

Лабораторна робота 6. Організація друку у формах windows

Вправа 1. Використання діалогових вікон для друку (3 бали)

Вправа 2. Створення документу друку (3 бали)

Вправа 3. Створення спеціалізованої форми попереднього перегляду (3 бали)

Мета роботи: Вивчення класів, які реалізують завдання програмування друку та отримання навичок по роботі в програмі з діалоговими вікнами.

Вправа 1. Використання діалогових вікон для друку (3 бали)

При друкуванні різних документів користувачам часто доводиться змінювати параметри друку. Зазвичай вони обмежуються заданням таких параметрів, як орієнтація сторінки, ширина полів і розмір паперу.

.NET Framework містить класи, які надають користувачам можливість здійснювати і складніші налаштування.

Під час виконання цієї вправи буде створено просте застосування, що дає користувачеві можливість використати PageSetupDialog, PrintDialog і PrintPreviewDialog для управління друком пробного документу.

Додавання компонентів друку

1. Створіть новий проєкт Windows Forms. Назвіть його **WinPrint**.
2. З **Toolbox** перенесіть на форму елемент **PrintDocument**.
3. За допомогою вікна **Properties** для компонента **printDocument1** додайте обробник події **PrintPage** і усередині його додайте наступний 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. Відкрийте **Form1** в режимі конструктора.
5. З **Toolbox** перетягніть три елементи **Button** на форму.
6. Вкажіть послідовно для кнопок значення властивості **Text**:
 - Page Setup,
 - Print,
 - Print Preview.
7. При потребі збільшіть розміри кнопок.
8. З **Toolbox** перетягніть у форму компоненти PageSetupDialog, PrintDialog і PrintPreviewDialog.
9. У вікні **Properties** вкажіть властивості **Document** кожного компонента діалогового вікна значення **PrintDocument1**.
10. Для елемента **PrintDialog** присвойте властивості **AllowSomePages** значення **True**.

Реалізація виклику діалогових вікон

11. У конструкторі двічі клацніть кнопку **Page Setup** і додайте наступний код:

```
pageSetupDialog1.ShowDialog();
```

12. У конструкторі двічі клацніть **Print** і додайте наступний код:

```
if (printDialog1.ShowDialog() == DialogResult.OK)
    printDocument1.Print();
```

13. У конструкторі двічі клацніть **Print Preview** і додайте наступний код:

```
printPreviewDialog1.ShowDialog();
```

14. Побудуйте і запустіть застосування. Виберіть кожну з кнопок, щоб перевірити відкриття різних діалогових вікон друку.

Вправа 2. Створення документа друку (3 бали)

Виконавши цю вправу, ви доповните рішення, розроблене у **Вправі 1**, і створите застосування, яке дозволяє користувачеві відкривати текстовий файл і друкувати його вміст.

1. Відкрийте рішення, виконане у **Вправі 1**.
2. З **Toolbox** перетягніть елемент - діалогове вікно **OpenFileDialog** на форму.
3. У діалоговому вікні **Properties** присвойте властивості **Filter** елемента **openFileDialog1** значення **Text Files | *.txt**, і очистіть поле властивості **FileName**.
4. На форму додайте нову кнопку з назвою **Open File**.
5. Двічі клацніть **Open File**, щоб відкрити редактор коду в обробнику події **Click** і вище обробника події додайте код:

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

6. У середині цього обробника події додайте 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 обробнику події **printDocument1_PrintPage** замініть існуючий код наступним:

```
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. Побудуйте і запустіть додаток. Виберіть **Open File** і відкрийте текстовий файл на своєму комп'ютері. Виберіть **Print Preview** для перегляду файла в діалоговому вікні **Print Preview**. Надрукуйте файл (якщо принтер підключений). Спробуйте з іншою сторінкою і параметрами друк.

Вправа 3. Створення спеціалізованої форми попереднього перегляду (3 бали)

Хоча компонент **PrintPreviewDialog** є простим, зручним у роботі способом надати у ваших застосуваннях функціональність попереднього перегляду, його важко налаштувати. Для додатків зі спеціалізованим попереднім переглядом можна для створення спеціалізованого компонента попереднього перегляду використати елемент управління **PrintPreviewControl**.

Виконавши цю вправу, ви створите спеціалізовану форму попереднього перегляду і додасте її до рішення, створеного у **вправі 2**. Додасте до форми **PrintPreviewControl** елементи управління, які дають користувачеві можливість вказувати масштаб, число рядків і стовпців, а також перемикає режим згладжування.

1. Відкрийте рішення, виконане у **Вправі 2**.
2. Додайте до проекту нову форму.
3. З **Toolbox** перетягніть **SplitContainer** у форму. У властивості **Orientation** має бути задана вертикаль.
4. З **Toolbox** перетягніть **PrintPreviewControl** в **Panel2** і присвойте властивості **Dock** значення **Fill**.
5. Для **printPreviewControl1** присвойте властивості **Modifiers** значення **Internal**.
6. З **Toolbox** додайте в **Panel1** три елементи управління **Label**, три **NumericUpDown** (напис **Label** визначає призначення відповідного елемента **NumericUpDown**), один **Checkbox** і один **Button**. Зв'яжіть написи з елементами управління **NumericUpDown** і встановіть властивості, як показано в наступній таблиці:

Елемент управління	Властивість	Значення
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. Двічі клацніть **NumericUpDown1** і додайте до обробника події **numericUpDown1_ValueChanged** наступний код:

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

8. У конструкторі двічі клацніть **NumericUpDown2** і додайте до обробникові події **numericUpDown2_ValueChanged** наступний код:

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

Властивість **Columns** вказує число сторінок, які відображаються, по горизонталі, а властивість **Rows** - по вертикалі.

9. У конструкторі двічі клацніть **NumericUpDown3** і додайте до обробника події **numericUpDown3_ValueChanged** наступний код

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

10. У конструкторі двічі клацніть **CheckBox1** і додайте до обробника події **checkBox1_CheckedChanged** наступний код:

```
printPreviewControl1.UseAntiAlias = checkBox1.Checked;
```

11. У конструкторі двічі клацніть **Button1** і додайте до обробника події **button1_Click** наступний код

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

12. У редакторі коду форми **Form1** закоментуйте код, який є в обробнику події **button3_Click**, і додайте наступний:

```
Form2 aForm = new Form2();
System.Windows.Forms.DialogResult aResult;
aForm.printPreviewControl1.Document = printDocument1;
aResult = aForm.ShowDialog();
if (aResult == System.Windows.Forms.DialogResult.OK)
    printDocument1.Print();
```

13. Побудуйте і виконайте застосування. За допомогою команди **Open** меню **File** відкрийте текстовий файл і потім клацніть **Print Preview** для перевірки вашої нової форми попереднього перегляду.

14. Збережіть застосування і закрийте Visual Studio .NET.