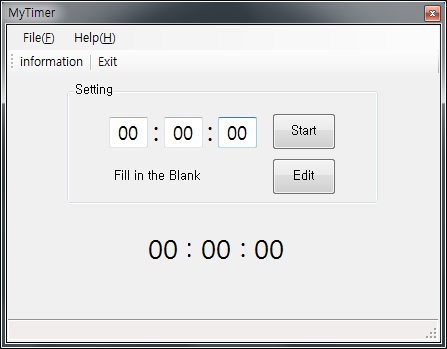


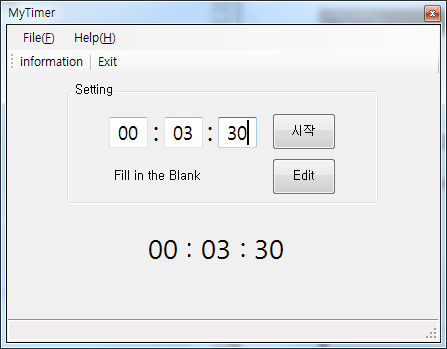
|  |
| --- |
| My Timer  (타이머 프로그램) |
| 2016-2 기말프로젝트 보고서 |



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 제출일 | 2016.12.12 |  | 전공 | 광고홍보학과 |
| 과목 | 윈도우프로그래밍 |  | 학번 | 20132537 |
| 담당교수 | 김동회 |  | 이름 | 김찬미 |

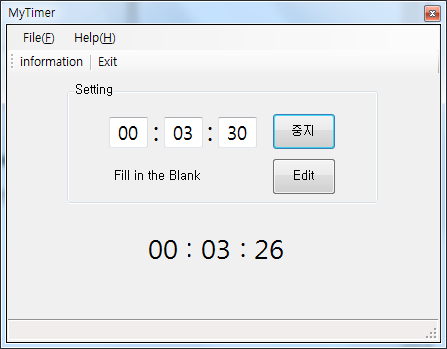
1. **기말프로젝트 보고서**
2. 프로젝트명 : MyTimer
3. 프로젝트 요약 : 사용자가 설정한 시간을 재는 기능이 있는 프로그램
4. **화면구성 및 기능설명**

MyTimer(시작화면) : 프로그램 실행 시 뜨는 화면

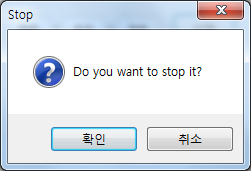


텍스트박스에 사용자가 원하는 값을 입력할 수 있다.

시작버튼을 누르면 프로그램 타이머가 작동된다.

사용자가 설정한 시간이 하단에 출력되고, 시작버튼이 중지버튼으로 바뀐다.

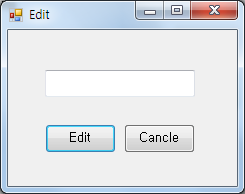
중지버튼을 클릭하면 메시지박스가 출력된다.



Stop(메시지 상자1) : 타이머 중단을 묻는 대화상자

확인을 누를 경우 시간이 멈추게 된다.

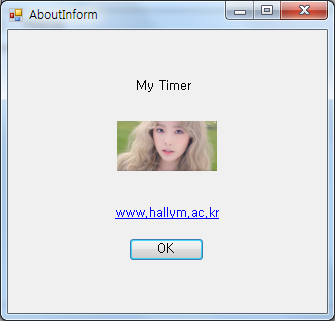
취소를 누르면 대화상자가 닫힌다.



Edit(사용자 정의 대화상자1) : 문구를 수정하는 대화상자

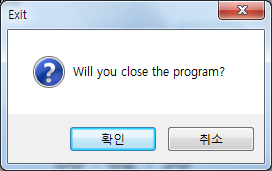
Edit 버튼을 누르면 문구가 바뀐다.

Cancle 버튼을 누르면 대화상자가 닫힌다.

AboutInform(사용자 정의 대화상자2) : 프로그램 설명

상황메뉴에 있는 Information를 눌러도 같은 화면이 출력된다.

OK 버튼을 누르면 대화상자가 닫힌다.



Exit(메시지 상자2) : 타이머 프로그램 종료를 묻는 대화상자

확인을 누를 경우 프로그램이 종료된다.

취소를 누르면 대화상자가 닫힌다.

1. **클래스 및 메소드 설명**

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

: 시, 분, 초에 맞춰 타이머 시간 제어

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

: 클릭 시 타이머 작동 버튼 이벤트

private void txt\_TextChanged(object sender, EventArgs e)

: 타이머 시간에 맞게 시간문구 변경

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

: 프로그램 정보를 표시할 폼 출력(사용자 정의 대화상자2)

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

: 프로그램 종료 메시지 박스를 이용하여 종료 여부 확인 버튼 이벤트(메시지 상자2)

private void MyTimer\_MouseMove(object sender, MouseEventArgs e)

: 마우스 포인터 위치 표시 출력

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

: 텍스트 문구 변경 버튼 이벤트

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

: 사용자가 입력한 문구가 타이머에 출력되게 설정하는 버튼 이벤트

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

: 대화상자를 닫는 버튼 이벤트

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

: 프로그램 정보를 보여주는 대화상자 닫기 버튼 이벤트

private void linkLabel1\_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)

: 하이퍼링크 출력

1. **소스코드(들여쓰기 포함)**

**============== MyTimer.cs(메인)**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace timer

{

public partial class MyTimer : Form

{

// 필요 변수 선언

int hour; // 시간을 나타내는 변수

int minute; // 분을 나타내는 변수

int sec; // 초를 나타내는 변수

public MyTimer()

{

InitializeComponent();

}

// 타이머

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

{

// 초기값 "00:00:00"에 값이 있는지 확인

if(sec > 0 || minute > 0 || hour > 0)

{

if(sec == 0)

{

if(minute == 0)

{

if(hour > 0)

{

// 시를 나타내는 변수값이 있으면

// 분을 60분으로 변경, 시 값 감소

minute = 60;

lblHour.Text = (hour <= 10 ? "0" : "") + --hour;

}

}

if(minute > 0)

{

// 분을 나타내는 변수값이 있으면

// 초를 60초로 변경, 분 값 감소

sec = 60;

lblMinute.Text = (minute <= 10 ? "0" : "") + --minute;

}

}

// 시를 나타내는 변수값이 있으면, 시 값 감소

lblSeconds.Text = (sec <= 10 ? "0" : "") + --sec;

}

else

{

// 타이머 중단

timer1.Stop();

btnStart.Text = "시작";

}

}

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

{

if(!timer1.Enabled) // 타이머가 작동되면

{

// 시, 분, 초에 해당하는 문자를 int값으로 변환

hour = int.Parse(txtHour.Text);

minute = int.Parse(txtMinute.Text);

sec = int.Parse(txtSeconds.Text);

// 시간 출력 시

// "0x:0y:0z"으로 보이도록 설정, x,y,z는 사용자로부터 받는 변수

// 텍스트길이로 판단

lblHour.Text = (txtHour.Text.Length == 1 ? "0" : "") + txtHour.Text;

lblMinute.Text = (txtMinute.Text.Length == 1 ? "0" : "") + txtMinute.Text;

lblSeconds.Text = (txtSeconds.Text.Length == 1 ? "0" : "") + txtSeconds.Text;

timer1.Start(); // 타이머 시작

btnStart.Text = "중지"; // Start 버튼을 중지 버튼으로 바꾸기

}

else

{

// 타이머가 멈추면 버튼 문구가 시작으로 바뀜

timer1.Stop();

btnStart.Text = "시작";

// 중단을 원하는지 묻는 OKCancel 메시지 박스 출력

if (MessageBox.Show("Do you want to stop it?", "Stop", MessageBoxButtons.OKCancel, MessageBoxIcon.Question) == DialogResult.OK)

this.Close(); // OK 클릭시 프로그램 종료

}

}

private void txt\_TextChanged(object sender, EventArgs e)

{

TextBox txt = sender as TextBox; // 시간을 보여줄 텍스트박스 변수

switch(txt.Name)

{

// 타이머가 작동됨에 따라 시, 분, 초의 텍스트 변경

case "txtHour":

lblHour.Text = (txt.Text.Length == 1 ? "0" : "") + txt.Text;

break;

case "txtMinute":

lblMinute.Text = (txt.Text.Length == 1 ? "0" : "") + txt.Text;

break;

case "txtSeconds":

lblSeconds.Text = (txt.Text.Length == 1 ? "0" : "") + txt.Text;

break;

}

}

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

{

// 프로그램 정보를 표시할 폼을 만들어 나타내기

AboutInform af = new AboutInform();

af.Show();

}

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

{

// 프로그램 종료 메시지 박스를 이용하여 종료 여부 다시 확인

// OK를 누르면 프로그램 종료

if (MessageBox.Show("Will you close the program?", "Exit", MessageBoxButtons.OKCancel, MessageBoxIcon.Question) == DialogResult.OK)

Application.Exit();

}

private void MyTimer\_MouseMove(object sender, MouseEventArgs e)

{

// 마우스 포인터 위치 표시

// 상태바에 마우스 포인터 출력

if (statusStrip1.Items.Count > 0)

statusStrip1.Items[0].Text = "마우스 포인터 : " + e.X + "," + e.Y;

}

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

{

// 사용자 정의 대화상자 호출 및 띄우기

Edit ed = new Edit();

ed.Show();

}

}

}

**======= Edit.cs(사용자 정의 대화상자1)**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace timer

{

// 사용자 정의 대화상자

public partial class Edit : Form

{

public Edit()

{

InitializeComponent();

}

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

{

// 사용자가 입력한 문구가 타이머에 출력되게 설정

//lblTextBox.Text = txtBox.Text;

}

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

{

// 취소 버튼을 누를 경우 대화상자 닫기

this.Close();

}

}

}

**======= AboutInform.cs(사용자 정의 대화상자2)**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace timer

{

public partial class AboutInform : Form

{

public AboutInform()

{

InitializeComponent();

linkLabel1.Links.Add(0, 0, "www.hallym.ac.kr"); // 하이퍼링크 출력

}

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

{

// 확인 버튼을 누르면 프로그램 창 닫힘

this.Close();

}

private void linkLabel1\_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)

{

// 하이퍼링크 출력

System.Diagnostics.Process.Start(e.Link.LinkData.ToString());

}

}

}

1. **느낀점**

윈도우프로그래밍 강의를 듣기 전까지는 Windows 상에서 동작하는 프로그램에 대해 잘 몰랐지만, 이번 기말프로젝트를 통해 직접 만들어봄으로써 더 깊이 이해할 수 있었습니다. 특히, 타이머 시, 분, 초에 해당하는 숫자 제어에 대해 방법을 몰라서 오래 걸렸지만, 기억에 남을 프로젝트입니다. 기능적인 면을 추구하여 간소한 디자인을 택했습니다. 더 추가하고 싶은 기능이 있었지만 구현하지 못해 아쉽습니다. 더 공부해보고 버전 업데이트를 해나갈 생각입니다.