

Если Вы знаете, что функции в Delphi могут возвращать любые значения объекта, типа **TStringList**

или

TButton

, Вы также должны знать, что этим нужно пользоваться очень осторожно.

Если Вы не знаете, как написать код функции в Delphi, чтобы вернуть объект или не знаете, почему нужно быть осторожным, тогда читайте далее...

Кто должен освобождать память?

В Delphi для Win32 каждый объект, созданный вручную, также вручную должен быть освобожден. Это просто. Если Вы не освободите память от объектов, которые Вы создавали, то Ваша программа в конце концов займет всю память Windows и будет получено сообщение об ошибке **Access Violations**

Вот простая функция, которая возвращает объект Delphi - **TStringList**.

```
function GetStringList: TStringList; begin result := TStringList.Create; result.Add('an item'); end;
```

Это выглядит очень просто. Обратите внимание: так как функция возвращает **TStringList**, а

переменная неявно объявляется в каждой функции -

result

это и есть переменная

TStringList

Более того, когда Вы выходите из функции Вы создаете объект **TStringList**, не используемый в памяти. Кто-то должен освободить объект перед завершением Вашего приложения.

Так как в большинстве случаев Вы вызываете функцию, когда Вам нужен ее результат, **GetStringList**

будет вызываться из другой части кода:

```
var sl : TStringList; begin sl := GetStringList; try ShowMessage(sl[0]) ; // или  
что-то вроде этого // ListBox1.Items.Assign(sl) ; finally sl.Free; end; end;
```

Вот что важно: мы имеем переменную типа **TStringList** ("*sl*"), которая получает результат функции **GetStringList**, что-то делает с этим и наконец, удаляет список из памяти, таким образом предотвращая утечку памяти.

Предупреждение: когда функция возвращает экземпляр объекта, всегда удостоверьтесь, что объект будет освобожден, когда в нем более нет нужды!

Обратите внимание: **TStringList** не нуждается во владельце, который будет создан. Для объектов, которые нуждаются во владельце, сам владелец ответственен за объект и сам уничтожает его!