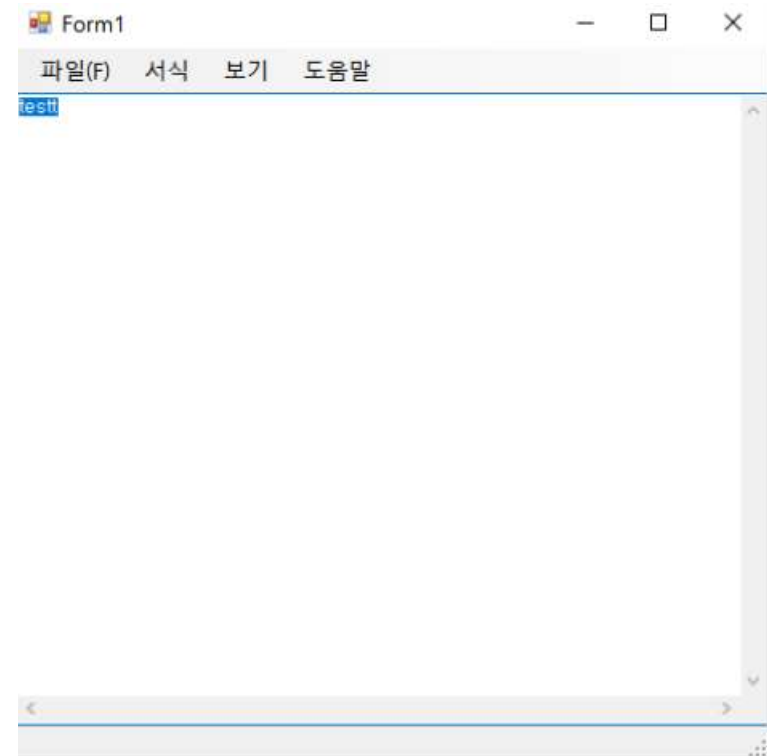


소프트웨어학과 전유림(2020775057)

메모장 업그레이드

원본소스

- MenuStrip
 - 파일(새로만들기, 열기, 저장, 다른이름으로저장, 끝내기)
 - 서식(자동 줄바꿈, 글꼴)
 - 보기(상태포시줄)
 - 도움말(메모장 정보)
- Textbox
- statuStrip



파일(새로만들기, 열기, 끝내기)

```
private void 새로만들기ToolStripMenuItem_Click(object sender, EventArgs e)
{
    :
    tbContents.Text = "";
}
```

```
string filename = "";
```

참조 1개

```
private void 열기ToolStripMenuItem_Click(object sender, EventArgs e)
{
    :
```

```
    //1. 사용자에게 열 파일을 선택 하게 함
    this.openFileDialog1.Filter = "텍스트 문서(+.txt)|+.txt|모든파일|+.*";
    openFileDialog1.ShowDialog();
```

```
    //1-1 선택한 파일명을 변수에 저장
    filename = openFileDialog1.FileName;
```

```
    //2. 파일의 내용을 읽는다.
    string Data = System.IO.File.ReadAllText(filename);
```

```
    //3. 화면에 표시 한다.
    tbContents.Text = Data;
}
```

```
private void 끝내기ToolStripMenuItem_Click(object sender, EventArgs e)
{
    :
    // 프로그램 종료
    this.Close();
}
```

파일(저장, 다른이름으로저장)

```
private void 저장ToolStripMenuItem_Click(object sender, EventArgs e)
{
    //그렇지 않으면 다른이름으로 저장과 동일
    if( filename == "")
    {
        //1. 사용자 에게 저장할 파일을 선택 하게 할
        this.saveFileDialog1.Filter = "텍스트 문서(*.txt)|*.txt|모든파일|*.*";
        this.saveFileDialog1.ShowDialog();

        //2. 파일을 저장 한다.
        System.IO.File.WriteAllText(saveFileDialog1.FileName, tbContents.Text);

        //3. 파일명을 기억
        filename = saveFileDialog1.FileName;
    }
    else
    {
        //해당 파일명으로 현재 내용을 저장

        //2. 파일을 저장 한다.
        System.IO.File.WriteAllText(filename, tbContents.Text);
    }
}
```

```
참조 1개
private void 다른이름으로저장ToolStripMenuItem_Click(object sender, EventArgs e)
{
    //1. 사용자 에게 저장할 파일을 선택 하게 할
    this.saveFileDialog1.Filter = "텍스트 문서(*.txt)|*.txt|모든파일|*.*";
    this.saveFileDialog1.ShowDialog();

    //2. 파일을 저장 한다.
    System.IO.File.WriteAllText(saveFileDialog1.FileName, tbContents.Text);
}

참조 1개
private void 끝내기ToolStripMenuItem_Click(object sender, EventArgs e)
{
    // 프로그램 종료
    this.Close();
}
```

서식

```
private void 자동줄바꿈ToolStripMenuItem_Click(object sender, EventArgs e)
{
    tbContents.WordWrap = !tbContents.WordWrap;

    //if(tbContents.WordWrap == true)
    //{
    //    tbContents.WordWrap = false;
    //}
    //else
    //{
    //    tbContents.WordWrap = true;
    //}
}
```

참조 1개

```
private void 글꼴ToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.fontDialog1.ShowDialog();
    tbContents.Font = fontDialog1.Font;
}
```

도움말

```
private void 메모장정보ToolStripMenuItem_Click(object sender, EventArgs e)
{
    fAbout f = new fAbout();
    f.ShowDialog();
}
```

수정한 내용

- MenuStrip
 - 파일(인쇄)
 - 편집(실행취소, 잘라내기, 찾기등)
 - 서식(배경색, 글자색)

파일(인쇄)

```
private void 인쇄ToolStripMenuItem_Click(object sender, EventArgs e)
{
    PrinterSettings printer = new PrinterSettings();
    PrintDocument pd = new PrintDocument();
    printDialog1.PrinterSettings = printer;
    printDialog1.Document = pd;
    printDialog1.ShowDialog();
}
```


편집(실행취소, 잘라내기, 복사, 붙여넣기, 삭제)

- 실행취소
 - undo
- 잘라내기
 - cut
- 복사
 - copy
- 붙여넣기
 - paste
- 삭제
 - 선택한 값을 empty로 없앴

```
private void 실행취소ToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (this.tbContents.CanUndo)
    {
        this.tbContents.Undo();
    }
}

참조 1개
private void 잘라내기ToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.tbContents.Cut();
}

참조 1개
private void 복사ToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.tbContents.Copy();
}

참조 1개
private void 붙여넣기ToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.tbContents.Paste();
}

참조 1개
private void 삭제ToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.tbContents.SelectedText = String.Empty;
}
```

편집(찾기)

- Forom2 추가

```
private void 찾기FToolStripMenuItem_Click(object sender, EventArgs e)
{
    Form2 find = new Form2();
    find.Show();
}
참조 1개
private void 다음찾기ToolStripMenuItem_Click(object sender, EventArgs e)
{
    찾기FToolStripMenuItem_Click(null, null);
}
```

찾기

찾을내용(N):

☐ 대/소문자 구분(C)

방향

☐ 위로(U) ☐ 아래로(D)

다음 찾기(E)

취소

찾기(Form2)

- button '찾기'를 활성화 시킴
- 찾을 단어가 textBox에 없으면
MessaageBox 출력

```
public partial class Form2 : Form
{
    참조 1개
    public Form2()
    {
        InitializeComponent();
    }
    private Form1 memo = null;

    참조 0개
    public Form2(Form1 objForm1)
    {
        memo = objForm1;
        InitializeComponent();
    }

    참조 1개
    private void Form2_Load(object sender, EventArgs e)
    {
    }

    참조 1개
    private void Find_TextChanged(object sender, EventArgs e)
    {
        this.btnFind.Enabled = true;
    }

    참조 1개
    private void bfind_Click(object sender, EventArgs e)
    {
        if (!FindText())
        {
            MessageBox.Show(this.TextFind.Text + "을(를) 찾을 수 없습니다",
                "메모장", MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
    }
}
```

- 대/소문자구분하기에체크
되어 있지 않으면 모두
소문자로 바꾸어 단어를 찾음
- 위로 찾기
 - (값을 선택할 지점-값을 선택할 크기값)
이 음수이면 더이상 찾을 수 없기 때문에
찾기는 끝남
 - 반대로 양수이면
끝에서부터 검색할 값을 찾기 시작
- 아래로 찾기
 - 위로찾기와는 반대로 시작점부터
검색할 값을 찾음

```
private bool FindText()
{
    int nFind, len;
    string strText, strFind;

    if (uplow.Checked)
    {
        strText = memo.tbContents.Text; //찾을 대상
        strFind = TextFind.Text; //찾을 단어
    }
    else
    {
        strText = memo.tbContents.Text.ToLower();
        strFind = TextFind.Text.ToLower();
    }

    len = TextFind.Text.Length; //텍스트 길이

    //검색
    if (rup.Checked) //위로
    {
        if ((memo.tbContents.SelectionStart - memo.tbContents.SelectionLength) < 0)
            nFind = -1;
        else
            nFind = strText.LastIndexOf(strFind, memo.tbContents.SelectionStart - memo.tbContents.SelectionLength);
    }
    else //아래로
    {
        nFind = strText.IndexOf(strFind, memo.tbContents.SelectionStart + memo.tbContents.SelectionLength);
    }

    if (nFind == -1)
        return false;
    else
    {
        memo.tbContents.SelectionStart = nFind;
        memo.tbContents.SelectionLength = len;
        memo.tbContents.Focus();
        return true;
    }
}

참조 1개
private void Cancel_Click(object sender, EventArgs e)
{
    this.Close();
}
```

편집(모두선택, 시간 날짜)

```
private void 모두선택ToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.tbContents.SelectAll();
}

참조 1개
private void 시간날짜ToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.tbContents.SelectedText=String.Format("{0} {1}", DateTime.Now.ToShortTimeString(), DateTime.Now.ToShortDateString());
}
```

- 모두 선택
 - TextBox에 있는 모든 문자를 선택
- 시간 날짜
 - TextBox에 시간(ToShortTimeString)과 날짜(ToShortDate)를 출력

서식(배경색, 글자색 변경)

```
private void 배경색ToolStripMenuItem1_Click(object sender, EventArgs e)
{
    colorDialog1.ShowDialog();
    tbContents.BackColor = colorDialog1.Color;
}

참조 1개
private void 글ToolStripMenuItem_Click(object sender, EventArgs e)
{
    colorDialog2.ShowDialog();
    tbContents.ForeColor = colorDialog2.Color;
}
```

- 배경색 - BackColor
- 글자색 - ForeColor

- 원본소스

- <https://tindevil.tistory.com/642>

- 참고자료

- 객체지향 자료(13, 14, 15, 16)

- <https://m.blog.naver.com/PostView.naver?isHttpsRedirect=true&blogId=syskiller&logNo=220721534117>

- <https://codingcoding.tistory.com/336>