Изучим этот класс. Класс **TStrings** предназначен для хранения набора строк. Этот класс является непосредственным потомком класса **TPersistent**, что означает в частности, что у него уже есть способность по сохранению и чтению своих данных.

Класс **TStrings** не используется в программах непосредственно (т. е. не надо пытаться создать переменную такого типа). Чаще всего он используется или как свойство некоторого компонента (например, у компонента **Memo** есть свойство**Lines** типа **TStrings**). Если же нам в программе нужен просто набор строк, то следует использовать экземпляр класса**TStringList**, который является потомком класса **TStrings**.

Рассмотрим основные свойства и методы класса **TString**. В наших примерах мы для конкретности будем иметь дело со свойством **Lines** экземпляра класса **TMemo** (компонент **Memo** расположен на вкладке **Standard** палитры компонентов). Его свойство **Lines** как раз и имеет тип **TStrings**.

Основное свойство для класса **TStrings** - это **Strings**, в которое можно записывать отдельные строки через индекс.

Код:

            Memo1.Lines.Strings[0]:='London';

            Memo1.Lines.Strings[0]:='Paris';

Свойство **Count** возвращает число элементов в списке:

ShowMessage(IntToStr(Memo1.Lines.Count));

А вот так можно пройтись по всем строкам, входящим в **TStrings**:

var

            i: Integer;

            ...

            for i := 0 to Memo1.Lines.Count - 1 do

            ShowMessage(Memo1.Lines[i]);

Свойство **IndexOf** позволяет выяснить, на каком месте в списке находится некоторая строка (и есть ли она вообще).

Вот пример его использования:

var

            k: Integer;

begin

*//Получаем номер строки "Igor".*

            k:=Memo1.Lines.IndexOf('Igor');

            ShowMessage(IntToStr(k));

end;

Свойство **IndexOf** равно номеру места, на котором находится нужная строка (нумерация идет с нуля). Если такой строки вообще нет, то свойство **IndexOf** будет равно -1.

Методы **Add** и **Insert** служат для добавления строк в конец и на определенное место соответственно.

Вот пример их применения:

*//Добавление в конец списка.*

            Memo1.Lines.Add('Roma');

*//Добавление на второе место.*

            Memo1.Lines.Insert(2, 'Berlin');

Удаление элемента с определенного места делаем методом **Delete**.

Вот пример его использования:

var

             i:Integer;

             ...

*//Получаем номер элемента Roma.*

            i:= Memo1.Lines.IndexOf('Roma');

*//Если элемент существует.*

            if i> -1 then

            Memo1.Lines.Delete(i);