Діалоги.
Стандартні діалоги

Клакович Л.М.

Діалог – це форма із специфічними властивостями

class MyDialog: Form {…}

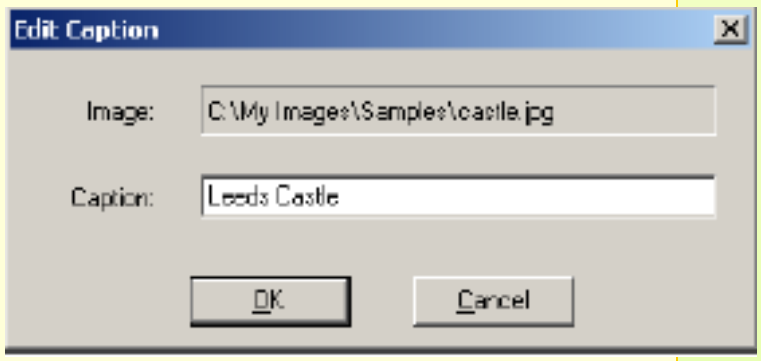
Вікно діалогу:

- не містить меню
- не змінює розмір:

```
dlg.FormBorderStyle=FormBorderStyle.FixedDialog;
```

- не містить кнопок мінімізації, максимізації:

```
dlg.MaximazeBox=False;
dlg.MinimazeBox=False;
```



- відсутність в TaskBar:

```
dlg.ShowInTaskBar=False;
```

- перше відображення

```
dlg.StartPosition=CenterParent;
```

Активізація діалогу

- **Модальні діалоги:**

  `DialogResult` **`ShowDialog`**`()`

  `DialogResult` **`ShowDialog`**`(IWin32Window` *`owner`*`)`

  Код, що слідує за цим методом буде виконуватись лише після закриття модального діалогу

```
dlg.ShowDialog();
if (dlg.DialogResult == DialogResult.OK) {
        MessageBox.Show("Натиснено OK ");
        dlg.Dispose();
}
```

- **Немодальні діалоги:**

  void **Show** ()

  void **Show** (IWin32Window *owner*)

Код, що слідує за цим методом буде виконуватись негайно, не очікуючи закриття діалогу

Конструювання модального діалогу:

- Встановлення необхідних властивостей форми-діалогу.
- Розміщення кнопок OK, Cancel і ін. для фіксування результату діалогу, його закриття і повернення до батьківської форми.
- Встановлення властивості **DialogResult** цих кнопок в одне із значень з перелічення **DialogResult.**

public enum DialogResult ;

OK, Abort , Cancel, Ignore, No, Yes, Retry,

None – модальний діалог продовжує працювати

- Після кліку на цих кнопках діалог закривається, а властивості **DialogResult** діалогу присвоюється значення однойменної властивості кнопки.

  • Визначення кнопок, які будуть активізовані при натисканні ENTER та ESC клавіш на клавіатурі (властивості AcceptButton та CancelButton)

```
MyDialog dlg = new MyDialog();
Button button1 = new Button ();
button1.Text = "OK";
button1.Location = new Point (10, 10);
button1.DialogResult = DialogResult.OK;
dlg.AcceptButton = button1;
dlg.Controls.Add(button1);
```

Отримання даних з модального діалогу

```
class MyDialog {

…
        private TextBox textBox1;
        public string Mes
            {
                        get { return textBox1.Text; }
                        set { textBox1.Text = value;}
            }
 }
```

butOK.DialogResult = DialogResult.OK;
butCan.DialogResult = DialogResult. Cancel;

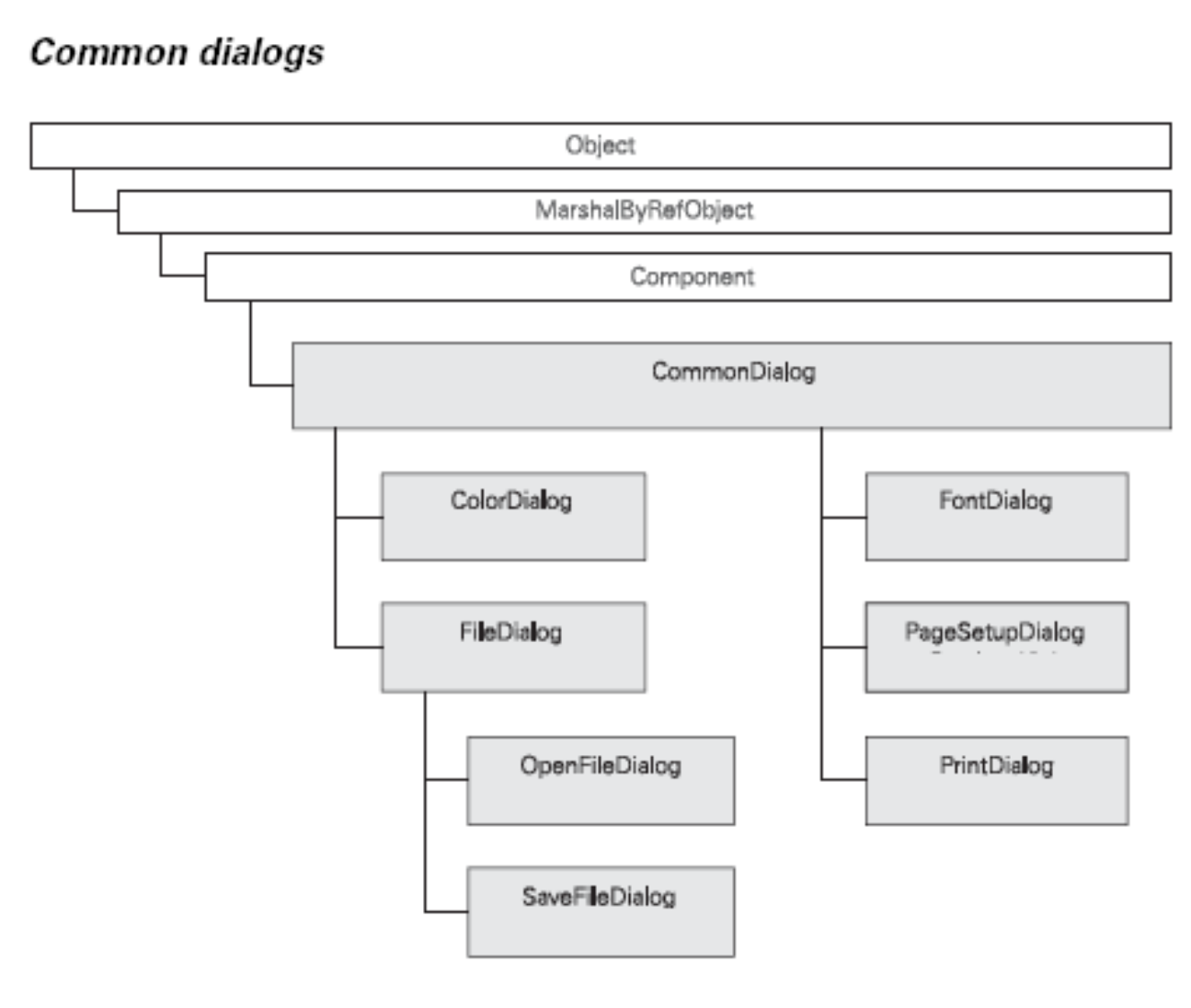

```
class MainForm{     …
        private void OnClick(object sender, EventArgs )
        {
            MyDialog dlg = new MyDialog();
            dlg.ShowDialog();
            if (dlg.DialogResult == DialogResult.OK) {
                    MessageBox.Show( dlg.Mes, “Натиснено OK ",);
              }
             dlg.Dispose();

        }}
```

Отримання даних з немодального діалогу

```
class MyModalessDialog { …
          private TextBox textBox1;
          public string Mes
          {
              get { return textBox1.Text; }
              set { textBox1.Text = value;}
          }
}
```

```
class MainForm {

 …
      private MyModalessDialog dlg=null;
      private void OnClick(object sender, EventArgs e){
                    dlg = new MyModalessDialog();
                    dlg.Owner = this;
                    dlg.Show();
        }
    …
         MessageBox.Show(dlg.Mes);
}
```

Стандартні діалоги



`FileDialog` – абстрактний клас

Public Properties:

AddExtension - чи діалог автоматично додає розширення для файлів

CheckFileExists – чи діалог показує зауваження, що файл не існує

FileName – рядок – ім”я вибраного файлу

FileNames – повертає масив рядків – імена вибраних файлів (якщо властивість `OpenFileDialog.Multiselect=true`)

Filter - рядок – фільтр для вибору типу файлів

InitialDirectory – каталог, що початково вибирається

RestoreDirectory – чи діалог відновлює поточний каталог до оригінальної версії перед закриттям

ShowHelp – чи буде help кнопка на діалозі

Title – рядок назви діалогу

Public Methods:

Reset – відновлення всіх властивостей діалогу до значень за замовчуванням

ShowDialog

Public Events:

FileOK – настає, коли Open або Save кнопки натиснені

HelpRequested – настає, коли кнопка Help натиснена

OpenFileDialog

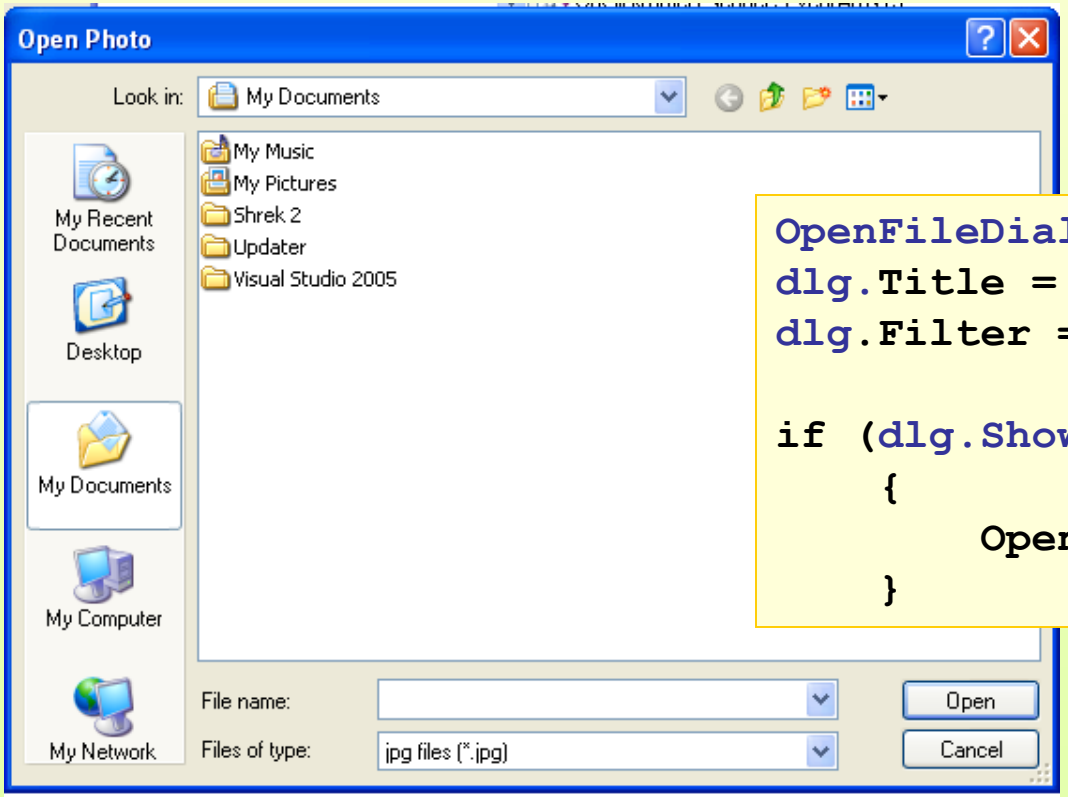
Public Properties:

Multiselect – get or set значення про можливість вибору декількох файлів.

ReadOnlyChecked - чи значок read-only вибрано.

Public Methods:

OpenFile – відкриває файл read/write вибраний користувачем. Повертає об”єкт stream




```
OpenFileDialog dlg = new OpenFileDialog();
dlg.Title = "Open Photo";
dlg.Filter = "jpg files (*.jpg)|*.jpg|
                All files (*.*)|*.*" ;
if (dlg.ShowDialog() == DialogResult.OK)
    {
        Open(dlg.FileName);
    }
```

SaveFileDialog

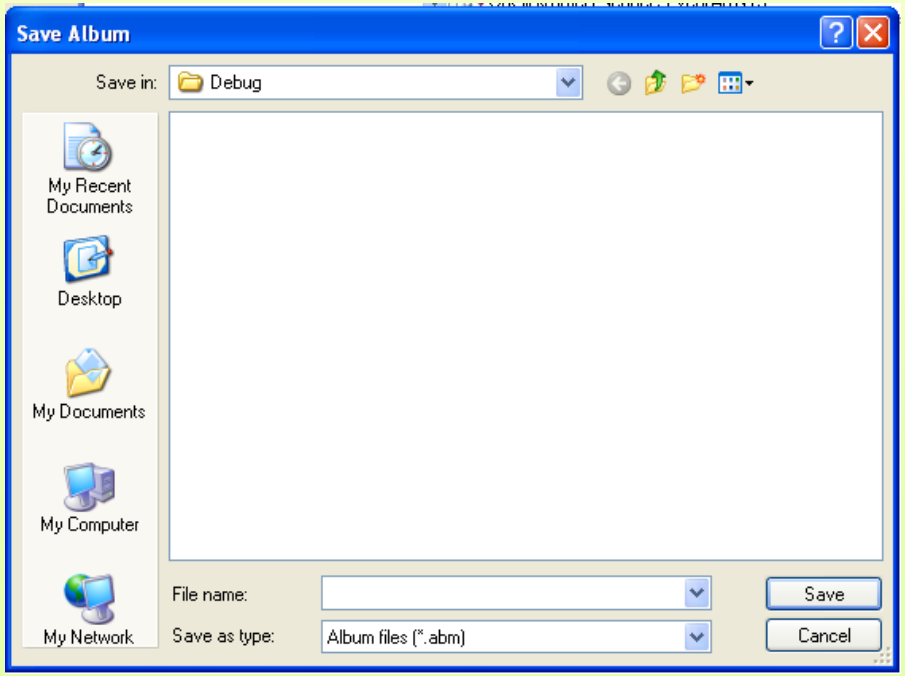
Public Properties:

CreatePrompt – повідомляти про необхідність створення файлу, якщо такий не існує (за замовчуванням –false)

OverwritePrompt – повідомляти, що файл буде перезаписано, якщо такий існує (за замовчуванням – true)

Public Methods:

OpenFile – повертає об”єкт stream з read/write властивістю для вибраного користувачем файлу



```
SaveFileDialog dlg = new SaveFileDialog();
dlg.Title = "Save Album";
dlg.DefaultExt = "abm";
dlg.Filter = "Album files (*.abm)|*.abm|"
             + "All files|*.*";
dlg.InitialDirectory=System.IO.Directory.GetCurrentDirectory();
dlg.RestoreDirectory = true;
if (dlg.ShowDialog() == DialogResult.OK)
  {
     Save(dlg.FileName);
  }
dlg.Dispose();
```

```
public void Save(string fileName)
{
       FileStream fs = new FileStream(fileName, FileMode.Create,
                       FileAccess.ReadWrite);
       StreamWriter sw = new StreamWriter(fs);
       sw.WriteLine(……);
       sw.Close();
       fs.Close();
}
```

ColorDialog

Public Properties:

AllowFullopen – чи буде користувач визначати колір самостійно (false – заборона, відключається кнопка Define Custom Colors). За замовчуванням true.

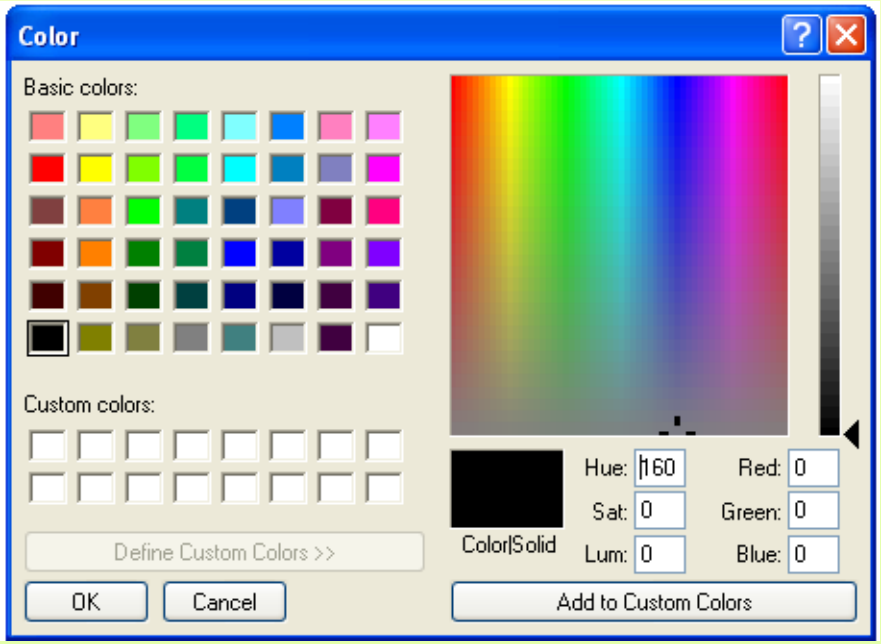
FullOpen – чи діалог відкривається з можливістю створення власних кольорів

AnyColor – чи буде діалог показувати всі допустимі кольори в списку основних кольорів.

Color – встановити чи отримати колір, вибраний користувачем

CustomColors - встановити чи отримати множину кольорів в діалозі

SolidColorOnly чи користувач буде вибирати тільки однорідні кольори



public Color **Color** { get; set; }

public struct Color

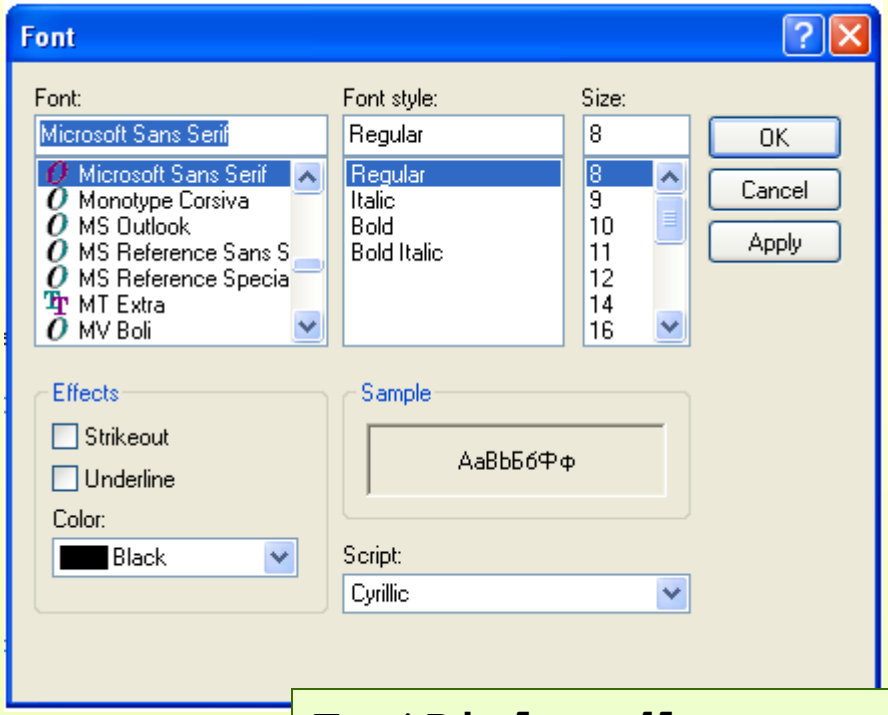
представляє ARGB колір.

FontDialog

- Дозволяє користувачу вибрати шрифт, стиль, розмір і колір і ін. Шрифта.

Public Properties:

AllowVectorFonts – чи можуть вибиратися векторні шрифти
AllowVerticalFonts - чи можуть вибиратися вертикальні шрифти
Font – вибраний фонт
FixedPitchOnly –чи в списку шрифтів будуть виводитися тільки шрифти фіксованого розміру (за замовчуванням – false)
MaxSize, MinSize – який максимальний (мінімальний) розмір шрифту можна вибрати
ShowApply – для виведення кнопки Apply – попереднього перегляду
ShowColor – можливість вибору кольору (за замовчуванням – відсутня false)
ShowEffects – можливість встановлення Strikeout і Underline для шрифту (за замовчуванням -false)
AllowScriptChange – чи можна змінювати стилі шрифту




```
FontDialog dlg = new FontDialog();

dlg.ShowApply=true;
dlg.ShowColor=true;
dlg.ShowEffects=true;
if (dlg.ShowDialog() == DialogResult.OK)
{
        textBox1.Font = dlg.Font;
}
dlg.Dispose();
```

Виведення на друк

- public class **PrintDocument** : Component

  - **Властивості**

DefaultPageSettings	Отримати чи встановити налаштування сторінки, які використовують за замовчуванням для всіх сторінок, що друкуватимуться.
DocumentName	Отримати чи встановити ім”я документа, що відображається, коли документ друкується
OriginAtMargins	Gets or sets a value indicating whether the position of a graphics object associated with a page is located just inside the user-specified margins or at the top-left corner of the printable area of the page.
PrintController	Отримати чи встановити контролера друку, що супроводжуватиме процес друку
PrinterSettings	Отримати чи встановити прінтер

- **Методи**

Print	починає процес друку
OnBeginPrint	Реалізує подію **BeginPrint.** Викликається після методу **Print** перед друком першої сторінки
OnEndPrint	Реалізує подію **EndPrint**. Викликається коли остання сторінка документа надруккується
OnPrintPage	Реалізує подію **PrintPage**. Викликається перед друком сторінки.

- **Події**

BeginPrint	Occurs when the Print method is called and before the first page of the document prints.
EndPrint	Occurs when the last page of the document has printed.
PrintPage	Occurs when the output to print for the current page is needed.
QueryPageSettings	Occurs immediately before each PrintPage event.

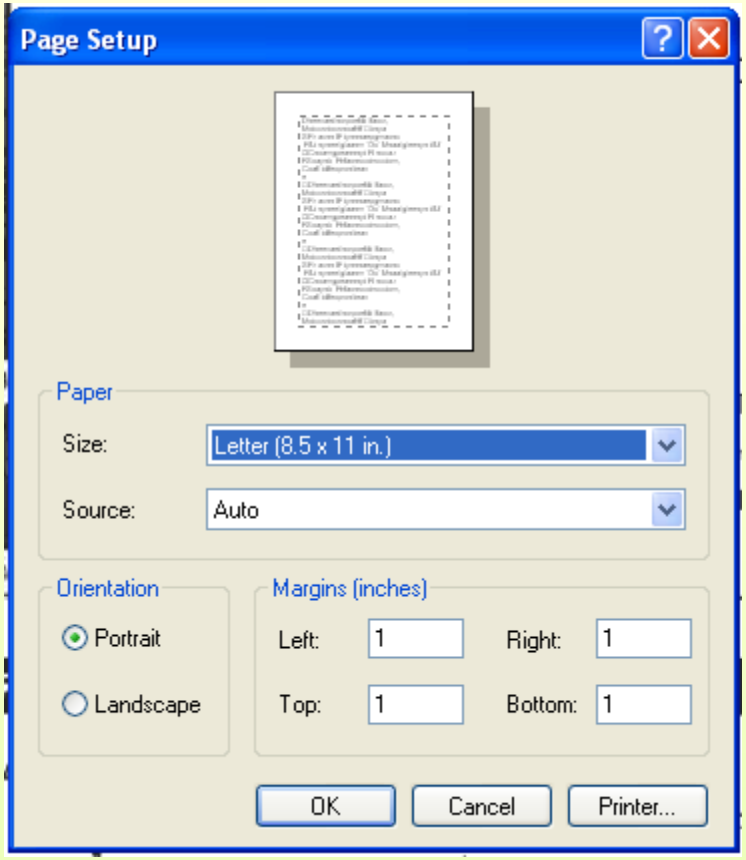
```
class Form1{
private System.Drawing.Printing.PrintDocument printDocument1;
...
            this.printDocument1 = new System.Drawing.Printing.PrintDocument();
            this.printDocument1.PrintPage += new
                        Drawing.Printing.PrintPageEventHandler(this.OnPrintPage);

private void OnPrintPage(object sender, Drawing.Printing.PrintPageEventArgs e)
        {
            e.Graphics.DrawString("Text to print", new Font("Arial", 20),
                                  Brushes.Black, 20, 20) ;
        }

        private void button1_Click(object sender, EventArgs e)
        {
            printDocument1.Print() ;
        }
```

PageSetupDialog

- дозволяє задавати розміри і джерело паперу, орієнтацію листків, розміри полів та вибирати прінтер

- **Властивості:**

AllowMargins	Чи може користувач встановлювати розмір полів
AllowOrientation	Чи може вибирати орієнтацію паперу. (встановлення – через PageSettings.Landscape)
AllowPaper	Чи може вибирати розмір паперу та джерело (PageSettings.PaperSize)
AllowPrinter	Чи буде доступна кнопка вибору прінтера **Printer**
Document	встановлює PrintDocument що буде друкуватися
MinMargins	Мінімальний розмір полів
PageSettings	Властивості сторінки
PrinterSettings	Властивості прінтера, коли натиснено Printer




```
PageSetupDialog  psdlg;
psdlg = new PageSetupDialog();
psdlg.Document = this.printDocument1;
if (psdlg.ShowDialog() == DialogResult.OK)
                {
                    .........
                }
 psdlg.Dispose();
```

PrintDialog

```
PrintDialog dlg = new PrintDialog();
dlg.Document = printDocument1;
if (dlg.ShowDialog() == DialogResult.OK)
            {
                printDocument1.Print();

            }
```