

Лабораторная работа 6. Организация печати в формах windows

Цель работы

Изучение классов, реализующих задачу программирования печати и получение навыков по работе в программе с диалоговыми окнами.

Упражнение 1. Использование диалоговых окон для печати

При печати различных документов пользователям часто приходится изменять параметры печати. Обычно они ограничиваются заданием таких параметров, как ориентация страницы, ширина полей и размер бумаги.

.NET Framework содержит классы, которые предоставляют пользователям возможность осуществлять и более сложные настройки.

В ходе выполнения данного упражнения будет создано простое приложение, дающее пользователю возможность использовать **PageSetupDialog**, **PrintDialog** и **PrintPreviewDialog** для управления печатью пробного документа.

Добавление компонентов печати

1. Создайте новый проект Windows Forms. Назовите его WinPrint.

2. Из Toolbox перенесите на форму элемент **PrintDocument**.
3. С помощью окна Properties для компонента **printDocument1** добавьте обработчик события *PrintPage* и внутри него добавьте следующий код:

```
Font myFont = new Font("Tahoma", 12, FontStyle.Regular, GraphicsUnit.Pixel);
string Hello = "Hello World!";
e.Graphics.DrawString(Hello, myFont, Brushes.Black, 20, 20);
```
4. Откройте **Form1** в режиме конструктора.
5. Из Toolbox перетащите три элемента **Button** на форму.
6. Укажите последовательно для кнопок значения свойства **Text**:
 - *Page Setup*,
 - *Print*,
 - *Print Preview*.
7. При необходимости увеличьте размеры кнопок.
8. Из Toolbox перетащите в форму компоненты **PageSetupDialog**, **PrintDialog** и **PrintPreviewDialog**.
9. В окне Properties укажите свойству **Document** каждого компонента диалогового окна значение **PrintDocument1**.
10. Для элемента **PrintDialog** присвойте свойству **AllowSomePages** в значение **True**.

Реализация вызова диалоговых окон

11. В конструкторе дважды щелкните кнопку **Page Setup** и добавьте следующий код:

```
pageSetupDialog1.ShowDialog();
```
12. В конструкторе дважды щелкните **Print** и добавьте следующий код:

```
if (printDialog1.ShowDialog() == DialogResult.OK)
    printDocument1.Print();
```
13. В конструкторе дважды щелкните **Print Preview** и добавьте следующий код:

```
printPreviewDialog1.ShowDialog();
```
14. Постройте и запустите приложение. Выберите каждую из кнопок, чтобы проверить открытие различных диалоговых окон печати.

Упражнение 2. Создание документа печати

Выполнив это упражнение, вы дополните решение, разработанное в упражнении 1, и создадите приложение, позволяющее пользователю открывать текстовый файл и печатать его содержимое.

1. Откройте решение, выполненное в упражнении 1.
2. Из Toolbox перетащите элемент - диалоговое окно **OpenFileDialog** на форму.
3. В диалоговом окне Properties присвойте свойству **Filter** элемента **openFileDialog1** значение **Text Files | *.txt**, и очистите поле свойства **FileName**.
4. На форму добавьте новую кнопку с названием **Open File**.

5. Дважды щелкните **Open File**, чтобы открыть редактор кода в обработчике события Click и **выше** обработчика события добавьте код:

```
string s;  
string[] strings;  
int ArrayCounter = 0;
```

6. Внутри этого обработчика события добавьте код:

```
System.Windows.Forms.DialogResult aResult;  
aResult = openFileDialog1.ShowDialog();  
if (aResult == System.Windows.Forms.DialogResult.OK)  
{  
    System.IO.StreamReader aReader =  
        new System.IO.StreamReader(openFileDialog1.FileName);  
    s = aReader.ReadToEnd();  
    aReader.Close();  
    strings = s.Split('\n');
```

7. В обработчике события printDocument1_PrintPage замените существующий код следующим:

```
float LeftMargin = e.MarginBounds.Left;  
float TopMargin = e.MarginBounds.Top;  
float MyLines = 0;  
float YPosition = 0;  
int Counter = 0;  
string CurrentLine;  
MyLines = e.MarginBounds.Height /  
this.Font.GetHeight(e.Graphics);  
while (Counter < MyLines && ArrayCounter <=  
strings.Length - 1)  
{  
    CurrentLine = strings[ArrayCounter];  
    YPosition = TopMargin + Counter *  
this.Font.GetHeight(e.Graphics);  
    e.Graphics.DrawString(CurrentLine, this.Font,  
Brushes.Black, LeftMargin, YPosition, new StringFormat());  
    Counter++;  
    ArrayCounter++;  
}  
if (!(ArrayCounter >= strings.GetLength(0) - 1))  
    e.HasMorePages = true;  
else  
    e.HasMorePages = false;
```

8. Постройте и запустите приложение. Выберите **Open File** и откройте текстовый файл на своем компьютере. Выберите **Print Preview** для просмотра файла в диалоговом окне **Print Preview**. Напечатайте файл (если принтер подключен). Попробуйте с другой страницей и параметрами печати.

Упражнение 3. Создание специализированной формы предварительного просмотра

Хотя компонент **PrintPreviewDialog** является простым, удобным в работе способом предоставить в ваших приложениях функциональность предварительного просмотра, его трудно настроить. Для приложений со специализированным предварительным просмотром можно для создания

специализированного компонента предварительного просмотра использовать элемент управления **PrintPreviewControl**.

Выполнив данное упражнение, вы создадите специализированную форму предварительного просмотра и добавите ее к решению, созданному в упражнении 2. Добавьте к форме **PrintPreviewControl** элементы управления, дающие пользователю возможность указывать масштаб, количество строк и столбцов, а также переключать режим сглаживания.

1. Откройте решение, выполненное в упражнении 2.
2. Добавьте к проекту новую форму.
3. Из Toolbox перетащите **SplitContainer** в форму. В свойстве **Orientation** должна быть задана вертикаль.
4. Из Toolbox перетащите **PrintPreviewControl** в *Panel2* и присвойте свойству **Dock** значение **Fill**.
5. Для **printPreviewControl1** присвойте свойству **Modifiers** значение **Internal**.
6. Из Toolbox добавьте в *Panel1* три элемента управления **Label**, три **NumericUpDown** (надпись **Label** определяет назначение соответствующего элемента **NumericUpDown**), один **Checkbox** и один **Button**. Свяжите надписи с элементами управления **NumericUpDown** и установите свойства, как показано в следующей таблице:

Элемент управления	Свойство	Значение
Label1	Text	Rows
Label2	Text	Columns
Label3	Text	Magnification
NumericUpDown1	Minimum	1
NumericUpDown2	Minimum	1
NumericUpDown3	Minimum	25
NumericUpDown1	Maximum	8
NumericUpDown2	Maximum	8
NumericUpDown3	Maximum	500
NumericUpDown3	Increment	25
CheckBox1	Text	AntiAlias
Button1	Text	Print

7. Дважды щелкните **NumericUpDown1** и добавьте к обработчику события **numericUpDown1_ValueChanged** следующий код:

```
printPreviewControl1.Rows = (int)numericUpDown1.Value;
```

8. В конструкторе дважды щелкните **NumericUpDown2** и добавьте к обработчику события **numericUpDown2_ValueChanged** следующий код:

```
printPreviewControl1.Columns = (int)numericUpDown2.Value;
```

Свойство **Columns** указывает количество отображаемых страниц по горизонтали, а свойство **Rows** – по вертикали.

9. В конструкторе дважды щелкните **NumericUpDown3** и добавьте к обработчику события **numericUpDown3_ValueChanged** следующий код

```
printPreviewControl1.Zoom = (double)numericUpDown3.Value / 100;
```

10. В конструкторе дважды щелкните **CheckBox1** и добавьте к обработчику события `checkBox1_CheckedChanged` следующий код:

```
printPreviewControll1.UseAntiAlias = checkBox1.Checked;
```

11. В конструкторе дважды щелкните **Button1** и добавьте к обработчику события `button1_Click` следующий код

```
this.DialogResult = System.Windows.Forms.DialogResult.OK;
```

12. В редакторе кода формы **Form1** закомментируйте код, существующий в обработчике события *PrintPreviewToolStripMenuItemClick*, и добавьте следующий:

```
Form2 aForm = new Form2();  
System.Windows.Forms.DialogResult aResult;  
aForm.printPreviewControll1.Document = printDocument1;  
aResult = aForm.ShowDialog();  
if (aResult == System.Windows.Forms.DialogResult.OK)  
    printDocument1.Print();
```

13. Постройте и выполните приложение. С помощью команды **Open** меню **File** откройте текстовый файл и затем щелкните **Print Preview** для проверки вашей новой формы предварительного просмотра.

14. Сохраните Ваше приложение и закройте Visual Studio .NET.