|  |
| --- |
| **Пример кода : Создание, присвоение, сортировка, и переделывание списка** |
| **unit Unit1;**  **interface**  **uses**   Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,   Dialogs, StdCtrls;  **type**   **// Определение класса клиента**   TCustomer = class     private       **// Поля данных этого нового класса**       CustomerName   : String;       CustomerNumber : Integer;      public       **// Свойства для чтения значений этих данных**       property Name : String           read CustomerName;       property Number : Integer           read CustomerNumber;        **// Коструктрор**       constructor Create(const CustomerName   : String;                          const CustomerNumber : Integer);   end;    **// Определение класса формы**   TForm1 = class(TForm)     procedure FormCreate(Sender: TObject);    private     **// TList объект мы использует в этом коде**     myList : TList;      **// Метод для показа содержимого нашего объекта списка**     procedure ShowListContents;    public    end;  **var**   Form1: TForm1;  **implementation**  **{$R \*.dfm}**   **// Конструктор Customer** **// --------------------------------------------------------------------------** **constructor TCustomer.Create(const CustomerName   : String;**                              const CustomerNumber : Integer); **begin**   **// Сохранение переданных параметров**   self.CustomerName   := CustomerName;   self.CustomerNumber := CustomerNumber; **end;**   **// Программа сортировки TList : сравните клиентов по имени** **// --------------------------------------------------------------------------** **// Возвращенное целое число имеет следующее значение :** **//** **//   > 0 : (положительное) Item1 является меньше чем Item2** **//     0 : Item1 равно Item2** **//   <0 : (negative) больше чем item2** **function compareByName(Item1 : Pointer; Item2 : Pointer) : Integer;** **var**   customer1, customer2 : TCustomer; **begin**   customer1 := TCustomer(Item1);   customer2 := TCustomer(Item2);    **// Теперь сравнение строк**   if      customer1.Name > customer2.Name   then Result := 1   else if customer1.Name = customer2.Name   then Result := 0   else Result := -1; **end;**   **// Подпрограмма для показа содержимого нашего списка** **// --------------------------------------------------------------------------** **procedure TForm1.ShowListContents;** **var**   i : Integer; **begin**   **// И повторный показ списка**   for i := 0 to myList.Count-1 do   begin     ShowMessage(TCustomer(myList[i]).Name+' is customer number '+                 IntToStr(TCustomer(myList[i]).Number));   end; **end;**  **// Конструктор формы** **// --------------------------------------------------------------------------** **procedure TForm1.FormCreate(Sender: TObject);** **var**   customer : TCustomer;  **begin**   **// Создание объекта TList для хранения набора объектов клиент**   myList := TList.Create;    **// Создание нескольких объектов клиентов и добавление их в наш объект список**   customer := TCustomer.Create('Neil Moffatt', 123);   myList.Add(customer);   customer := TCustomer.Create('Bill Gates', 64);   myList.Add(customer);    **// Мы можем добавить объект, не присваивая в промежуточную переменную**   myList.Add(TCustomer.Create('Henry Cooper', 999));   myList.Add(TCustomer.Create('Alan Sugar', 2));    **// Теперь показываем список**   ShowListContents;    **// Теперь мы сортируем список в последовательность имён и повторяем показ**   myList.Sort(compareByName);   ShowListContents;    **// Теперь удаляем и вставляем некоторые объекты**   **// Обратите внимание, что индексы начинаются с 0**   myList.Insert(2, TCustomer.Create('Added as item 3', 33));   myList.Delete(4);    **// И повторно показываем список**   ShowListContents; **end;**  **end.** |