

Clock

Furqan Ahmed | Object Oriented Programming | 16B-117-SE

**ABSTRACT**

The idea behind the project is to learn using different classes not taught under the course outline to implement a very basic yet, essential component of everyday life, Time, The entire project revolves around the idea of time, it features multi window GUI based application having, a timer, stopwatch, current time and a time span calculator, which calculates difference between two given dates.

**What Did I Learn?**

Given the basic approach of the project under discussion one might falsely assume that learning something new is next to nothing, but contrary to this, The author came across several new things including but not limited to:

1. Threading in computer systems using C#, To cause a delay of a second, but then the author discovered that it causes slowing of CPU as threading directly affects CPU’s performance.
2. To get an alternative to threading which was compromising the speed of execution, the author came up with using timer class and a custom event handler.
3. Author came up with its own logic to shift digits in all the clock’s compnents.
4. Author came to know proper use of DateTime Class.
5. It also came to know that adding or subtracting Two DateTime objects result in a System.TimeSpan object.
6. How to use Windows Forms, How to trigger events, how to use delegates to call upon multiple functions in an event handler.
7. How to make multiple Windows forms originate from a main windows form
8. How to restrict resize of the windows screen, how to validate data.

**Limitations**

The program in itself, isn’t innovative or doesn’t introduce any new concept.

WORKING

The program features 4 windows forms, First a simple current time telling clock, which just uses DateTime to display current time.

Second, is a timer that takes input from user, and then counts down from that point till the end and then prompts the user, while implementing this section the author faced many hurdles, as this was the first major module after learning Windows forms, including the issue in which the timer will prompt the user but still decrease the time and takes stopwatch to a negative timer.

Third, is a stopwatch, here the window provides 3 buttons: Start, Pause, Reset. Which perform their usual functions. The idea to incorporate the lap recorder was destroyed due to lack of free time with the author. Which is bad!

Fourth, is a TimeSpan Calculator but the catch here is that this option isn’t as obvious as other option clicks. This section uses multiple validations to prevent invalid outputs like different number of months, date etc.

Code:

using System;

using System.Windows.Forms;

namespace Clock

{

public partial class Clock : Form

{

public Clock()

{

InitializeComponent();

}

private void Image\_Timer\_Click(object sender, EventArgs e)

{

Timer\_Form timerObj = new Timer\_Form();

timerObj.Show();

}

private void Image\_StopWatch\_Click(object sender, EventArgs e)

{

StopWatch\_Form stopWatchobj=new StopWatch\_Form();

stopWatchobj.Show();

}

private void Image\_Clock\_Click(object sender, EventArgs e)

{

TimeNow TimeNowObj=new TimeNow();

TimeNowObj.Show();

}

private void Panel\_TimeSpan\_Click(object sender, EventArgs e)

{ TimeSpan TimeSpanObj = new TimeSpan();

TimeSpanObj.Show();

} }}

using System;

using System.Windows.Forms;

namespace Clock

{

public partial class TimeNow : Form

{

public TimeNow()

{

InitializeComponent();

L\_Time.Text = DateTime.Now.ToString();

}

private void btn\_Refresh\_Click(object sender, EventArgs e)

{L\_Time.Text = DateTime.Now.ToString();}}}

using System;

using System.Windows.Forms;

namespace Clock

{

public partial class StopWatch\_Form : Form

{

public StopWatch\_Form()

{

InitializeComponent();

}

private void btn\_Start\_Click(object sender, EventArgs e)

{

timer1.Enabled = true;

}

private void btn\_Reset\_Click(object sender, EventArgs e)

{

timer1.Enabled = false;

L\_MS.Text = "00";

L\_S.Text = "00";

L\_M.Text = "00";

L\_H.Text = "00";

}

private void btn\_Pause\_Click(object sender, EventArgs e)

{

timer1.Enabled = false;

}

private void timer1\_Tick(object sender, EventArgs e)

{

L\_MS.Text = Convert.ToString(Convert.ToInt32(L\_MS.Text) + 01);

if (Convert.ToInt32(L\_MS.Text)==79)

{

L\_MS.Text = "00";

L\_S.Text = Convert.ToString(Convert.ToInt32(L\_S.Text) + 1);

}

if (Convert.ToInt32(L\_S.Text) == 59)

{

L\_S.Text = "00";

L\_M.Text = Convert.ToString(Convert.ToInt32(L\_M.Text) + 1);

}

if (Convert.ToInt32(L\_M.Text) == 59)

{

L\_M.Text = "00";

L\_H.Text = Convert.ToString(Convert.ToInt32(L\_H.Text) + 1); }}}}

using System;

using System.Runtime.InteropServices.WindowsRuntime;

using System.Threading;

using System.Timers;

using System.Windows.Forms;

namespace Clock

{

public partial class Timer\_Form : Form

{

private double i,M,S;

private int \_60SecDelay = 0;

System.Timers.Timer myTimer1 = new System.Timers.Timer();

public Timer\_Form()

{

InitializeComponent();

myTimer1.Elapsed += new ElapsedEventHandler(DigitChanger);

}

/\*COUNTDOWN TIMER STARTS\*/

private void Start\_Timer\_Click(object sender, EventArgs e)

{

if (Convert.ToDouble(txt\_Min.Text) == 0 && Convert.ToDouble(txt\_Sec.Text) == 0)

{

MessageBox.Show("Invalid Duration");

myTimer1.Enabled = false;

}

else

{

myTimer1.Interval = 1000;

myTimer1.Enabled = true;

M = Convert.ToDouble(txt\_Min.Text);

S = Convert.ToDouble(txt\_Sec.Text);

}

}

private void DigitChanger(object source, ElapsedEventArgs e)

{

\_60SecDelay += 1;

txt\_Sec.Text = S.ToString();

txt\_Min.Text = M.ToString();

if (S == 0)

{

S = 0;

}

else

{

S -= 1;

}

if (\_60SecDelay%59==0)

{

if (M == 0)

{

M = 0;

}

else

{

M -= 1;

}

}

if (S==0&&M>0)

{

M -= 1;

S = 59;

}

if (M == 0 && S == 0 && i==0)

{

i += 1;

MessageBox.Show("Time's up!");

}}

/\*COUNTDOWN TIMER END\*/ }}

using System;

using System.Windows.Forms;

namespace Clock

{

static class Program

{

/// <summary>

/// The main entry point for the application.

/// </summary>

[STAThread]

public static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Clock());

}

}

}