

Лабораторная работа № 5

Работа с файлами

Цель работы: изучить приёмы работы с текстовыми файлами.

Часть 1

Задание:

1. Создать текстовый файл (имя – любое, расширение – *.txt*) в любом редакторе, записав в него любимое стихотворение, в котором не менее четырех четверостиший. Четверостишия отделять друг от друга пустыми строками.
2. Считать содержимое файла в компонент *Memo*, не используя его метод чтения *LoadFromFile*.
3. Найти и вывести строки, которые начинаются на заданную букву, нумеруя их последовательно, начиная с единицы.
4. Подсчитать общее количество слов, содержащих заданную букву.
5. Выполнить самостоятельное задание согласно варианту.

Теоретические сведения

Файл – это именованная структура данных, представляющая собой последовательность элементов данных одного типа, расположенных на диске или любом другом внешнем носителе.

Для файла существует понятие текущей позиции. Она показывает номер элемента, который будет прочитан или записан при очередном обращении к файлу. Чтение-запись каждого элемента продвигает текущую позицию на единицу вперед. Для большинства файлов можно менять текущую позицию чтения-записи, выполняя прямой доступ к его элементам.



Для работы с текстовым файлом в программе объявляется файловая переменная (логический файл) в разделе описания переменных:

*<имя_файловой_переменной > : **TextFile**;*

После объявления файловой переменной необходимо связать ее с конкретным физическим файлом. Для этого используется процедура:

***AssignFile** (var *f* : **file**; <имя_файла>:**String**);*

Практическая часть

Пусть текстовый файл, созданный в блокноте, имеет имя *Test.txt* и находится в том же каталоге, что и проект.

Подготовка к выполнению работы

Создаем новый проект и изменяем у формы свойство *Caption* на название лабораторной работы. Сохраняем проект в рабочую папку командой *File* → *Save Project as*.

Формирование интерфейса программы

Размещаем на форме компоненты:

- *Memo* – для отображения текста (*Name* = '*MemoText*', *Lines* = '');
 - *Label* – для комментария к *Memo* (*Caption* = '*Исходный текст*');
 - *Button* – для открытия файла и считывания текста (*Name* = '*ButtonRead*'; *Caption* = '*Открыть*');
 - *Button* – для поиска строк текста (*Name* = '*ButtonFind*'; *Caption* = '*Найти*');
 - *Edit* – для ввода символа поиска строк текста (*Name* = '*EditLetter*', *MaxLenght* = 1 – максимальная длина вводимой строки – один символ);
 - *OpenDialog* – для ввода имени текстового файла (*Title* = '*Открыть текстовый файл*', *Filter* = '*Текстовые файлы|*.txt*');
 - *Memo* – для отображения результатов поиска (*Name* = '*MemoResult*', *Lines* = '');
 - *Label* – для комментария к *Memo* (*Caption* = '*Результаты поиска*').
 - *Label* – для вывода количества слов (*Name* = '*LabelKol*', *Visible* = *false*).
- Примерный вариант интерфейса программы показан на рис. 5.1.

Открытие файла

Создаем процедуру обработчика щелчка мышью на кнопке *Открыть*.

Поскольку компонент *TOpenDialog* является невизуальным, для его активации используем метод *Execute* (переменную *f* : *TextFile* необходимо описать в секции *var*):

```
if OpenDialog1.Execute then  
begin  
  AssignFile(f, OpenDialog1.FileName);  
  Reset(f);  
  CloseFile(f);  
end;
```

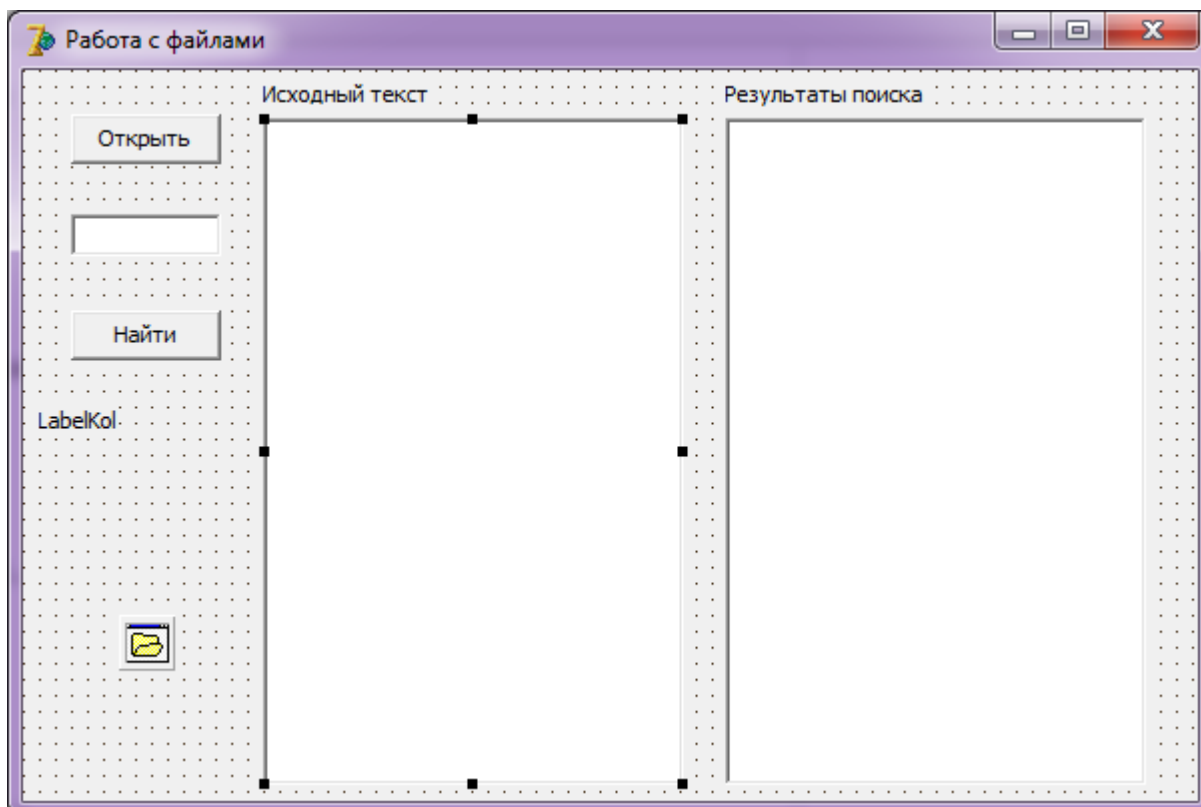


Рис. 5.1. Интерфейс программы



Невизуальные компоненты чаще всего представляют собой компоненты, позволяющие получить доступ к диалоговым окнам, которые позволяют управлять различными режимами работы программы и сообщать ей необходимую информацию. Работа со стандартными диалоговыми окнами осуществляется в три этапа.

Вначале на форму помещается соответствующий компонент и осуществляется настройка его свойств. В данном случае это компонент **TOpenDialog**. Настройка свойств может проходить как на этапе конструирования, так и в ходе выполнения программы.

На втором этапе осуществляется вызов стандартного для диалоговых компонентов метода *Execute*, который отображает соответствующее диалоговое окно. Этот метод является логической функцией булевского типа. Она возвращает значение *True*, если результат диалога с пользователем был успешным (нажата кнопка *OK*), и *False* – в противном случае (кнопка *Cancel*).

Считывание строк из файла

Для считывания строк из текстового файла удобно использовать цикл с предусловием *while*, а для записи строк в *Memo* – метод *Add* (переменную *s : string* необходимо описать в секции *var*). Прежде чем заполнять *Memo* строками из текстового файла, свойство *Lines* необходимо очистить.

```
MemoText.Lines.Clear;  
While not eof(f) do  
begin  
  Readln(f, s);  
  MemoText.Lines.Add(s);  
end;
```



Для считывания данных из текстового файла у компонента **Memo** есть специальный метод **LoadFromFile**. Если бы по условию задания его можно было использовать, то все написанные выше команды можно было бы заменить одной:

```
MemoText.Lines.LoadFromFile('Test.txt');
```

После считывания открытый файл необходимо закрыть процедурой **CloseFile**.

Для того, чтобы можно было вводить букву сразу после считывания содержимого файла, передадим фокус вводу компоненту **EditLetter**:

```
EditLetter.SetFocus;
```

Поиск строк

Создаем процедуру обработчика щелчка мышью на кнопке **Найти**. Прежде чем написать программу для поиска нужных строк, необходимо проверить два условия:

- считана ли информация в компонент **MemoText**;
- введена ли буква в компонент **EditLetter**.

После этого надо «пройти» по всем строкам в **Memo**, сравнить первую букву каждой строки с введенной буквой, и при их совпадении записать строку в **MemoResult**. Поскольку сравниваемые буквы могут быть как заглавными, так и строчными, приведем их к заглавным буквам функцией **AnsiUpperCase**.

По условия задачи строки необходимо нумеровать. Для этого введем переменную **nom**, которую обнулим перед началом цикла. В дальнейшем перед каждой записью строки в **MemoResult** будем увеличивать ее на единицу.

В считываемом файле могут находиться пустые строки, которые надо просто пропускать. Для этого используется оператор **continue**, который осуществляет переход на следующий шаг цикла.

```
Nom:= 0;  
for i := 0 to MemoText.Lines.Count - 1 do  
begin
```

```

s:= MemoText.Lines[i];
if s = '' then continue;
if AnsiUpperCase(s[1]) = AnsiUpperCase(EditLetter.Text[1]) then
begin
  nom:= nom + 1;
  MemoResult.Lines.Add(IntToStr(nom) + ' . ' + s);
end;

```

Подсчет слов

Подсчет количества слов выполняется в том же цикле. Для хранения данного значения вводим переменную *kol*, которую необходимо обнулить перед началом цикла.

По условию разделителем слов в строке является пробел, причем количество пробелов может быть произвольным. Другими словами пробел является признаком окончания всех слов строки, кроме последнего слова в строке. Чтобы пробел стал признаком окончания и последнего слова, просто добавим его к строке: *s:= s + ' '*;

Для разделения строки на слова будем использовать следующий алгоритм (переменная *sw* всегда содержит символы текущего слова):

- Просматриваем строку посимвольно.
- Если очередной символ строки является пробелом, это означает, что в *sw* сформировано очередное слово. Надо просмотреть все его символы, сравнивая каждый с заданной буквой. При совпадении переменную *kol* увеличиваем на единицу. В любом слове заданная буква может встречаться несколько раз, поэтому сразу же после обнаружения первой буквы цикл прекращаем командой *break*. Очищаем переменную *sw*.
- Если очередной символ строки не является пробелом, просто добавляем его к переменной *sw*, накапливая, таким образом, символы текущего слова.

```

For j:= 1 to length(s) do
if s[j] = ' ' then
begin
  for k:= 1 to length(sw) do
if AnsiUpperCase(sw[k]) = AnsiUpperCase(EditLetter.Text[1]) then
begin
  kol:= kol + 1;
  break;
end;
sw:= '';
end
else
sw:= sw + s[j];
end;

```

После завершения цикла выводим полученное число *kol* с необходимыми комментариями:

```
LabelKol.Caption := 'Букву ' + EditLetter.Text + ' содержат ' +  
    IntToStr(kol) + ' слов';  
LabelKol.Visible := true;
```

Результат работы программы показан на рис. 5.2.

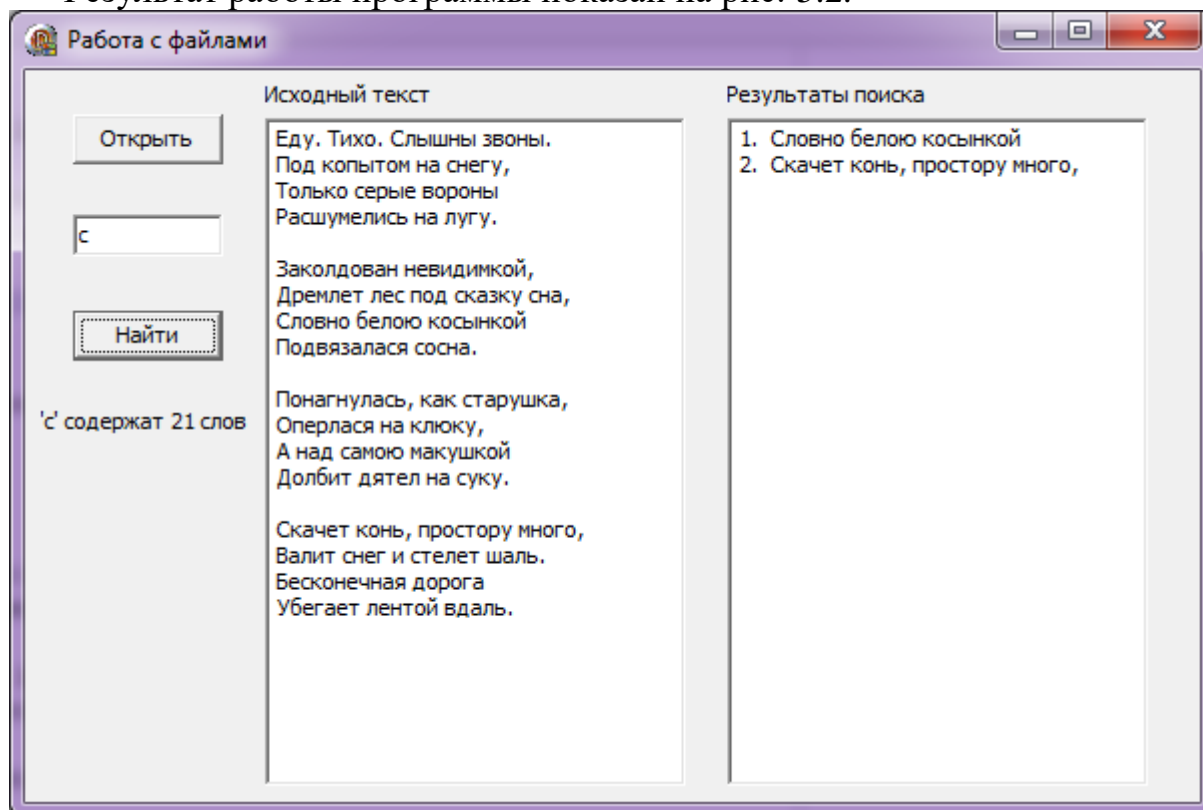


Рис. 5.2. Результат работы программы



Варианты заданий для самостоятельной работы

Задание:

В текстовом редакторе создать текстовый файл, состоящий не менее чем из 5 строк. Содержимое файла определено в задании. Каждая строка должна содержать не менее трех значений, разделенных пробелами. Количество значений в каждой строке разное. Содержимое строк зависит от номера варианта. Вывести строки файла в компонент *Memo*, не используя его метод чтения *LoadFromFile*.

Далее выполнить задание согласно варианту.



Для работы со строками используются следующие функции:

<i>Pos(Substr : String; Str : String): Integer</i>	<i>Возвращает позицию (индекс) первого вхождения Substr в строке Str. Если Substr нет в Str, возвращает 0.</i>
<i>Insert(Source: String; var S: String; Index: Integer): Integer</i>	<i>Вставляет строку Source в строку S, начиная с номера символа, равного Index</i>
<i>Delete(var S: String; Index, Count: Integer)</i>	<i>Удаляет из строки S подстроку, начинающуюся с номера символа, равного Index, и содержащую Count символов.</i>
<i>Copy(S: String; Index, Count: Integer): String</i>	<i>Возвращает подстроку строки S, начиная с номера символа, равного Index и содержащую Count символов.</i>

Вариант	Задание
1	Текстовый файл содержит целые положительные и отрицательные числа. <ul style="list-style-type: none"> Ввести целое число. Вывести строки, длина которых больше или равна этому числу. Найти сумму всех положительных и всех отрицательных чисел.
2	Текстовый файл содержит целые и вещественные числа. <ul style="list-style-type: none"> Вывести строки, в которых содержатся только целые числа. Найти среднее арифметическое всех чисел файла.
3	Текстовый файл содержит целые числа. <ul style="list-style-type: none"> Ввести целое число и вывести номера строк и сами строки, содержащие заданное число. Найти максимальное число и номер первой строки, в которой оно находится.
4	Текстовый файл содержит целые числа (положительные, отрицательные и нулевые). <ul style="list-style-type: none"> Вывести строки, в которых хотя бы одно число совпадает с первым числом строки. Ввести число и найти наиболее близкое к нему значение.
5	Текстовый файл содержит целые положительные и отрицательные числа. <ul style="list-style-type: none"> Из каждой строки удалить числа, предшествующие первому положительному числу. Если в строке нет положительных чисел, оставить её без изменений. Найти минимальное положительное и отрицательное число
6	Текстовый файл содержит слова.

	<ul style="list-style-type: none"> • Вывести первое слово каждой строки и количество слов в ней. • Найти количество слов, начинающихся и заканчивающихся на одну и ту же букву.
7	<p>Текстовый файл содержит вещественные числа (положительные, отрицательные и нулевые).</p> <ul style="list-style-type: none"> • Вывести строки, удалив из них нулевые значения. • Найти минимальное и максимальное количество чисел в строках.
8	<p>Текстовый файл содержит целые числа.</p> <ul style="list-style-type: none"> • Ввести два числа и вывести строки, которые не содержат ни одного числа из заданного интервала. • Найти сумму и количество чётных чисел в файле.
9	<p>Текстовый файл содержит слова.</p> <ul style="list-style-type: none"> • Удалить первое слово из каждой строки файла и вывести получившиеся строки. • Ввести символ и подсчитать количество слов, оканчивающихся на этот символ.
10	<p>Текстовый файл содержит слова.</p> <ul style="list-style-type: none"> • Перед первым словом поместить номер строки, затем в круглых скобках ее длину, затем через пробел саму строку. • Ввести число и подсчитать количество слов, длина которых равна этому числу.
11	<p>Текстовый файл содержит целые числа.</p> <ul style="list-style-type: none"> • Найти и вывести максимальные элементы в каждой строке исходного файла. • Подсчитать среднее арифметическое найденных максимальных элементов.
12	<p>Текстовый файл содержит слова.</p> <ul style="list-style-type: none"> • Найти и вывести максимальную длину слов файла. • Вывести все слова, имеющие длину, большую или равную максимальной длине минус три.
13	<p>Текстовый файл содержит вещественные числа (положительные, отрицательные и нулевые).</p> <ul style="list-style-type: none"> • Удалить все числа, расположенные за первым положительным числом и вывести полученные строки. Если положительное число является последним в строке, то оставить ее без изменения. • Найти максимальный элемент среди положительных чисел и номер строки, в которой он первый раз встретился.
14	<p>Текстовый файл содержит целые числа.</p> <ul style="list-style-type: none"> • Удалить из файла все числа, меньшие заданного числа, и вывести полученные строки. Если в строке таких чисел нет, оставить ее без изменений. • Найти максимальное из удаленных чисел, его номер строки и

	номер позиции в строке.
15	<p>Текстовый файл содержит слова.</p> <ul style="list-style-type: none"> • В начало каждой строки, содержащей заданное слово, добавить номер этого слова в строке. • Подсчитать количество слов, начальная и конечная буква которых совпадают с начальной и конечной буквами заданного слова.
16	<p>Текстовый файл содержит информацию о студентах в следующем формате: фамилия (20 позиций), группа (10 позиций), адрес (30 позиций), год рождения (4 позиции).</p> <ul style="list-style-type: none"> • Ввести две даты (года) и вывести информацию обо всех студентах, родившихся в заданный период. • Вывести средний возраст студентов на текущий год.
17	<p>Текстовый файл содержит целые числа.</p> <ul style="list-style-type: none"> • Найти минимальный и максимальный элементы в файле. • Найти все строки, содержащие минимальный или максимальный элементы. Вывести эти строки, записав в их начало минимальный или максимальный элемент соответственно.
18	<p>Текстовый файл содержит слова.</p> <ul style="list-style-type: none"> • Найти максимальную и минимальную длину строки. • Вывести строки максимальной и минимальной длины, добавив в начало каждой ее номер в исходном файле.
19	<p>Текстовый файл содержит целые положительные и отрицательные числа.</p> <ul style="list-style-type: none"> • Извлечь квадратный корень из каждого числа в тех строках, все элементы которых имеют положительные значения. Вывести полученные строки. • Подсчитать и вывести количество изменённых строк и общее количество чисел, из которых извлекался квадратный корень.
20	<p>Текстовый файл содержит целые числа.</p> <ul style="list-style-type: none"> • Подсчитать среднее арифметическое всех чисел. • Вывести строки, содержащие значения отклонения каждого числа от среднего арифметического значения.
21	<p>Текстовый файл содержит слова, в том числе заключенные в круглые скобки. Предполагается, что внутри каждой пары скобок может быть несколько слов, но не может быть вложенных скобок.</p> <ul style="list-style-type: none"> • Удалить текст, расположенный в скобках, включая сами скобки, и вывести полученные строки. • Вывести количество удалений.
22	<p>Текстовый файл содержит слова.</p> <ul style="list-style-type: none"> • Подсчитать количество слов в каждой строке и вставить полученное значение в начало строки. • Вывести среднее арифметическое значение количества слов в

	строках.
23	<p>Текстовый файл содержит слова.</p> <ul style="list-style-type: none"> • Ввести строку не более чем из трех символов. В строках файла каждое вхождение любого из введенных символов в слово заменить его номером 1, 2 или 3. • Вывести количество замен каждого вида с комментариями.
24	<p>Текстовый файл содержит слова.</p> <ul style="list-style-type: none"> • Ввести целое число и удалить из каждой строки слово, порядковый номер которого равен этому числу. Если введенное значение превышает количество слов в строке, то соответствующую строку оставить без изменений. • Вывести количество измененных и оставшихся без изменений строк.
25	<p>Текстовый файл содержит целые числа.</p> <ul style="list-style-type: none"> • Заменить в каждой строке элементы с наибольшим значением нулём. • Вывести максимальное и минимальное значения среди замененных чисел.
26	<p>Текстовый файл содержит информацию о студентах в следующем формате: фамилия (20 позиций), группа (10 позиций), адрес (30 позиций), год рождения (4 позиции).</p> <ul style="list-style-type: none"> • Ввести несколько букв и вывести информацию обо всех студентах, фамилии которых начинаются на эти буквы. • Вывести фамилии самых молодых студентов.
27	<p>Текстовый файл содержит целые положительные и отрицательные числа.</p> <ul style="list-style-type: none"> • Вывести строки, в которых количество отрицательных чисел не превышает количество положительных чисел. • Вывести количество положительных и количество отрицательных чисел в файле.
28	<p>Текстовый файл содержит слова.</p> <ul style="list-style-type: none"> • В конец каждой строки поместить минимальную длину слова в ней. • Вывести количество слов минимальной длины.
29	<p>Текстовый файл содержит слова.</p> <ul style="list-style-type: none"> • Для каждой строки подсчитать и вывести количество символов в ней, отличных от пробела. • Найти максимальное, минимальное и среднее арифметическое значение данных чисел.
30	<p>Текстовый файл содержит целые положительные и отрицательные числа.</p> <ul style="list-style-type: none"> • Извлечь квадратный корень из положительных чисел, а отрицательные числа заменить их абсолютными значениями.

- | |
|--|
| <ul style="list-style-type: none">• Вывести средние арифметические значения для положительных и отрицательных чисел. |
|--|

Часть 2

Задание:

1. Даны вещественные числа A , B и целое число N . Создать таблицу (компонент *StringGrid*), содержащую заданное количество значений функций $\sin(x)$ и $\cos(x)$ на промежутке $[A, B]$. Таблица состоит из трех столбцов: значение аргумента x , значение $\sin(x)$ и значение $\cos(x)$.

2. В последнюю строку таблицы поместить средние арифметические значения по каждому столбцу.

3. Создать текстовый файл (имя – любое, расширение – *.txt*) и записать в него полученные результаты, округленные до 0,001.

4. Выполнить самостоятельное задание согласно варианту.

Подготовка к выполнению работы

Создаем новый проект и изменяем у формы свойство *Caption* на название лабораторной работы. Сохраняем проект в рабочую папку командой *File* → *Save Project as*.

Формирование интерфейса программы

Создаем интерфейс программы в соответствии с рис. 5.3. Имена основных компонентов:

- *EditA*;
- *EditB*;
- *EditN*;
- *StringGridFunc*;
- *ButtonCalc*;
- *ButtonSave*.

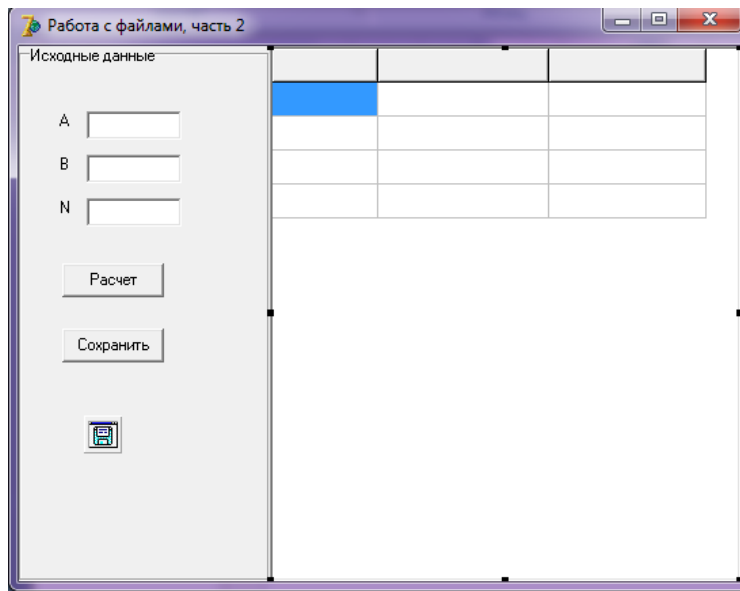


Рис. 5.3. Интерфейс программы

Массив ячеек таблицы *StringGrid* не доступен в инспекторе объектов, поэтому заголовок таблицы необходимо оформить в процессе работы программы. Это можно сделать в событии *OnCreate* формы, которое возникает при ее создании, поэтому в нем удобно выполнять сделать какие-либо предварительные действия. В данном случае они будут выглядеть следующим образом:

```
procedure TForm1.FormCreate(Sender: TObject);
begin
  StringGridFunc.Cells[0, 0]:= 'x';
  StringGridFunc.Cells[1, 0]:= 'sin(x)';
  StringGridFunc.Cells[2, 0]:= 'cos(x)';
end;
```

Ввод исходных данных

Первая операция, которую необходимо сделать в обработчике события нажатия кнопки «Расчет» – это проверка наличия введенных символов:

```
if (Trim(EditA.Text) = '') or (Trim(EditB.Text) = '') or
  (Trim(EditN.Text) = '') then
begin
  ShowMessage('Не введены исходные данные');
  exit;
end;
```



Организируйте запрет ввода нечисловых данных. Обратите внимание, что значения A и B являются вещественными числами, а значение N – целым числом.

Для перекодировки значений используем функции *StrToFloat* и *StrToInt* (все переменные необходимо описать в секции *var*):

```
a:= StrToFloat(EditA.Text);  
b:= StrToFloat(EditB.Text);  
n:= StrToInt(EditN.Text);
```

Заполнение таблицы

Условие задачи предполагает, что конец интервала не должен совпадать с его началом. Кроме того, выполнять расчет имеет смысл, если требуется получить не менее двух значений функций. Введем соответствующие проверки:

```
if (n < 2) or (a = b) then  
begin  
  ShowMessage('Неверные данные');  
  exit;  
end;
```

Очевидно, что число, определяющее начало интервала (A), должно быть меньше числа, определяющего конец интервала (B). Если при вводе ошибочно было задано иное соотношение между ними, исправим его (переменную *step* в данном случае используем в качестве временной переменной):

```
if a > b then  
begin  
  step:= a;  
  a:= b;  
  b:= step;  
end;
```

Для определения шага вычисления значений функций надо разделить длину интервала на количество значений минус один: *step*:= ($b - a$) / ($n - 1$);

Количество строк таблицы равно количеству значений (n) плюс строка-заголовок: *StringGridFunc.RowCount*:= $n + 1$;

Далее в цикле вычисляем значения функций и заносим их в таблицу:

```
for i:= 1 to n do  
begin  
  StringGridFunc.Cells[0, i]:= FloatToStr(a);
```

```
StringGridFunc.Cells[1, i]:= FloatToStr(sin(a));  
StringGridFunc.Cells[2, i]:= FloatToStr(cos(a));  
a:= a + step;  
end;
```

Вычисление средних арифметических значений

Среднее арифметическое – это отношение суммы некоторых чисел к их количеству. Для его вычисления надо отдельно сложить все значения, расположенные в столбцах таблицы, и каждую сумму разделить на одно и то же число – количество значений.

Это можно сделать непосредственно при заполнении таблицы, добавив команды суммирования в предыдущий цикл. Однако для закрепления навыков работы с таблицами сделаем расчет в отдельном цикле. Для суммирования введем переменные *sumx*, *sumsin* и *sumcos*, которые перед началом цикла надо обнулить:

```
sumx := 0;  
sumsin := 0;  
sumcos:= 0;  
for i:= 1 to StringGridFunc.RowCount - 1 do  
begin  
sumx := sumx + StrToFloat(StringGridFunc.Cells[0, i]);  
sumsin := sumsin + StrToFloat(StringGridFunc.Cells[1, i]);  
sumcos := sumcos + StrToFloat(StringGridFunc.Cells[2, i]);  
end;
```

Средние арифметические значения запишем в те же переменные, что и суммы:

```
sumx := sumx / n;  
sumsin := sumsin / n;  
sumcos:= sumcos / n;
```

Для того чтобы записать найденные значения в последнюю строку таблицы, количество ее строк надо увеличить на единицу. Номер последней строки таблицы всегда на единицу меньше, чем количество строк в ней:

```
StringGridFunc.RowCount := StringGridFunc.RowCount + 1;  
StringGridFunc.Cells[0, StringGridFunc.RowCount - 1]:= FloatToStr(sumx);  
StringGridFunc.Cells[1, StringGridFunc.RowCount - 1]:= FloatToStr(sumsin);  
StringGridFunc.Cells[2, StringGridFunc.RowCount - 1]:= FloatToStr(sumcos);
```

Запись таблицы в файл

Создаем обработчик *OnClick* для кнопки *Сохранить*. В нем определяем файловую переменную *f : TextFile* и вызываем диалог сохранения файла *SaveDialog1.Execute*.

После выбора имени файла и подтверждения выбора необходимо связать файловую переменную с именем файла процедурой *AssignFile(f, SaveDialog1.FileName)* и создать новый файл процедурой *Rewrite(f)*.

Выходной файл должен иметь следующую структуру (все значения округляются до 0,001):

```
x      sin(x) cos(x)
1.000  0.841  0.540
```

```
...   ...   ....
2.000  0.909 -0.416
```

```
-----
1.500  0.875  0.062
```

Вначале записываем заголовок, отведя на каждый столбец 12 позиций:

```
WriteLn(f, ' ', 'x', ' ', 'sin(x)', ' ', 'cos(x)');
```

После вывода всех значений надо «изобразить» подчеркивание и только потом вывести средние значения, которые находятся в последней строке таблицы. Организуем цикл от первой строки таблицы (в нулевой строке находится заголовок) до предпоследней строки, номер которой *StringGridFunc.RowCount - 2*.

Вначале преобразуем текстовые строки в вещественные числа (*x, sx, cx: real*), а затем запишем их в файл одной командой, указав в ней формат вывода:

```
for i:= 1 to StringGridFunc.RowCount - 2 do
begin
  x := StrToFloat(StringGridFunc.Cells[0, i]);
  sx:= StrToFloat(StringGridFunc.Cells[1, i]);
  cx:= StrToFloat(StringGridFunc.Cells[2, i]);
  WriteLn(f, x : 12 : 3, sx : 12 : 3, cx : 12 : 3);
end;
```

Каждый столбец занимает 12 позиций, поэтому для подчеркивания выведем 12 знаков минус без перевода курсора на новую строку, а после окончания цикла просто переведем его на новую строку в файле:

```
for i:= 1 to 36 do write(f, '-');
writeln(f);
```

Аналогично выводу значений таблицы выведем средние арифметические значения:

```
x := StrToFloat(StringGridFunc.Cells[0, StringGridFunc.RowCount - 1]);
sx:= StrToFloat(StringGridFunc.Cells[1, StringGridFunc.RowCount - 1]);
cx:= StrToFloat(StringGridFunc.Cells[2, StringGridFunc.RowCount - 1]);
```

WriteLn(f, x : 12 : 3, sx : 12 : 3, cx : 12 : 3);

Последняя операция – закрытие файла: ***CloseFile(f);***



1. Даны вещественные или целые в зависимости от задания числа A и B . Дано целое число N (количество значений функций) или третье вещественное число S (шаг расчета). Создать таблицу, содержащую необходимое количество значений заданных функций на промежутке $[A, B]$. Таблица состоит из трех столбцов: значение аргумента x , значение первой функции и значение второй функции.

2. В последнюю строку и/или столбец таблицы (в зависимости от задания) поместить результаты заданных вычислений.

3. Создать текстовый файл (имя – любое, расширение – *.txt*) и записать в него полученные результаты, с точностью до 0,00001.

4. Предусмотреть в программе реакцию на случаи некорректного ввода исходных данных.

Вариант	Функции	Шаг / Кол-во	Задание
1	$\frac{\sqrt{2x-5} - \sqrt{4x+9}}{3(x+2)}$ $\frac{x \cdot \sin(x+1) - 3 \sin(x)}{(2x-6)^3}$	Шаг	Найти средние арифметические значения по всем столбцам и в последнюю строку вывести максимальное отклонение их от среднего арифметического значения
2	$\frac{(-1)^{n-1}n}{5n-6}$ $\frac{2^n}{n\sqrt{n}}$	Кол-во	В последнюю строку вывести середину интервала и максимальные значения функций
3	$\frac{\sin(x)}{\ln(x)}$ $\frac{(x+1)^3}{\sin(x)}$	Шаг	В четвертый столбец вывести единицу, если значения функций имеют разные знаки, и нуль – в противном случае
4	$\frac{\sqrt{3n+1}}{6n}$ $\left(\frac{2n+1}{3n+1}\right)^n$	Кол-во	В четвертый столбец вывести максимальные значения в соответствующих строках
5	$\frac{\sqrt{x}}{\sin(x)}$ x	Шаг	В четвертый столбец вывести единицу, если значение первой функции в соответствующей строке больше

			значения второй функции, и нуль – в противном случае.
6	$\frac{\sin(2x)}{x^3+1}$ $\frac{x \cdot \cos(x) - 5 \sin(x)}{(x-3)^2}$	Кол-во	В четвертый столбец вывести квадратные корни из значений первой функции. Если их невозможно вычислить, то ячейку оставить пустой.
7	$\frac{tg(x)}{\cos(x)}$ x	Кол-во	В последнюю строку вывести среднее значение квадратов аргумента, среднее арифметическое первой функции и сумму значений второй функции.
8	$\frac{\sin(x) \cdot \ln(x)}{2tg(x)}$ $1 - tg^2(x)$	Кол-во	В последнюю строку вывести середину интервала, сумму значений первой функции и максимальное значение второй функции
9	$\frac{3}{2} n^{\frac{2}{3}}$ $tg \frac{\pi}{4n}$	Кол-во	В четвертый столбец вывести частные от деления значений первой функции на соответствующее значения второй функции
10	$\cos(\ln(x)) + \sin(\ln(x))$ $\frac{3x+5}{x^2-3x+2}$	Кол-во	В последнюю строку вывести середину интервала и минимальные значения функций
11	$\frac{\ln(x) + \ln(x+2)}{\ln x }$ $\frac{3,2e^x}{\ln(x)}$	Шаг	В последнюю строку вывести количество значений аргумента и суммы квадратов значений функций
12	$\left(1 + \frac{1}{n}\right)^n$ $\frac{n^2+1}{n^3+1}$	Кол-во	В четвертый столбец вывести единицу, если у аргумента первая цифра в дробной части четная, и нуль – в противном случае.
13	$\frac{\cos(x) \cdot \sin\left(\frac{x}{2}\right)}{\sin(x)}$ $1 + ctg^2(x)$	Шаг	В четвертый столбец вывести знак значения первой функции в виде символов плюс или минус, а в пятый – знак значения второй функции в том же виде
14	$\frac{1}{\sqrt{1-x^2}}$ $\frac{\cos(x)}{x^2-9}$	Кол-во	В последнюю строку вывести длину интервала и суммы значений функций
15	$\frac{2tg(x)}{1-tg(x)}$ $\frac{1+x-x^2}{\frac{x^3}{2}-4}$	Кол-во	В четвертый столбец вывести минимальные значения функций в соответствующих строках

16	$\frac{n^2 + \sqrt[4]{n}}{n\sqrt{n} - 8}$ $\frac{(2-n)^4 - (1-n)^4 + n}{(3-n)^4}$	Кол-во	В четвертый столбец вывести целые части значений первой функции, а в пятый – их дробные части
17	$\frac{1 - \cos(x)}{x^2(1+x)}$ $-x + \frac{x^2}{x+1}$	Шаг	В последнюю строку вывести количество значений аргумента и количество отрицательных значений функций
18	$\frac{\sin(x)}{\sqrt[3]{(1 - \cos(x))^2}}$ $\frac{\sqrt{x^2 + 9} - 3}{\sqrt{x+4} - 2}$	Кол-во	В четвертый столбец вывести единицу, если значение первой функции в соответствующей строке меньше значения второй функции, и нуль – в противном случае.
19	$\frac{\ln(x+1)}{(x+1)^2}$ $\frac{(-1)^n}{x^2}$	Кол-во	В четвертый столбец вывести произведения соответствующего значения первой функции на значение второй функции
20	$\frac{\sin^2(x) - \cos(2x)}{x+1}$ $\frac{x}{\sin(x)}$	Кол-во	В четвертый столбец вывести единицу, если абсолютная величина значения первой функции больше абсолютной величины значения второй функции, и нуль – в противном случае
21	$ctg(2x) - 5$ $2\cos(\sqrt{x}) + 3$	Шаг	В последнюю строку вывести длину интервала, максимальное абсолютное значение первой функции и количество положительных значений второй функции
22	$\frac{10}{x^2 \sin^2(x)}$ $tg\left(\frac{x}{3}\right)$ $\frac{2}{x} \sin(x)$	Кол-во	В последнюю строку вывести значение шага, максимальное значение первой функции и минимальное значение второй функции
23	$\frac{e^x}{(x-5)^5}$ $\frac{x \cdot \cos(x) - 5 \sin(x)}{\ln(x-1)}$	Кол-во	В четвертый столбец вывести единицу, если значение хотя бы одной функции меньше значения аргумента, и нуль – в противном случае
24	$\frac{4 + \cos\left(\frac{\pi}{4}n\right)}{n^2}$ $\frac{\ln(2)}{\ln(n)}$	Кол-во	В четвертый столбец вывести символ < (меньше), если значение первой функции в соответствующей строке меньше значения второй функции, и символ > (больше) – в противном случае

25	$\frac{5x^3 \cos(x)}{(x+2)^3 \sin(x+1)}$ $1 + \frac{1}{\operatorname{tg}(x)}$	Шаг	В четвертый столбец вывести единицу, если значение первой функции меньше значения аргумента, а значение второй функции больше аргумента, и нуль – в противном случае
26	$\frac{x\sqrt{2} \cdot \cos\left(\frac{\pi}{4} - \ln(x)\right)}{x}$ $2 + \frac{2}{\sin(x)}$	Кол-во	В четвертый столбец вывести максимальные значения функций.
27	$\frac{2x+3}{3x+1,5} \operatorname{tg}(x)$ $\frac{\sin^3(x)}{1 - \cos(x)}$	Шаг	В четвертый столбец вывести единицу, если значения функций имеют одинаковые знаки, и нуль – в противном случае
28	$\frac{1 + \cos(x)}{1 - \cos(x)}$ $\operatorname{tg}^2(x) + \operatorname{ctg}^2(x)$	Кол-во	В последнюю строку вывести длину интервала, минимальное значение первой функции и сумму значений второй функции
29	$\frac{1}{2} - \ln(5 - 2x)$ $\frac{\cos^6(x)}{\sin^4(x)}$	Кол-во	В четвертый столбец вывести единицу, если значение обеих функций меньше значения аргумента, и нуль – в противном случае
30	$\frac{\sqrt{3} - 2 \sin(x)}{\cos(x) - 1}$ $\frac{\operatorname{tg}(x) \cdot (1 + \operatorname{tg}(x))}{4 - 2x^2}$	Кол-во	В четвертый столбец вывести минимальные значения функций, а в пятый – знаки этих значений в виде символов плюс или минус



- 1) *Что такое файл?*
- 2) *Как объявляется файловая переменная в программе? Как объявляется файловая переменная для работы с текстовым файлом?*
- 3) *Как осуществляется запись и чтение из текстового файла?*
- 4) *Назовите основные процедуры и функции работы со строками.*
- 5) *Опишите алгоритм нахождения заданного слова в текстовом файле.*
- 6) *Назовите основные свойства компонента **StringGrid**.*
- 7) *Опишите алгоритм нахождения заданной строки в ячейках компонента **StringGrid**.*
- 8) *Опишите алгоритм переноса строк компонента **StringGrid** в строки компонента **Memo**.*

