для збільшення кількості хвилин на 1, постфіксний "++" - для збільшення кількості секунд на 1; ">" – для порівняння моментів часу. Створити три об'єкта часу (T1, T2, T3), використовуючи різні конструктори. Збільшити час T1 на 1 хвилину, а час T2 – на 1 секунду. З'ясувати, який із цих моментів часу (T1 чи T2) є більшим. Для часу T3 визначити час, що залишився до заданого.

#### **Постановка задачі**

Необхідно створити клас Час, що відповідає вимогам умови, створити 3 його об'єкти випадковим образом та обробити, порівняти третій час із введеним з клавіатури.

Програма на мові C#:

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace PB_lab2._4
8 {
9     0 references
10     class Program
11     {
12         0 references
13         static void Main()
14         {
15             IOManager iOManager = new IOManager();
16             string firstTimeName = "T1";
17             string secondTimeName = "T2";
18             string thirdTimeName = "T3";
19             Random random = new Random();
20             Time firstTime = new Time(random.Next(0, 24));
21             Time secondTime = new Time(random.Next(0, 24), random.Next(0, 60));
22             Time thirdTime = new Time(random.Next(0, 24), random.Next(0, 60), random.Next(0, 60));
23             iOManager.OutputTime(firstTimeName, firstTime);
24             iOManager.OutputTime(secondTimeName, secondTime);
25             iOManager.OutputTime(thirdTimeName, thirdTime);
26             firstTime = +firstTime;
27             secondTime++;
28             Console.WriteLine("After increment:");
29             iOManager.OutputTime(firstTimeName, firstTime);
30             iOManager.OutputTime(secondTimeName, secondTime);
31             iOManager.CompareTimes(firstTime, secondTime, firstTimeName, secondTimeName);
32             Console.WriteLine($"Enter time to see number of seconds from {thirdTimeName} to it");
33             Time timeToCompare = iOManager.GetTime();
34             int difference = thirdTime.TimeToMoment(timeToCompare);
35             Console.WriteLine($"Difference is {difference}");
36         }
37     }
38 }
```

```

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace PB_lab2._4
8  {
9      31 references
10     public class Time
11     {
12         const int SecondsInMinute = 60;
13         const int MinutesInHour = 60;
14         private int seconds;
15         private int minutes;
16         private int hours;
17         9 references
18         public int Seconds
19         {
20             get => seconds;
21             set
22             {
23                 if (value < 0 || value > 59)
24                 {
25                     throw new ArgumentException("Invalid seconds input");
26                 }
27                 else
28                 {
29                     seconds = value;
30                 }
31             }
32         }
33         13 references
34         public int Minutes
35         {
36             get => minutes;
37             set
38             {
39                 if (value < 0 || value > 59)
40                 {
41                     throw new ArgumentException("Invalid minutes input");
42                 }
43                 else
44                 {
45                     minutes = value;
46                 }
47             }
48         }
49     }
50 }

```

```

31 13 references
32 public int Minutes
33 {
34     get => minutes;
35     set
36     {
37         if (value < 0 || value > 59)
38         {
39             throw new ArgumentException("Invalid minutes input");
40         }
41         else
42         {
43             minutes = value;
44         }
45     }
46 13 references
47 public int Hours
48 {
49     get => hours;
50     set
51     {
52         if (value < 0 || value > 23)
53         {
54             throw new ArgumentException("Invalid hours input");
55         }
56         else
57         {
58             hours = value;
59         }
60     }
61 5 references
62 public Time(int hours, int minutes, int seconds)
63 {
64     Hours = hours;
65     Minutes = minutes;
66     Seconds = seconds;
67 }
68 2 references
69 public Time(int hours, int minutes) : this(hours, minutes, 0)
70 { }
71 2 references

```

```

69 2 references
70 public Time(int hours) : this(hours, 0)
71 { }
72 0 references
73 public Time() : this(0)
74 { }
75 1 reference
76 public static Time operator +(Time time)
77 {
78     return new Time(time.Hours, time.Minutes + 1, time.Seconds);
79 }
80 1 reference
81 public static Time operator ++(Time time)
82 {
83     return new Time(time.Hours, time.Minutes, time.Seconds + 1);
84 }
85 1 reference
86 public static bool operator >(Time time1, Time time2)
87 {
88     if (time1.Hours != time2.Hours)
89     {
90         return time1.Hours > time2.Hours;
91     }
92     else if (time1.Minutes != time2.Minutes)
93     {
94         return time1.Minutes > time2.Minutes;
95     }
96     else
97     {
98         return time1.Seconds > time2.Seconds;
99     }
100 }
101 0 references
102 public static bool operator <(Time time1, Time time2)
103 {
104     if (time1.Hours != time2.Hours)
105     {
106         return time1.Hours < time2.Hours;
107     }
108     else if (time1.Minutes != time2.Minutes)
109     {
110         return time1.Minutes < time2.Minutes;
111     }
112     else
113     {
114         return time1.Seconds < time2.Seconds;
115     }
116 }

```

```

95     }
96     0 references
97     public static bool operator <(Time time1, Time time2)
98     {
99         if (time1.Hours != time2.Hours)
100         {
101             return time1.Hours < time2.Hours;
102         }
103         else if (time1.Minutes != time2.Minutes)
104         {
105             return time1.Minutes < time2.Minutes;
106         }
107         else
108         {
109             return time1.Seconds < time2.Seconds;
110         }
111     }
112     /// <summary>
113     /// Calculates number of seconds to a specific moment, which could be negative
114     /// </summary>
115     1 reference
116     public int TimeToMoment(Time moment)
117     {
118         return (moment.Hours - this.hours) * MinutesInHour * SecondsInMinute +
119             (moment.Minutes - this.minutes) * SecondsInMinute +
120             (moment.Seconds - this.seconds);
121     }
122 }
123

```



```

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace PB_lab2._4
8  {
9      2 references
10     public class IOManager
11     {
12         5 references
13         public void OutputTime(string name, Time time)
14         {
15             Console.WriteLine("{0} is {1:d2}:{2:d2}:{3:d2}", name, time.Hours, time.Minutes, time.Seconds);
16         }
17         1 reference
18         public void CompareTimes(Time time1, Time time2, string time1Name, string time2Name)
19         {
20             if (time1 > time2)
21             {
22                 Console.WriteLine($"{time1Name} is bigger");
23             }
24             else
25             {
26                 Console.WriteLine($"{time2Name} is bigger");
27             }
28         }
29
30         1 reference
31         public Time GetTime()
32         {
33             Time time;
34             int hours;
35             int minutes;
36             int seconds;
37             Console.Write("Enter hours:");
38             while (!int.TryParse(Console.ReadLine(), out hours))
39             {
40                 Console.WriteLine("Invalid input, try again");
41             }
42             Console.Write("Enter minutes:");
43             while (!int.TryParse(Console.ReadLine(), out minutes))
44             {
45                 Console.WriteLine("Invalid input, try again");
46             }
47             seconds = 0;
48             while (!int.TryParse(Console.ReadLine(), out seconds))
49             {
50                 Console.WriteLine("Invalid input, try again");
51             }
52             time = new Time(hours, minutes, seconds);
53             return time;
54         }
55     }
56 }

```



```
27 1 reference
28 public Time GetTime()
29 {
30     Time time;
31     int hours;
32     int minutes;
33     int seconds;
34     Console.Write("Enter hours:");
35     while(!int.TryParse(Console.ReadLine(), out hours))
36     {
37         Console.WriteLine("Invalid input, try again");
38     }
39     Console.Write("Enter minutes:");
40     while (!int.TryParse(Console.ReadLine(), out minutes))
41     {
42         Console.WriteLine("Invalid input, try again");
43     }
44     Console.Write("Enter seconds:");
45     while (!int.TryParse(Console.ReadLine(), out seconds))
46     {
47         Console.WriteLine("Invalid input, try again");
48     }
49     time = new Time(hours, minutes, seconds);
50     return time;
51 }
52 }
53 }
```

Виконання коду на мові C#:

```
T1 is 11:00:00
T2 is 00:46:00
T3 is 02:20:00
After increment:
T1 is 11:01:00
T2 is 00:46:01
T1 is bigger
Enter time to see number of seconds from T3 to it
Enter hours:3
Enter minutes:21
Enter seconds:50
Difference is 3710
```

## Висновок

Отже, ми вивчили механізми створення класів з використанням перевантаження операторів на прикладі класу часу.