

Создание меню в PascalABC.NET

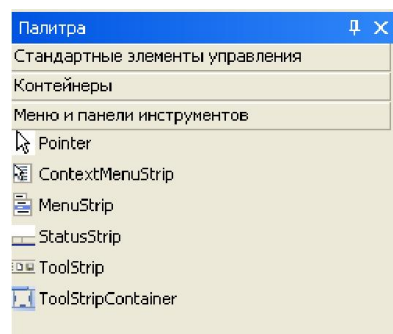
Для создания меню в Windows Forms применяется элемент **MenuStrip**. Данный класс унаследован от **ToolStrip** и поэтому наследует его функциональность.

Наиболее важные свойства компонента **MenuStrip**:

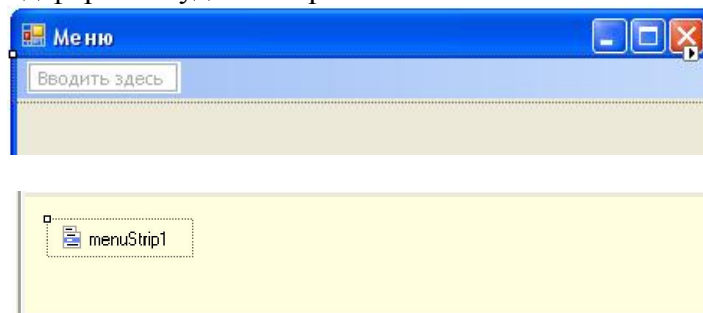
- **Dock**: прикрепляет меню к одной из сторон формы
- **LayoutStyle**: задает ориентацию панели меню на форме. Может также, как и с **ToolStrip**, принимать следующие значения
 - **HorizontalStackWithOverflow**: расположение по горизонтали с переполнением - если длина меню превышает длину контейнера, то новые элементы, выходящие за границы контейнера, не отображаются, то есть панель переполняется элементами
 - **StackWithOverflow**: элементы располагаются автоматически с переполнением
 - **VerticalStackWithOverflow**: элементы располагаются вертикально с переполнением
 - **Flow**: элементы размещаются автоматически, но без переполнения - если длина панели меню меньше длины контейнера, то выходящие за границы элементы переносятся
 - **Table**: элементы позиционируются в виде таблицы
- **ShowItemToolTips**: указывает, будут ли отображаться всплывающие подсказки для отдельных элементов меню
- **Stretch**: позволяет растянуть панель по всей длине контейнера
- **TextDirection**: задает направление текста в пунктах меню

MenuStrip выступает своего рода контейнером для отдельных пунктов меню, которые представлены объектом **ToolStripMenuItem**.

Создадим простое многоуровневое меню. Для этого выберем в Палитре элемент **MenuStrip**.

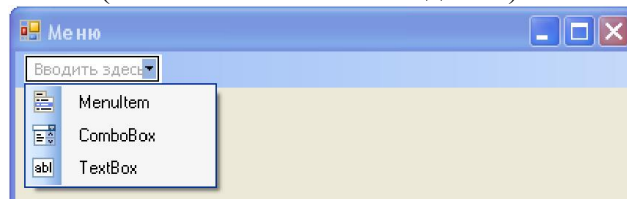


Добавим его на форму. В верхней части формы появится заготовка нашего будущего меню, а под формой будет отображаться сам элемент.

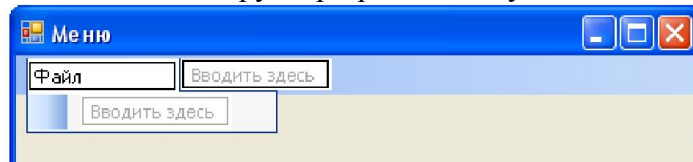


Для добавления доступно три вида элементов: **MenuItem** (объект **ToolStripMenuItem**), **ComboBox** и **TextBox**. Таким образом, в меню мы можем использовать выпадающие списки и текстовые поля, однако, как правило, эти элементы

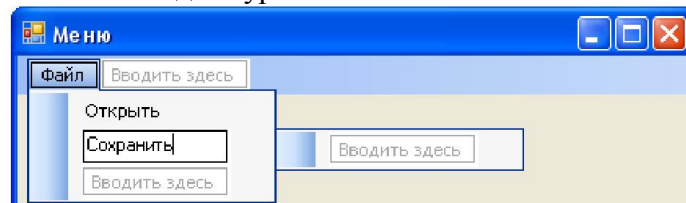
применяются в основном на панели инструментов. Меню же обычно содержит набор объектов ToolStripMenuItem. (Обычные текстовые надписи)



Если щелкнуть мышкой не по выпадающему списку, а по надписи «Вводить здесь» Можно начать создание меню. Конструктор примет следующий вид.



При вводе каждого последующего пункта, конструктор будет предлагать или добавить пункт на этом же уровне или создать уровень ниже.



Когда меню будет готово, двойным щелчком мышки по надписи можно будет создать события соответствующие этим пунктам. Программа приобретет вид:

implementation

```
procedure Form1.toolStripMenuItem3_Click(sender: Object; e: EventArgs);
```

```
begin
```

```
    // это пункт открыть
```

```
end;
```

```
procedure Form1.toolStripMenuItem4_Click(sender: Object; e: EventArgs);
```

```
begin
```

```
    // это пункт сохранить
```

```
end;
```

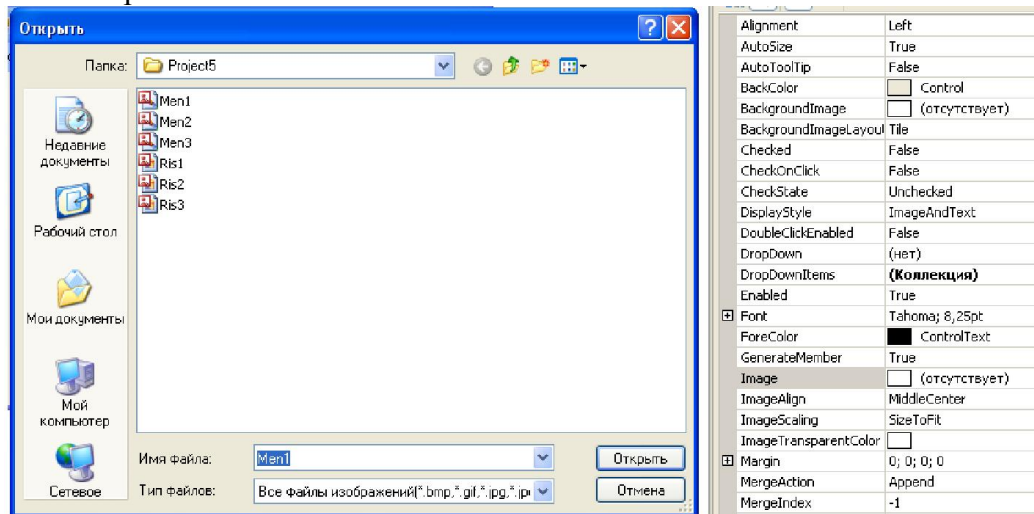
```
end.
```

Сколько будет пунктов в меню, столько будет и процедур (событий).

С помощью изображений мы можем разнообразить внешний вид пунктов меню. Для этого мы можем использовать следующие свойства:

- DisplayStyle: определяет, будет ли отображаться на элементе текст, или изображение, или и то и другое.
- Image: указывает на само изображение
- ImageAlign: устанавливает выравнивание изображения относительно элемента
- ImageScaling: указывает, будет ли изображение растягиваться, чтобы заполнить все пространство элемента
- ImageTransparentColor: указывает, будет ли цвет изображения прозрачным

Если изображение для пункта меню устанавливает в режиме дизайнера, то нам надо выбрать в окне свойство пункт Image, после чего откроется окно для импорта ресурса изображения в проект



Для демонстрации работы пунктов меню добавим компонент PictureBox. Растянем его на всю форму. В папку с проектом добавим изображения котиков, переименуем файлы в Ris1, Ris2, Ris3.



В событиях меню про котэ вставим следующие строки:

```
procedure Form1.toolStripMenuItem6_Click(sender: Object; e: EventArgs);
```

```
begin
```

```
    pictureBox1.Image := Image.FromFile('Ris1.jpg');
```

```
end;
```

Что бы разные по размеру картинки, отображались точно по размеру окна, свойство **SizeMode** для PictureBox изменим на **StretchImage**: изображение растягивается или сжимается таким образом, чтобы вместиться по всей ширине и высоте элемента PictureBox.