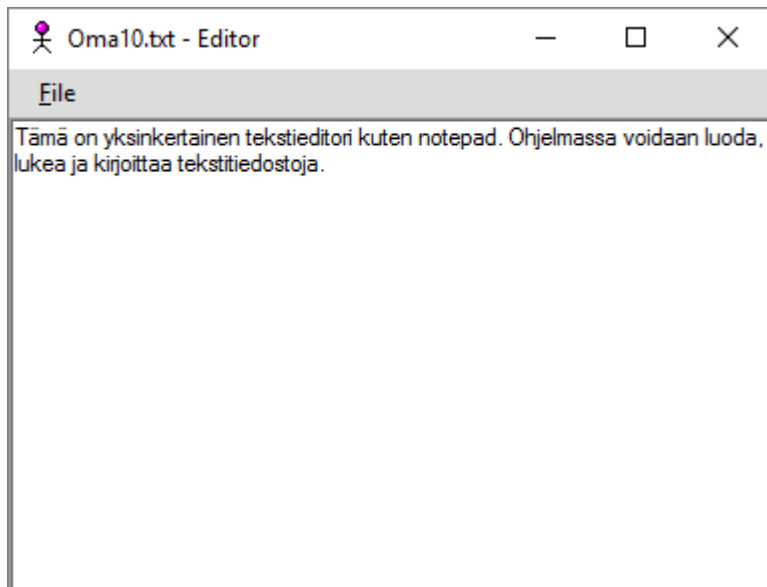


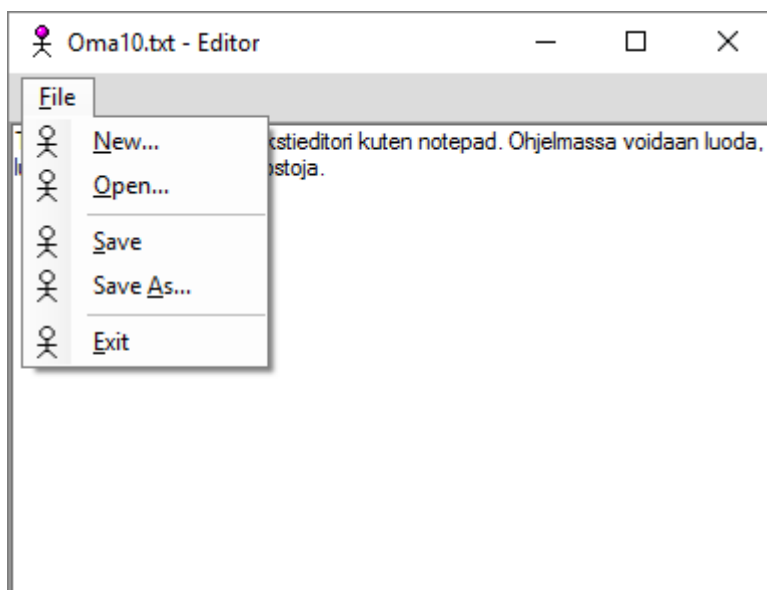
14 File

Tässä tehtävässä tutustutaan RichTextBox-komponenttiin ja tekstitiedoston käsittelyyn (Luonti, talletus ja luku) sekä Windowsin valmiisiin standardeihin avaus/talletus dialogeihin. Valmiin editoriohjelman ikkuna näyttää kuvan 1-1 mukaiselta. Tehtävän osissa on annettu valmiita koodipätkiä (code snippets) joita käyttämällä tiedostokäsittelyn voi toteuttaa.



Kuva 1-1. Pääformi

1. Ota tehtävän pohjaksi perusprojekti, jossa on valmiina menu ja sen *File*-valikossa *Exit*-toiminto.
2. Lisää formiin RichTextBox-komponentti ja sido se formin työalueen reunoihin asettamalla *Dock*-ominaisuus arvoon *Fill*.
3. Luo *File*-valikkoon kuvan 1-2 mukainen valintarakenne ja valinnoille tapahtumankäsittelijät.



Kuva 1-2. Pääformin File-menu.

4. Lisää ohjelman alkuun uusi nimiavaruus, jossa olevia luokkia tarvitaan tiedostojen käsittelyssä:

```
using System.IO;
```

5. Lisää pääformin luokan alkuun luokan sisällä käytettävä globaali muuttuja:

```
private String EditorFileName = "Untitled";
```

6. Lisää luokkaan apumetodi tekstin tulostamista varten formin otsikkopalkkiin.

```
private void SetFormTitleText()
{
    // Tiedoston nimi formiin.
    FileInfo fileinfo = new FileInfo(EditorFileName);
    Text = fileinfo.Name + " - Editor";
}
```

7. Kirjoita ohjelmakoodit uuden dokumentin luontia varten File-valikon New... - valinnan tapahtumankäsittelijään:

```
EditorFileName = "Untitled";
DocumentRichTextBox.Clear();
SetFormTitleText();
```

8. Lisää tiedoston lukemista ja kirjoittamista varten 2 apumetodia ReadFile() ja SaveFile():

```
ReadFile()
try
{
    // Luodaan StreamReader objekti ja luetaan tiedosto.
    using (StreamReader Reader = new StreamReader(EditorFileName))
    {
        // Luku
        DocumentRichTextBox.Clear();
        DocumentRichTextBox.Text = Reader.ReadToEnd();
    }
}
catch (IOException ex)
{
    MessageBox.Show("Error: " + ex.Message, "Open File",
        MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
}
```

```
SaveFile()
try
{
    // Luodaan StreamWriter objekti ja kirjoitetaan teksti
    tiedostoon.
```

```

        using (StreamWriter StrWriter = new StreamWriter(EditorFileName
    ))
    {
        // Kirjoitus
        StrWriter.Write(DocumentRichTextBox.Text);
    }
}
catch (IOException ex)
{
    MessageBox.Show("Error: " + ex.Message, "Save File",
        MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
}

```

9. Kirjoita Open...-valinnan tapahtumankäsittelijään ohjelmakoodit avausdialogin aukaisemista ja dokumentin aukaisemista varten:

```

        // Luodaan standardi avausdialogiobjekti ja alustetaan se.
        OpenFileDialog OpenFileDlg = new OpenFileDialog();
        OpenFileDlg.Title = "Open";
        OpenFileDlg.ShowReadOnly = true;
        OpenFileDlg.Filter = "Text documents
(*.txt)|*.txt|All files|*.*";

        // Avataan windowsin standardi avausdialogi.
        if (OpenFileDlg.ShowDialog() == DialogResult.OK)
        {
            // Talletetaan tiedoston nimi ja polku lukemista
            EditorFileName = OpenFileDlg.FileName;

            // Luetaan tiedoston sisältö ja tuodaan se näytölle.
            ReadFile();

            SetFormTitleText();
        }

```

varten.

10. Kirjoita ohjelmakoodit Save-valinnan tapahtumankäsittelijään:

```

        if (EditorFileName == "Untitled")
        {
            MenuFileSaveAs_Click(sender, e);
        }
        else
        {
            SaveFile();
        }

```

11. Kirjoita koodit Save As...-valinnan tapahtumankäsittelijään:

```

        // Luodaan standardi talletusdialogiobjekti ja alustetaan se.
        SaveFileDialog SaveFileDlg = new SaveFileDialog();

```

```
SaveFileDialog.Filter = "Text documents (*.txt)|*.txt|All files|*.*";

// Avataan windowsin standardi talletusdialogi.
if (SaveFileDialog.ShowDialog() == DialogResult.OK)
{
    // Talletetaan tiedoston nimi ja polku talletusta varten.
    EditorFileName = SaveFileDialog.FileName;

    SaveFile();

    SetFormTitleText();
}
```

12. Testaa lopuksi editorin toimivuus.