Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

**МЕХАНІКО-ТЕХНОЛОГІЧНИЙ ФАКУЛЬТЕТ**

Кафедра програмування та захисту інформації

**Звітц**

з виконаної лабораторної роботи № 4

дисципліни “ Візуальне програмування ”

на тему

**«РОБОТА З РЯДКАМИ»**

Виконав :

студент академічної групи КІ-15

Аннаєв А. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Перевірив :

Викладач

Смiрнова Н.В.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кропивницкий- 2017

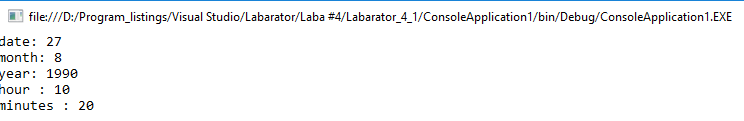
**Лабораторная работа № 4**

**Тема:** Особенности ООП в C # и обработка исключительных ситуаций

**Цель**: Получить практические навыки по созданию и использованию классов в языке C #, научиться реализовывать обработку исключительных ситуаций

**ЗАДАНИЕ**

**Вариант 8** :

Составить описание класса для представления даты. Предусмотреть возможности установления даты и изменения ее отдельных полей (год, месяц, день) с проверкой допустимости вводимых значений. В случае недопустимых значений полей выбрасываются исключения. Создать методы изменения даты на заданное количество дней, месяцев и лет.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication1

{

// структура времени

public struct time{

public int \_hour; // переменная для хранения значение часов

public int \_minutes; // переменная для хранения значение минут

public time(int h, int m) {

this.\_hour = h;

this.\_minutes = m;

}

}

// структура даты

public struct date{

public int \_day; // переменная для хранения значение дня

public int \_month; // переменная для хранения значение месяца

public int \_year; // переменная для хранения значение года

public date(int d, int m, int y) {// конструктор

this.\_day = d;

this.\_month = m;

this.\_year = y;

}

}

// Исключение

public class TimeException : System.Exception

{

public TimeException() { } // конструктор

public override string Message {// переопределение метода

get { return "Enter no correct value for this class"; }

} }

class Program {

public class Time {

time time\_object; // экземпляр структуры времени

date date\_object; // экземпляр структуры даты

//===========================================================================

// конструкторы

public Time(date \_date, time \_time)

{

this.time\_object = CheckCorTime(\_time); // пров. и устан. кор. значения

this.date\_object = CheckCorDate(\_date); // пров. и устан. кор. значения

}

// обратная перегрузка конструктора

public Time(time \_time, date \_date){

this.time\_object = CheckCorTime(\_time); // пров. и устан. кор. значения

this.date\_object = CheckCorDate(\_date); // пров. и устан. кор. значения

}

// перегруженный конструктор

public Time(time \_time) {

this.time\_object = CheckCorTime(\_time); // пров. и устан. кор. значения

}

// перегруженный конструктор

public Time(date \_date){

this.date\_object = CheckCorDate(\_date); // пров. и устан. кор. значения

}

//===========================================================================

// сеттер часов

public void SetTime(time \_time){

this.time\_object = CheckCorTime(\_time); // пров. и устан. кор. значения

}

// сеттер даты

public void SetDate(date \_date) {

this.date\_object = CheckCorDate(\_date); // пров. и устан. кор. значения

}

//===========================================================================

// геттер часов

public time GetTime() {

return time\_object; // возвращает текущее время

}

// геттер даты

public date GetDate() {

return date\_object; // возвращает текущее время

}

//===========================================================================

// отображение времени

public void DisplayTime() {

Console.WriteLine("hour : " + GetTime().\_hour); // вывод часов

Console.WriteLine("minutes : " + GetTime().\_minutes); // вывод минут

}

// отображение даты

public void DisplayDate(){

Console.WriteLine("date: " + GetDate().\_day); // вывод дня

Console.WriteLine("month: " + GetDate().\_month); // вывод месяца

Console.WriteLine("year: " + GetDate().\_year); // вывод года

}

public void Show() {

DisplayDate(); // отображение даты

DisplayTime(); // отображение времени

}

//===========================================================================

// проверка корректности введеного значения

public time CheckCorTime(time t) {

try { // если введено не корректное значение то

if (!(t.\_hour >= 0 && 24 > t.\_hour) ||

!(t.\_minutes >= 0 && 60 > t.\_minutes))

throw new TimeException(); // исключение

}

catch (TimeException e) { Console.WriteLine(e.Message); }

return t;

}

// проверка корректности введеного значения

public date CheckCorDate(date d) {

try { // если введено не корректное значение то

if (!(d.\_day > 0 && 32 > d.\_day) ||

!(d.\_month > 0 && 12 >= d.\_month) ||

!(d.\_year > 1970 && 2017 >= d.\_year))

throw new TimeException();

}

catch (TimeException e) { Console.WriteLine(e.Message); }

return d;

}

}//===========================================================================

static void Main(string[] args)

{

time obj\_1 = new time(21,50);

date obj\_2 = new date(27,08,1990);

time obj\_3 = new time(10, 20);

Time object\_1 = new Time(obj\_1, obj\_2);

object\_1.SetTime(obj\_3);

object\_1.Show();

Console.ReadKey();

}