КПІ ім. Ігоря Сікорського

Кафедра ІПІ

ЗВІТ

про виконання комп’ютерного практикуму № 4

з кредитного модуля

«Основи програмування-2. Методології програмування»

Тема: Перевантаження операторів

Варіант №3

Виконала:

студентка 1-го курсу

гр. ІП-з21 ФІОТ

Гавриленко Даяна Юріївна

Київ 2023

1. Умова завдання:

Изображение выглядит как текст, Шрифт, белый, документ

Автоматически созданное описание

1. Текст програми на мові C#:

Program.cs

using System;  
using System.Collections.Generic;  
using System.Text;  
using System.Linq;  
  
namespace Lab4  
{  
 public class Program  
 {  
 static void Main(string[] args)  
 {  
 ConsoleManager consoleManager = new ConsoleManager();  
   
 Time[] timeArray =  
 {  
 new Time(12, 30, 0),  
 new Time(8, 45),  
 new Time(83)  
 };  
  
 consoleManager.PrintTimeArray(timeArray, "Initial times");  
   
 timeArray[0] += 17;  
 timeArray[1] += 34;  
 timeArray[2] += 72;  
  
 consoleManager.PrintTimeArray(timeArray, "Modified times");  
  
 Console.WriteLine("Time difference between the first and second times");  
 Console.WriteLine($"{timeArray[0] - timeArray[1]}\n");  
  
 Console.WriteLine("Time from time 3 to the end of the day");  
 Console.WriteLine(timeArray[2].GetTimeRemainingUntilEndOfTheDay());  
  
 Time[] userTimes = consoleManager.OutputDataOfTimes(2);  
  
 Console.WriteLine("\nUser inputted times");  
 consoleManager.PrintTimeArray(userTimes, "User times");  
 }  
 }  
}

Time.cs

using System;  
using System.Linq;  
  
namespace Lab4  
{  
 public class Time  
 {  
 private int \_hour;  
 private int \_minute;  
 private int \_second;  
  
 public Time()  
 {  
 Hour = 0;  
 Minute = 0;  
 Second = 0;  
 }  
  
 public Time(int hour)  
 {  
 Hour = hour;  
 Minute = 0;  
 Second = 0;  
 }  
  
 public Time(int hour, int minute) : this(hour)  
 {  
 Minute = minute;  
 Second = 0;  
 }  
  
 public Time(int hour, int minute, int second) : this(hour, minute)  
 {  
 Second = second;  
 }  
  
 public int Hour  
 {  
 get => \_hour;  
 private set => \_hour = value >= 0 ? value % 24 : 0;  
 }  
  
 public int Minute  
 {  
 get => \_minute;  
 private set  
 {  
 if (value >= 0)  
 {  
 Hour += value / 60;  
 \_minute = value % 60;  
 }  
 else  
 {  
 Hour += value / 60 - 1;  
 \_minute = 60 + value % 60;  
 }  
 }  
 }  
  
 public int Second  
 {  
 get => \_second;  
 private set  
 {  
 if (value >= 0)  
 {  
 Minute += value / 60;  
 \_second = value % 60;  
 }  
 else  
 {  
 Minute += value / 60 - 1;  
 \_second = 60 + value % 60;  
 }  
 }  
 }  
  
 public Time GetTimeRemainingUntilEndOfTheDay() => new Time(24 - \_hour, 0 - \_minute, 0 - \_second); // повертає залишковий час до кінця дня  
  
 public override string ToString() // отримання рядкового представлення часу  
 {  
 string[] printable =  
 {  
 \_hour.ToString().PadLeft(2, '0'),  
 \_minute.ToString().PadLeft(2, '0'),  
 \_second.ToString().PadLeft(2, '0')  
 };  
  
 return $"{printable[0]}:{printable[1]}:{printable[2]}";  
 }  
  
 public static Time? Parse(string str) // конвертує числа з рядку в цілі числа  
 {  
 string[] substrings = str.Split(new[] {' ', ',', '.', ':', ';', '/'}, StringSplitOptions.RemoveEmptyEntries);  
 int[] nums = new int[substrings.Length];  
   
 for (int i = 0; i < substrings.Length; i++)  
 {  
 nums[i] = int.Parse(substrings[i]);  
 }  
  
 return nums.Length switch  
 {  
 3 => new Time(nums[0], nums[1], nums[2]),  
 2 => new Time(nums[0], nums[1]),  
 1 => new Time(nums[0]),  
 \_ => null  
 };  
 }  
  
 public static Time operator +(Time time1, Time time2) // додає два об'єкти  
 {  
 return new Time(time1.Hour + time2.Hour, time1.Minute + time2.Minute, time1.Second + time2.Second);  
 }  
  
 public static Time operator +(Time time, int minutes) // додає певну кількість хвилин до об'єкту   
 {  
 return new Time(time.Hour, time.Minute + minutes, time.Second);  
 }  
  
 public static Time operator -(Time time1, Time time2) // віднімає один об'єкт від іншого об'єкту  
 {  
 return new Time(time1.Hour - time2.Hour, time1.Minute - time2.Minute, time1.Second - time2.Second);  
 }  
  
 public static Time operator -(Time time, int minutes) // віднімає певну кількість хвилин від об'єкту  
 {  
 return new Time(time.Hour, time.Minute - minutes, time.Second);  
 }  
 }  
}

ConsoleManager.cs

using System;  
  
namespace Lab4  
{  
 public class ConsoleManager  
 {  
 public void PrintTimeArray(Time[] timeArray, string prePrint = "") //Вивести масив часу  
 {  
 Console.WriteLine(prePrint);  
 foreach (Time time in timeArray)  
 {  
 Console.WriteLine(time);  
 }  
 Console.WriteLine();  
 }  
  
 public Time[] OutputDataOfTimes(int size) //масив часу з даними з консолі  
 {  
 var timeArray = new Time[size];  
 for (int i = 0; i < size; i++)  
 {  
 Console.Write("Enter time(hh:mm:ss): ");  
 string input = Console.ReadLine()!;  
 timeArray[i] = Time.Parse(input)!;  
 }  
  
 return timeArray;  
 }  
 }  
}

1. Відеокопія результатів роботи програми:

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание