Міністерство освіти і науки України

Національний аерокосмічний університет

ім. М.Є. Жуковського «ХАІ»

Кафедра №302

Звіт з лабораторної №4

**З предмету:** «ООП»

**Тема:** «Похідні класи в C#»

**Виконав:**

студент групи 316ст

Волківський В.В.

**Перевірив:**

Момот М.А.

Харків 2014

**Постановка завдання**

Варіант №2

Для варіантів завдань, виконаних на попередній лабораторні роботі створити похідний клас, додавши до нього додаткові члени даних і функції члени:

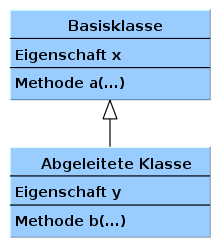
Додати клас розклад. Він вміщує: дисципліну, аудиторію, час початку заняття і функцію обрахування часу до початку заняття.

Реалізувати множинне наслідування за допомогою реалізації інтерфейсу. Змінити демонстраційну програму так щоб вона показувала поліморфну поведінку класів(хоча б одна функція віртуальна).

**Теоретичні відомості**

В [об'єктно-орієнтованому програмуванні](http://uk.wikipedia.org/wiki/%D0%9E%D0%B1%27%D1%94%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D1%96%D1%94%D0%BD%D1%82%D0%BE%D0%B2%D0%B0%D0%BD%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F), **успадкува́ння** — механізм утворення нових [класів](http://uk.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B0%D1%81_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)) на основі використання вже існуючих. При цьому властивості і функціональність батьківського класу переходять до класу нащадка(дочірнього).

## Принцип успадкування

[](http://uk.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB:InheritancePgmUML.svg)

Визначення нового класу (*породжений клас*, *підклас*, [англ.](http://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) *derived class*, [англ.](http://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) *subclass*) може базуватись на визначенні вже існуючого (*базовий клас*, [англ.](http://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) *base class*, [англ.](http://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) *superclass*). В такому випадку, новий клас отримає властивості та поведінку базового класу, та доповнить їх своїми власними. У випадку одиничного успадкування, у кожного класу може бути лише один безпосередній базовий клас. У випадку множинного успадкування, дозволяється існування декількох безпосередніх надкласів.

Застосування механізму успадкування дозволяє покращити повторне використання коду шляхом використання вже визначених властивостей та [методів](http://uk.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)) (поведінки) базових класів.

**Вихідний код**

CodeFile1.cs

using System;

//Класс для работы с временем

public class Time

{

private int hours, minutes, seconds;

//Конструктор по умолчанию

public Time()

{

hours = minutes = seconds = 0;

}

//Конструктор инициализации

public Time(int hours, int minutes, int seconds)

{

if ((hours >= 0 && hours < 24) && (minutes >= 0 && minutes < 60) && (seconds >= 0 && seconds < 60))

{

this.hours = hours;

this.minutes = minutes;

this.seconds = seconds;

}

else

{

this.hours = 23;

this.minutes = 59;

this.seconds = 59;

}

}

//Конструктор копирования

public Time(Time t)

{

this.hours = t.getHours();

this.minutes = t.getMinutes();

this.seconds = t.getSeconds();

}

//Перегрузка оператора +, для сложения двух объектов типа Time

public static Time operator +(Time t1, Time t2)

{

int dH = 0, dM = 0, dS = 0;

if ((t1.getSeconds() + t2.getSeconds()) > 59)

{

dS = t1.getSeconds() + t2.getSeconds() - 60;

dM++;

}

else dS = t1.getSeconds() + t2.getSeconds();

if ((t1.getMinutes() + t2.getMinutes()) > 59)

{

dM += t1.getMinutes() + t2.getMinutes() - 60;

dH++;

}

else dM += t1.getMinutes() + t2.getMinutes();

if ((t1.getHours() + t2.getHours()) > 23)

{

dH += t1.getHours() + t2.getHours() - 24;

}

else dH += t1.getHours() + t2.getHours();

return new Time(dH, dM, dS);

}

//Перегрузка оператора +, для сложения объекта Time с int

public static Time operator +(Time t1, int s)

{

Time tR = new Time();

int dS = 0;

if ((t1.getSeconds() + s) > 59)

{

dS = t1.getSeconds() + s - 60;

tR.set(t1.getHours(), t1.getMinutes(), dS);

tR.addMinutes();

}

else

{

dS = t1.getSeconds() + s;

tR.set(t1.getHours(), t1.getMinutes(), dS);

}

return tR;

}

//Перегрузка оператора +, для сложения объекта int с Time

public static Time operator +(int s, Time t1)

{

Time tR = new Time();

int dS = 0;

if ((t1.getSeconds() + s) > 59)

{

dS = t1.getSeconds() + s - 60;

tR.set(t1.getHours(), t1.getMinutes(), dS);

tR.addMinutes();

}

else

{

dS = t1.getSeconds() + s;

tR.set(t1.getHours(), t1.getMinutes(), dS);

}

return tR;

}

//Перегрузка оператора -, для вычитания двух объектов типа Time

public static Time operator -(Time t1, Time t2)

{

int dH = 0, dM = 0, dS = 0;

if ((t1.getSeconds() - t2.getSeconds()) < 0)

{

dS = (t1.getSeconds() - t2.getSeconds()) + 60;

dM--;

}

else dS = t1.getSeconds() - t2.getSeconds();

if ((t1.getMinutes() - t2.getMinutes()) < 0)

{

dM += (t1.getMinutes() - t2.getMinutes()) + 60;

dH--;

}

else dM += t1.getMinutes() - t2.getMinutes();

if((t1.getHours() - t2.getHours()) < 0)

{

dH += (t1.getHours() - t2.getHours()) + 24;

}

else dH += t1.getHours() - t2.getHours();

return new Time(dH, dM, dS);

}

//Перегрузка оператора -, для вычитания объекта типа Time и int

public static Time operator -(Time t1, int s)

{

Time tR = new Time();

int dS = 0;

if ((t1.getSeconds() - s) < 0)

{

dS = (t1.getSeconds() - s) + 60;

tR.set(t1.getHours(), t1.getMinutes(), dS);

tR.minMinutes();

}

else

{

dS = t1.getSeconds() - s;

tR.set(t1.getHours(), t1.getMinutes(), dS);

}

return tR;

}

//Перегрузка оператора -, для вычитания int и объекта типа Time

public static Time operator -(int s, Time t1)

{

Time tR = new Time();

int dS = 0;

if ((t1.getSeconds() - s) < 0)

{

dS = (t1.getSeconds() - s) + 60;

tR.set(t1.getHours(), t1.getMinutes(), dS);

tR.minMinutes();

}

else

{

dS = t1.getSeconds() - s;

tR.set(t1.getHours(), t1.getMinutes(), dS);

}

return tR;

}

public static Time operator ++(Time t1)

{

t1.addSeconds();

return t1;

}

public static Time operator --(Time t1)

{

t1.minSeconds();

return t1;

}

public static bool operator ==(Time t1, Time t2)

{

if ((t1.getHours() == t2.getHours()) && (t1.getMinutes() == t2.getMinutes()) && (t1.getSeconds() == t2.getSeconds()))

return true;

else return false;

}

public static bool operator !=(Time t1, Time t2)

{

if(!(t1==t2))

return true;

else return false;

}

public static bool operator >(Time t1, Time t2)

{

if (t1.getHours() > t2.getHours()) return true;

else if (t1.getHours() == t2.getHours() && t1.getMinutes() > t2.getMinutes()) return true;

else if (t1.getHours() == t2.getHours() && t1.getMinutes() == t2.getMinutes()

&& t1.getSeconds() > t2.getSeconds()) return true;

else return false;

}

public static bool operator <=(Time t1, Time t2)

{

if (!(t1 > t2)) return true;

else return false;

}

public static bool operator <(Time t1, Time t2)

{

if (t1.getHours() < t2.getHours()) return true;

else if (t1.getHours() == t2.getHours() && t1.getMinutes() < t2.getMinutes()) return true;

else if (t1.getHours() == t2.getHours() && t1.getMinutes() == t2.getMinutes()

&& t1.getSeconds() < t2.getSeconds()) return true;

else return false;

}

public static bool operator >=(Time t1, Time t2)

{

if (!(t1 < t2)) return true;

else return false;

}

public int getHours()

{

return hours;

}

public int getMinutes()

{

return minutes;

}

public int getSeconds()

{

return seconds;

}

public virtual string getAll()

{

return hours.ToString() + ":" + minutes.ToString() + ":" + seconds.ToString();

}

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 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());

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--;

}

}

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;

Time cTime;

Schedule schedule;

public Form1()

{

InitializeComponent();

//time = new Time();

//time = new Time(56,100,200);

time = new Time(22,44,56);

cTime = new Time(time);

refreshTime();

schedule = new Schedule("Кукушка", 814, time.getHours(), time.getMinutes(), time.getSeconds());

}

void refreshTime()

{

if(time.getHours() / 10 == 0)

{

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

}

else textBox1.Text = time.getHours().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 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 (textBox1.Text != "" & textBox2.Text != "" && textBox3.Text != "")

{

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

refreshTime();

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

}

else MessageBox.Show("Заповніть всі поля!");

}

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

{

Form2 form2 = new Form2();

form2.Show();

}

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

{

MessageBox.Show("Copy object hours=" + cTime.getHours() + " minutes=" + cTime.getMinutes()

+ " seconds=" + cTime.getSeconds());

}

private void toolStripMenuItem3\_Click(object sender, EventArgs e)

{

Time t1 = new Time(time);

Time t2 = new Time(10, 10, 10);

time -= t2;

MessageBox.Show("time: " + t1.getHours() + ":" + t1.getMinutes() + ":" + t1.getSeconds()

+ "\nt2: " + t2.getHours() + ":" + t2.getMinutes() + ":" + t2.getSeconds()

+ "\ntime -= t2: " + time.getHours() + ":" + time.getMinutes() + ":" + time.getSeconds());

refreshTime();

}

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

{

Time t1 = new Time(time);

time -= 5;

MessageBox.Show("time: " + t1.getHours() + ":" + t1.getMinutes() + ":" + t1.getSeconds()

+ "\ntime -= 5: " + time.getHours() + ":" + time.getMinutes() + ":" + time.getSeconds());

refreshTime();

}

private void toolStripMenuItem4\_Click(object sender, EventArgs e)

{

Time t1 = new Time(time);

Time t2 = new Time(10, 10, 10);

time += t2;

MessageBox.Show("time: " + t1.getHours() + ":" + t1.getMinutes() + ":" + t1.getSeconds()

+ "\nt2: " + t2.getHours() + ":" + t2.getMinutes() + ":" + t2.getSeconds()

+ "\ntime += t2: " + time.getHours() + ":" + time.getMinutes() + ":" + time.getSeconds());

refreshTime();

}

private void числоToolStripMenuItem1\_Click(object sender, EventArgs e)

{

Time t1 = new Time(time);

time += 5;

MessageBox.Show("time: " + t1.getHours() + ":" + t1.getMinutes() + ":" + t1.getSeconds()

+ "\ntime += 5: " + time.getHours() + ":" + time.getMinutes() + ":" + time.getSeconds());

refreshTime();

}

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

{

time++;

refreshTime();

}

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

{

--time;

refreshTime();

}

private void toolStripMenuItem5\_Click(object sender, EventArgs e)

{

Time t1 = new Time(10, 10, 10);

MessageBox.Show("time: " + time.getHours() + ":" + time.getMinutes() + ":" + time.getSeconds()

+ "\nt1: " + t1.getHours() + ":" + t1.getMinutes() + ":" + t1.getSeconds()

+ "\ntime == t1: " + (time == t1).ToString()

+ "\ntime != t1: " + (time != t1).ToString());

}

private void toolStripMenuItem6\_Click(object sender, EventArgs e)

{

Time t1 = new Time(16, 24, 42);

MessageBox.Show("time: " + time.getHours() + ":" + time.getMinutes() + ":" + time.getSeconds()

+ "\nt1: " + t1.getHours() + ":" + t1.getMinutes() + ":" + t1.getSeconds()

+ "\ntime > t1: " + (time > t1).ToString() + "\ntime < t1: " + (time < t1).ToString()

+ "\ntime >= t1: " + (time >= t1).ToString() + "\ntime <= t1: " + (time <= t1).ToString());

}

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

{

Time t2 = new Time(10, 10, 10);

Time tRP1 = time + t2;

Time tRP2 = time + 5;

Time tRP3 = 10 + time;

Time tRM1 = time - t2;

Time tRM2 = time - 5;

Time tRM3 = 10 - time;

MessageBox.Show("time: " + time.getHours() + ":" + time.getMinutes() + ":" + time.getSeconds()

+ "\nt2: " + t2.getHours() + ":" + t2.getMinutes() + ":" + t2.getSeconds()

+ "\ntime + t2: " + tRP1.getHours() + ":" + tRP1.getMinutes() + ":" + tRP1.getSeconds()

+ "\ntime + 5: " + tRP2.getHours() + ":" + tRP2.getMinutes() + ":" + tRP2.getSeconds()

+ "\n10 + time: " + tRP3.getHours() + ":" + tRP3.getMinutes() + ":" + tRP3.getSeconds()

+ "\ntime - t2: " + tRM1.getHours() + ":" + tRM1.getMinutes() + ":" + tRM1.getSeconds()

+ "\ntime - 5: " + tRM2.getHours() + ":" + tRM2.getMinutes() + ":" + tRM2.getSeconds()

+ "\n10 - time: " + tRM3.getHours() + ":" + tRM3.getMinutes() + ":" + tRM3.getSeconds());

}

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

{

}

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

{

Time t1 = time;

schedule.set(time.getHours(), time.getMinutes(), time.getSeconds());

Time cal = schedule.timeTo();

string output = "Time: " + t1.getAll();

t1 = schedule;

output+="\nSchedule: " + t1.getAll() + "\nЧас до початку: " + cal.getAll();

MessageBox.Show(output);

}

}

}

**Shedule.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace WindowsFormsApplication1

{

public interface inSchedule

{

Time timeTo();

}

public class Schedule : Time, inSchedule

{

string subject;

short classroom;

public Schedule() : base()

{

subject = "";

classroom = 0;

}

public Schedule(string subject, short classroom, int h, int m, int s)

: base(h, m, s)

{

this.subject = subject;

this.classroom = classroom;

}

public Time timeTo()

{

Time now = new Time(DateTime.Now.Hour, DateTime.Now.Minute, DateTime.Now.Second);

Time time = new Time(getHours(), getMinutes(), getSeconds());

return time - now;

}

public override string getAll()

{

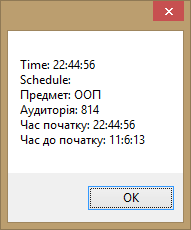
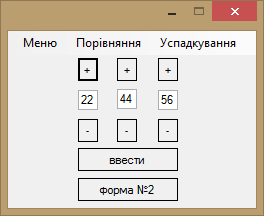
return "\nПредмет: " + subject + "\nАудиторія: " + classroom + "\nЧас початку: " + base.getAll();

}

}

}

**Скріншоти виконання програми**

**Висновки:**

Під час виконання даної лабораторної роботи я набув практичних навичок перевантаження операцій.