

Статические и динамические переменные

Цели:

- Получить представление о статических и динамических переменных.
- Рассмотреть динамическое создание переменных
- Рассмотреть динамическое создание пользовательского интерфейса.

Статические и динамические переменные

- Статические переменные.

Статические переменные создаются с учетом структуры программы. Они описываются в теле программы, память под них резервируется после запуска и освобождается после завершения программы.

Управление памятью для размещения статических переменных осуществляется автоматически.

Статические компоненты VCL размещаются на макете формы на этапе создания приложения. Такие компоненты создаются автоматически после запуска приложения.

- Динамические переменные.

Динамические переменные создаются и уничтожаются в процессе работы программы. Управление динамическими переменными осуществляется посредством указателей.

Указатель (pointer) — это переменная, диапазон значений которой состоит из адресов ячеек памяти и специального значения — нулевого адреса (nil). Значение nil показывает, что в данный момент указатель пуст и не ссылается на какую-либо ячейку памяти.

Память под динамические переменные выделяется вызовом процедуры New() и освобождается вызовом процедуры Dispose().

Для описания переменной-указателя используется специальный тип pointer, либо любой стандартный тип, перед которым устанавливается символ '^':

Var

a : pointer;	// Простой указатель
i : ^integer;	// Указатель на целое число

Для обращения к данным, на которые указывает указатель используется механизм **разыменования указателя**. В языке паскаль это осуществляется установкой символа 'Λ' после имени указателя.

Таким образом, для переменной **i : ^integer** получим:

i – статическая переменная – указатель (тип - указатель на целое - **^integer**)

i^ - динамическая переменная типа **integer** (целое).

Таким образом, программа использующая указатели может выглядеть следующим образом:

```
Var  
  i : ^integer;  
begin  
  new(i);           // Создание динамической переменной  
  i^:=5;            // Работа с динамической переменной  
  dispose(i);       // Уничтожение динамической переменной  
end.
```

- Указатели в VCL.

В VCL все типы данных, по сути, являются указателями. Статические объекты, такие как размещенные на макете формы на этапе проектирования компоненты, создаются автоматически с помощью программного кода.

В процессе работы программы можно создавать дополнительные компоненты (в том числе и визуальные), с помощью механизмов динамического управления памятью. В отличие от простых переменных, классы VCL имеют специальные методы как для создания экземпляров, так и для их уничтожения.

Для создания экземпляра класса используется конструктор **Create**, для его уничтожения – деструктор **Free**.

Следует, однако, отметить тот факт, что объект имеющий владельца (свойство **Owner**) будет уничтожен автоматически – вызовом деструктора **Free** при уничтожении родителя.

Создание динамических объектов

Запустите Delphi и создайте новое приложение. Компонент `TPanel` описан в модуле `ExtCtrls`, поэтому для работы с ним добавим его в раздел `uses`.

`uses`

`Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls, ExtCtrls;`

Перед разделом `var` добавьте раздел `const`:

`const`

`cstr = 'Йа панелько';`

`var`

`Form1: TForm1;`

Разместим на макете формы одну кнопку и зададим для нее обработчик события **OnClick** следующим образом:

```
procedure TForm1.Button1Click(Sender: TObject);  
var  
    pan : TPanel;  
begin  
    if Button1.Tag >= Length(cstr) then exit;  
    Button1.Tag := Button1.Tag + 1;  
    pan := TPanel.Create(Form1);  
    pan.Parent := Form1;  
    pan.Left := 50 + Button1.Tag * 25; pan.Top := 50;  
    pan.Width := 20; pan.Height := 20;  
    pan.Caption := cstr[Button1.Tag];  
end;
```

Скомпилируем и запустим приложение, нажмем на кнопку `Button1` несколько раз.

В данном примере динамически создается набор панелей, каждая из которых смещена на 25 точек вправо относительно предыдущей. В каждой из созданных панелей помещается одна буква из константной строки `cstr` (в свойство `Caption`).

Свойство `Tag` компонента `Button1` – это целое число. В программе оно используется для хранения количества уже выведенных панелей.

В параметре конструктора `Create` указывается `Form1` – это владелец для вновь создаваемой панели. Таким образом, данный родитель будет отвечать за уничтожение динамически созданной панели. Если в качестве владельца указать `nil`, то уничтожение панелей придется делать вручную (вызовом деструктора `Free`).

Указание значения для свойства **Parent** создаваемой панели, помещает ее в иерархию отображаемых в процессе работы объектов. Визуальный родитель (в данном случае **Form1**) вызывает методы отрисовки своих дочерних объектов.

Если это свойство установить в **nil**, то панель будет создана, но не будет видна (т.к. для нее не указан объект, на котором она должна появиться).

Указание значений для **Left, Top, Width, Height** задает позицию и размер панели. По умолчанию значения данных свойств равны нулю, соответственно панель будет не видна.