```
using System;
using System.IO;
using iText.Kernel.Pdf;
using iText.Layout;
using iText.Layout.Element;
using System.Windows.Forms;

namespace FileToPDF
{
public partial class Welcome : Form
{
private int _counter;
private bool _isCode = true;

public Welcome()
{
InitializeComponent();
}

private void Welcome_Load(object sender, EventArgs e)
{
AllowDrop = true;
}

private void Welcome_DragEnter(object sender, DragEventArgs e)
{
e.Effect = DragDropEffects.Copy;
}

private void Welcome_DragDrop(object sender, DragEventArgs e)
{
string filePath = SetupFilePathFromDragAndDrop(e);
if (!PathIsCorrect(filePath)) return;

string destinationPath = SetupDestinationPath();
if (string.IsNullOrEmpty(destinationPath)) return;

if (FileWasConverted(destinationPath, filePath))
{
_counter++;
Properties.Settings.Default.Counter = _counter;
Properties.Settings.Default.Save();
```

```
DisplayCounter(_counter);
}
else
{
return;
}
}

private string SetupFilePathFromDragAndDrop(DragEventArgs e)
{
if (e.Data.GetDataPresent(DataFormats.FileDrop))
{
return ((string[])e.Data.GetData(DataFormats.FileDrop))[0];
}
else
{
return null;
}
}

private bool PathIsCorrect(string path)
{
DialogResult result = MessageBox.Show($"Is this your correct file?\n\n{path}", "Confirmation",
MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (result == DialogResult.Yes)
{
return true;
}
else
{
return false;
}
}

private string SetupDestinationPath()
{
string rootFolder = Path.GetDirectoryName(Application.ExecutablePath);
string newFolderName = "Exports";
string newFolderPath = Path.Combine(rootFolder, newFolderName);

if (!Directory.Exists(newFolderPath))
```

```
{
Directory.CreateDirectory(newFolderPath);
}

return newFolderPath;
}

private bool FileWasConverted(string destinationpath, string filepath)
{
try
{
string exportFilename = $"{_counter:0000}ConvertedFile.pdf";
string exportPath = Path.Combine(destinationpath, exportFilename);

using (var writer = new PdfWriter(exportPath))
{
using (var pdf = new PdfDocument(writer))
{
using (var document = new Document(pdf))
{
string content = ReadFileContent(filepath);

if (_isCode)
{
var codeElement = new Code(content).SetLanguage("cs");
return true;
}
else
{
document.Add(new Paragraph(content));
return true;
}
}
}
}
}
catch (Exception exception)
{
MessageBox.Show(exception.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
Clipboard.SetText(exception.ToString());
return false;
```

```
}
}

private string ReadFileContent(string filePath)
{
try
{
using (StreamReader reader = new StreamReader(filePath))
{
string fileContent = reader.ReadToEnd();
return fileContent;
}
}
catch (Exception exception)
{
MessageBox.Show($"Error reading file: {exception.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
return null;
}
}

private void DisplayCounter(int counter)
{
CounterLabel.Text = $"{counter:0000}";
}

}
}
```