

# Списки и коллекции

# Соотношение с изложенным материалом

- Практически любое приложение должно уметь выполнять ряд стандартных операций по обработке каких-либо данных. К ним относятся загрузка данных при открытии приложения, представление данных в удобном виде для использования внутри приложения, сохранение данных при завершении работы. Перечисленные действия необходимы и приложениям баз данных, и играм, и научным программам.
- Хранение и использование наборов значений можно обеспечить при помощи хорошо всем известных массивов. при реализации программной логики необходимо добавлять в массив новые элементы, изменять существующие и удалять ненужные. Кроме этого, часто бывает необходимо найти элемент массива по значению. Все эти операции стандартны и повторяются для наборов любых типов данных.
- Для решения перечисленных задач в Delphi доступны для использования специальные классы. Помимо хранения наборов значений в них реализованы свойства, позволяющие контролировать состояние списка и методы, обеспечивающие редактирование списка и поиск в нем отдельных элементов. Для загрузки и сохранения данных используются потоки — классы, инкапсулирующие механизмы доступа к различным хранилищам информации — файлам, памяти и т. д. Их общим предком является класс Tstream.

# План

- что такое список; как устроено основное свойство всех списков, объединяющее его элементы;
- добавление, изменение и удаление элементов списка;
- поиск заданного элемента;
- механизм выделения памяти под элементы списка;
- список строк;
- список указателей;
- чем отличается коллекция от списка;
- коллекции;
- использование потоков.

# Список строк

- TStringList - класс является абстрактным и служит платформой для создания реально работающих потомков.
- TStringList - класс реализует работоспособный список строк

# Класс TString

- Класс TString является базовым классом, который обеспечивает потомков основными свойствами и методами, позволяющими создавать работоспособные списки строк. Его прямым предком является класс TPersistent.
- Класс TString реализует все вспомогательные свойства и методы, которые обеспечивают управление списком. При этом методы, непосредственно добавляющие и удаляющие элементы списка, не реализованы и объявлены как абстрактные.
- Попытка прямого использования в приложении экземпляра класса TString вызовет ошибку применения абстрактного класса на этапе выполнения программы, а именно при попытке заполнить список значениями.

# Класс TStringList

- Класс TStringList обеспечивает реальное использование списков строк в приложении. По существу, класс представляет собой оболочку вокруг динамического массива значений списка, представленного свойством strings.

- Обращение к отдельному элементу списка может осуществляться через свойство `strings` таким образом:

```
SomeStrings.Strings[i] := Edit1.Text;
```

или так:

```
SomeStrings[i] := Edit1.Text;
```

- При помощи простого присваивания можно задавать новые значения только тогда, когда элемент уже создан.

- При помощи простого присваивания можно задавать новые значения только тогда, когда элемент уже создан.



- Для добавления нового элемента используются методы Add И AddStrings.

- Функция: `function Add(const S: string): Integer;`

добавляет в конец списка новый элемент, присваивая ему значение s и возвращая индекс нового элемента в списке.

- Метод: `procedure Append(const S: string);`

просто вызывает функцию Add. Единственное отличие заключается в том, что метод не возвращает индекс нового элемента.

- Метод: `procedure AddStrings(Strings: TStrings);`

добавляет к списку целый набор новых элементов, которые должны быть заданы другим списком, передаваемым в параметре strings.

- При необходимости можно добавить новый элемент в произвольное место списка. Для этого применяется метод:

`procedure Insert(Index: Integer; const S: string);`

который вставляет элемент s на место элемента с индексом index. При этом все указанные элементы смещаются на одну позицию вниз.

- Для удаления элемента списка используется метод

`procedure Delete(Index: Integer);`

- Метод: `procedure Move(CurIndex, NewIndex: Integer);`

перемещает элемент, заданный индексом curindex, на новую позицию, заданную индексом NewIndex.

- А метод: `procedure Exchange(Index1, Index2: Integer);`

меняет местами элементы с индексами index1 и index2.

- **Пример проекта**

В следующем примере проекта будет исследовано применение объекта **TStringList** и других производных **TStrings**, чтобы улучшить понимание, как работают различные части строковых классов. Между различными производными типов строковых классов имеется много общего. Такое сходство обусловлено тем, что упомянутые классы имеют общий родительский класс **TStrings**. Таким образом, за исключением специального поведения, определенного перекрывающимися методами, большая часть методов **TStrings** применима ко всем классам сходным образом, и после овладения общей частью класса **TStrings** легко разобраться в производных классах. Еще более важно то, что большинство перекрывающих функций написано таким образом, чтобы обеспечить более или менее функциональную эквивалентность между различными производными класса **TStrings**. При этом следует помнить о том, что любые компоненты, которые используют строковую информацию как часть изображения, поддерживают свои строковые значения в классах, произведенных от класса **TStrings**. Следовательно при разработке совершенно нового пользовательского элемента управления можно ознакомиться с реализацией некоторых стандартных элементов управления Delphi, чтобы получить ощущение способов взаимодействия элементов управления и строковых классов. В следующем примере разрабатывается новый класс, порожденный от **TStringList**. Если вы помните, то класс **TStringList** реализует метод сортировки, но сортирует всегда по возрастанию. В примере к новому классу добавляется метод, обеспечивающий возможность выполнения сортировки по убыванию. Затем новый класс используется для реализации простой программы, предоставляющей возможность открыть текстовый файл для просмотра, выполнить необязательную сортировку и сохранить файл.

Форма содержит элемент управления **Memo** и несколько кнопок, управляющих операциями сортировки. Соответствующий модуль кода, реализующий программу, показан ниже:

