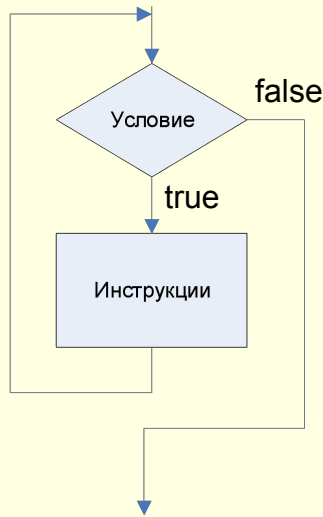


Программирование циклических алгоритмов



Инструкция цикла с предусловием



Пример:
вычислить сумму

$$S = 1 + 1/2 + 1/3 + \dots + 1/100$$

```
while <лог. выражение> do  
begin  
    <инструкции>  
end;
```

```
s:=0; n:=1;
```

```
while n<=100 do
```

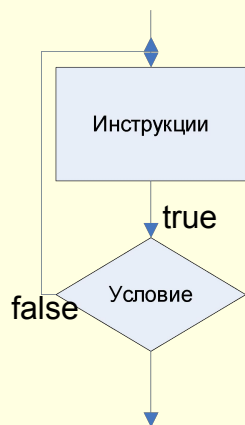
```
begin
```

```
    s:=s+1/n;
```

```
    n:=n+1;
```

```
end;
```

Инструкция цикла с постусловием



repeat

<инструкции>

until <лог. выражение>

Пример:

вычислить сумму

$$S = 1 + 1/2 + 1/3 + \dots + 1/100$$

```
s:=0; n:=1;
```

repeat

```
s:=s+1/n;
```

```
n:=n+1;
```

until n>100;

Инструкция цикла с параметром

```
for параметр:=нач_знач downto кон_знач do  
begin  
    <инструкции>  
end;
```

Пример:

вычислить сумму

$S=1 + 1/2 + 1/3 + \dots + 1/100$

```
s:=0;  
for n:=1 to 100 do  
begin  
    s:=s+1/n;  
end;
```

Сравнение циклов разных типов

While	Repeat	For
универсальный	всегда выполняется хотя бы один раз	пригоден только при заданном числе повторений
требует инициализации и изменения управляющих переменных	требует инициализации и изменения управляющих переменных	автоматически инициализирует и изменяет управляющую переменную
может приводить к заикливанию	может приводить к заикливанию	заикливание исключено

– Массивы



Массивы

Массив — это структура данных, представляющая собой набор переменных одинакового типа, имеющих общее имя

Массивы удобно использовать для хранения однородной по своей природе информации, например, таблиц и списков и т.п.

Одномерные массивы

Объявление массива в разделе *var*

имя: array [начальный_индекс .. конечный_индекс] of тип

где:

имя — имя массива;

array — зарезервированное слово языка Delphi, обозначающее, что объявляемая переменная является массивом;

начальный_индекс и конечный_индекс – константы типа integer или char, определяющие границы массива;

тип — тип элементов массива.

Примеры объявления массивов:

```
koef      : array[0..99] of integer;
```

```
name      : array[1..30] of string;
```

```
temper    : array['a'.. 'z'] of real;
```


Использование одномерных массивов в программах

Обращение к конкретному элементу массива производится по общему имени и номеру элемента, заключенному в квадратные скобки, например:

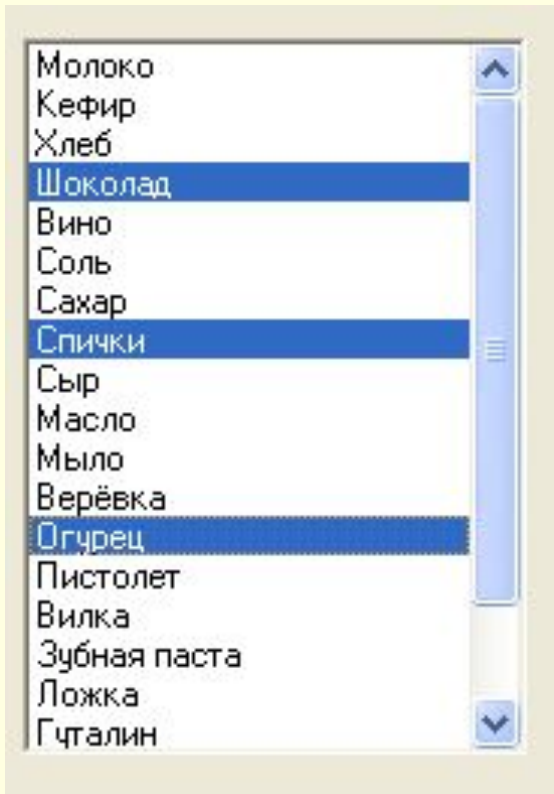
```
koef[1],  
name[15],  
temper['f']
```

Для последовательной обработки всех элементов массива чаще всего используется цикл с параметром:

```
for i:=... to ... do
```

Компоненты Delphi для работы со списками

Класс TListBox (вкладка Standard)



Компоненты этого класса представляют собой многострочные окна, подобно Мемо, но без возможности редактирования

Они предназначены для работы с одномерным массивом строк

Нумерация строк начинается с нуля

Одна или несколько строк могут быть выделены с помощью мыши

Основные свойства компонентов класса TListBox

Свойство	Тип	Пояснение	Пример
----------	-----	-----------	--------

Основные методы компонентов класса TListBox

Метод	Пояснение	Пример
-------	-----------	--------

Пример 1: заполнение списка значениями функции $\frac{1}{\sqrt{x^2-1}}$ на отрезке [2, 3] с шагом 0,01

```
var
...
i : integer;
x : real;
...
x:=2;
for i := 0 to 100 do
begin
    ListBox1.Items.Add(FloatToStr(1/sqrt(x*x-1)));
    x:=x+0.01;
end;
```

Возможен и такой вариант:

```
x:=2;
while x<=3 do
begin
    ListBox1.Items.Add(FloatToStr(1/sqrt(x*x-1)));
    x:=x+0.01;
end;
```

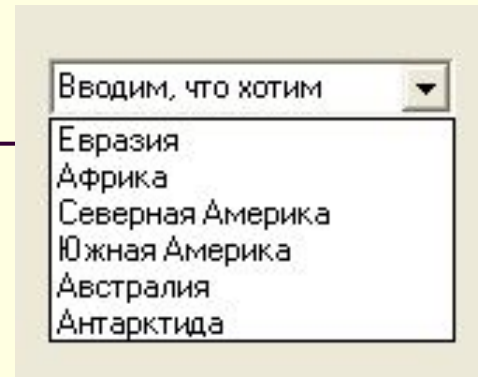
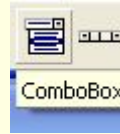
Пример 2: копирование выделенных строк из одного списка в другой

```
for i := 0 to ListBox1.Count-1 do
begin
    if ListBox1.Selected[i] then
        begin
            ListBox2.Items.Add(ListBox1.Items[i]);
        end;
    end;
end;
```

А если нужно не скопировать, а перенести, то так:

```
i := 0;
while i <= ListBox1.Count-1 do
begin
    if ListBox1.Selected[i] then
        begin
            ListBox2.Items.Add(ListBox1.Items[i]);
            ListBox1.Items.Delete(i);
        end
    else i := i + 1;
end;
end;
```

Класс TComboBox (вкладка Standard)



Компоненты этого класса представляют собой выпадающий список

Имеют свойства Items и Count как у класса TListBox

Имеют все методы класса TListBox

Свойства, отсутствующие у класса TListBox:

Свойство	Тип	Пояснение
Text	String	Текст, видимый в окне нераскрытого списка
ItemIndex	Integer	Номер видимой строки или -1, если введена отсутствующая в списке строка

Пример 1: пополнение списка элементом, введенным пользователем

```
ComboBox1.Items.Add(ComboBox1.Text);
```

Пример 2: пополнение списка элементом, введенным пользователем, если этот элемент отсутствует в списке

```
if ComboBox1.ItemIndex = -1 then  
    ComboBox1.Items.Add(ComboBox1.Text);
```

Пример 3: подсчет количества нулевых элементов в списке

```
k:=0;  
for i := 0 to ComboBox1.Count-1 do  
begin  
    if ComboBox1.Items[i] = '0' then  
        k := k + 1;  
end;  
Edit2.Text:= IntToStr(k); {вывод результата на экран}
```

Лекция окончена
Спасибо за внимание

