Министерство образования и науки Украины

Национальный аэрокосмический университет

Им. Н.Е. Жуковского «ХАИ»

Кафедра №302

Отчет по лабораторной работе №1

**По предмету:** «ООП»

**Тема:** «Использование структур»

**Выполнил:**

студент группы 316ст

Волковский В.В.

**Проверил:**

Момот М.А.

Харьков 2014

**Постановка задачи**

Вариант №2

1. Построить программу для работы со структурой Время (часы, минуты, секунды – целый тип).

Программа должна обеспечивать простейшие функции для работы с данными структурами: увеличение/уменьшение на 1 час, минуту, секунду, ввод значений, вывод значений. Функции сделать внешними по отношению к структуре. Составить демонстрационную программу. Для реализации демонстрационной программы использовать отдельный модуль. Посмотреть работу программы в отладчике.

1. Модифицировать программу, используя функции-члены.

**Теоретические сведенья**

Структуры используют большую часть того же синтаксиса, что и классы, однако они более ограничены по сравнению с ними.

* В объявлении структуры поля не могут быть инициализированы до тех пор, пока они будут объявлены как постоянные или статические.
* Структура не может объявлять используемый по умолчанию конструктор (конструктор без параметров) или деструктор.
* Структуры копируются при присваивании. При присваивании структуры к новой переменной выполняется копирование всех данных, а любое изменение новой копии не влияет на данные в исходной копии. Это важно помнить при работе с коллекциями типов значений, такими как Dictionary<string, myStruct>.
* Структуры являются Типами Значений, а классы — Ссылочными Типами.
* В отличие от классов, структуры можно создавать без использования оператора new.
* Структуры могут объявлять конструкторы, имеющие параметры.
* Структура не может быть унаследованной от другой структуры или класса и не может быть основой для других классов. Все структуры наследуют непосредственно от **System.ValueType**, который наследует от **System.Object**.
* Структуры могут реализовывать интерфейсы.
* Структура может использоваться как тип, допускающий значение NULL, и ей можно назначить значение NULL.

**Исходный код**

Class2.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace WindowsFormsApplication1{

class Class2 { public struct Time {

public int hours, minutes, seconds; }

public int getHour(Time t){

return t.hours; }

public int getMinutes(Time t) { return t.minutes; }

public int getSeconds(Time t) { return t.seconds; }

public Time set(int h, int m, int s, Time t) {

if ((h >= 0 && h < 24) && (m >= 0 && m < 60) && (s >= 0 && s < 60)) {t.hours = h; t.minutes = m;t.seconds = s; }else {t.hours = 23; t.minutes = 59;

t.seconds = 59; } return t;}

public void input(Time t){try {

Console.Write("input hours="); int h = Convert.ToInt32(Console.ReadLine());

Console.Write("input minutes=");

int m = Convert.ToInt32(Console.ReadLine());

Console.Write("input seconds="); int s = Convert.ToInt32(Console.ReadLine());

t=set(h, m, s, t); }catch { Console.WriteLine("input error");} }

public void output(Time t) {

Console.WriteLine("Time: " + t.hours + ":" + t.minutes + ":" + t.seconds); }

public Time addHours(Time t){

if (t.hours > 22) t.hours = 0; else t.hours++; return t; }

public Time minHours(Time t) {

if (t.hours < 1) t.hours = 23;

else t.hours--; return t;}

public Time addMinutes(Time t) { if (t.minutes > 58) {

t = addHours(t); t.minutes = 0; }

else t.minutes++; return t; public Time minMinutes(Time t){ if (t.minutes < 1) { t = minHours(t);

t.minutes = 59; }

else t.minutes--; return t; }public Time addSeconds(Time t){

if (t.seconds > 58){t = addMinutes(t);

t.seconds = 0;}else t.seconds++;return t; }public Time minSeconds(Time t){

if (t.seconds < 1) {

t = minMinutes(t); t.seconds = 59;}

else t.seconds--; return t; } public Time initialize(Time t) {

t.hours = 0; t.minutes = 0;t.seconds = 0;

return t; } }}

CodeFile1.cs

using System;

public struct Time

{

public int hours, minutes, seconds;

public int getHour()

{ return hours; }

public int getMinutes() {

return minutes; }

public int getSeconds() {

return seconds; }

public void set(int h, int m, int s)

{ if ((h >= 0 && h < 24) && (m >= 0 && m < 60) && (s >= 0 && s < 60)) { hours = h;

minutes = m;

seconds = s; }

else {

hours = 23;

minutes = 59;

seconds = 59; } }

public void input() {

try {

Console.Write("input ours=");

int h = Convert.ToInt32(Console.ReadLine());

Console.Write("input minutes=");

int m = Convert.ToInt32(Console.ReadLine());

Console.Write("input seconds=");

int s = Convert.ToInt32(Console.ReadLine());

set(h, m, s);

} catch {

Console.WriteLine("input error"); } }

public void output() {

Console.WriteLine("Time: " + hours + ":" + minutes + ":" + seconds); } public void addHours() {

if (hours > 22)

hours = 0; else

hours++; }

public void minHours() {

if (hours < 1) hours = 23; else hours--; } public void addMinutes() {

if (minutes > 58) { addHours(); minutes = 0; } else minutes++; }

public void minMinutes() {

if (minutes < 1) {

minHours(); minutes = 59; }

else minutes--; }

public void addSeconds() {

if (seconds > 58) { addMinutes(); seconds = 0; } else seconds++; }

public void minSeconds() {

if (seconds < 1) {

minMinutes(); seconds = 59; }

else seconds--; }

public void initialize() {

hours = 0; minutes = 0; seconds = 0; }};

Form1.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace WindowsFormsApplication1

{ public partial class Form1 : Form

{ Time time; public Form1() {

InitializeComponent(); }

void refreshTime() {

if(time.getHour() / 10 == 0) {

textBox1.Text = "0" + time.getHour().ToString(); }

else textBox1.Text = time.getHour().ToString();

if(time.getMinutes() / 10 == 0) {

textBox2.Text = "0" + time.getMinutes().ToString(); }

else textBox2.Text = time.getMinutes().ToString();

if (time.getSeconds() / 10 == 0) {

textBox3.Text = "0" + time.getSeconds().ToString(); }

else textBox3.Text = time.getSeconds().ToString(); }

private void button1\_Click(object sender, EventArgs e) {

time.addHours(); refreshTime(); }

private void button2\_Click(object sender, EventArgs e) {

time.addMinutes(); refreshTime(); }

private void button3\_Click(object sender, EventArgs e) {

time.addSeconds(); refreshTime(); }

private void Form1\_Load(object sender, EventArgs e) {

time.initialize(); }

private void button6\_Click(object sender, EventArgs e) {

time.minHours(); refreshTime(); }

private void button5\_Click(object sender, EventArgs e) {

time.minMinutes(); refreshTime(); }

private void button4\_Click(object sender, EventArgs e) {

time.minSeconds(); refreshTime(); }

private void button7\_Click(object sender, EventArgs e) {

if (Convert.ToInt32(textBox1.Text) < 24 && Convert.ToInt32(textBox2.Text) < 60 && Convert.ToInt32(textBox3.Text) < 60)

{ time.set(Convert.ToInt32(textBox1.Text), Convert.ToInt32(textBox2.Text), Convert.ToInt32(textBox3.Text));

refreshTime(); } else {

textBox1.Text = "23";

textBox2.Text = "59";

textBox3.Text = "59";

time.set(23, 59, 59); } MessageBox.Show("Час встановлено: " + textBox1.Text + ":" + textBox2.Text + ":" + textBox3.Text);

} private void button8\_Click(object sender, EventArgs e) {

Form2 form2 = new Form2(); form2.Show(); } }}

Form2.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace WindowsFormsApplication1

{

public partial class Form2 : Form

{

public Form2()

{

InitializeComponent();

}

Class2 MyTime = new Class2();

Class2.Time struTime = new Class2.Time();

void refreshTime()

{

if (MyTime.getHour(struTime) / 10 == 0)

{

textBox1.Text = "0" + MyTime.getHour(struTime).ToString();

}

else textBox1.Text = MyTime.getHour(struTime).ToString();

if (MyTime.getMinutes(struTime) / 10 == 0)

{

textBox2.Text = "0" + MyTime.getMinutes(struTime).ToString();

}

else textBox2.Text = MyTime.getMinutes(struTime).ToString();

if (MyTime.getSeconds(struTime) / 10 == 0)

{

textBox3.Text = "0" + MyTime.getSeconds(struTime).ToString();

}

else textBox3.Text = MyTime.getSeconds(struTime).ToString();

}

private void button7\_Click(object sender, EventArgs e)

{

if (Convert.ToInt32(textBox1.Text) < 24 && Convert.ToInt32(textBox2.Text) < 60 && Convert.ToInt32(textBox3.Text) < 60)

{

struTime = MyTime.set(Convert.ToInt32(textBox1.Text), Convert.ToInt32(textBox2.Text), Convert.ToInt32(textBox3.Text), struTime);

refreshTime();

}

else

{

textBox1.Text = "23";

textBox2.Text = "59";

textBox3.Text = "59";

struTime = MyTime.set(23, 59, 59, struTime);

}

MessageBox.Show("Час встановлено: " + textBox1.Text + ":" + textBox2.Text + ":" + textBox3.Text);

}

private void button1\_Click(object sender, EventArgs e)

{

struTime = MyTime.addHours(struTime);

refreshTime();

}

private void Form2\_Load(object sender, EventArgs e)

{

struTime = MyTime.initialize(struTime);

}

private void button2\_Click(object sender, EventArgs e)

{

struTime = MyTime.addMinutes(struTime);

refreshTime();

}

private void button3\_Click(object sender, EventArgs e)

{

struTime = MyTime.addSeconds(struTime);

refreshTime();

}

private void button6\_Click(object sender, EventArgs e)

{

struTime = MyTime.minHours(struTime);

refreshTime();

}

private void button5\_Click(object sender, EventArgs e)

{

struTime = MyTime.minMinutes(struTime);

refreshTime();

}

private void button4\_Click(object sender, EventArgs e)

{

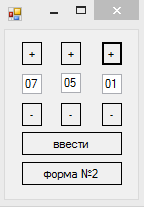
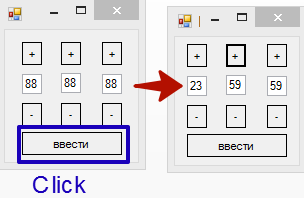
struTime = MyTime.minSeconds(struTime);

refreshTime();

}

}

}

**Скриншоты работы программы**

**Вывод**

На данной лабораторной работе я закрепил теоретические знания, а также приобрел практические навыки по реализации структур в программировании. Также узнал, что структуры являются малозащищенным типом данных. По умолчанию они public и явно нельзя изменить спецификатор доступа, что не есть хорошо.