С. В. Журавлев

МЕТОДИКА ПРОВЕДЕНИЯ ЗАНЯТИЙ ПО ДИСЦИПЛИНЕ «ПРАКТИКУМ ПО РЕШЕНИЮ ЗАДАЧ НА ЭВМ» НА ТЕМУ «ТЕКСТОВЫЕ ФАЙЛЫ» НА ОСНОВЕ КОМПЕТЕНТНОСТНОГО, ИНФОРМАЦИОННОГО И ДЕЯТЕЛЬНОСТНОГО ПОДХОДОВ (ЭТАП ПРИОБЩЕНИЯ)

Все задачи данной темы ориентированы на формирование следующих компетенций:

владение культурой мышления, способность к обобщению, анализу, восприятию информации, постановке цели и выбору путей её достижения (ОК-1) — студенты обобщают знания и отрабатывают навыки работы с текстовыми файла-ми, учатся анализировать методы подсчёта количества указанных символов в файле, длины слов и строк файла, выделения из файла чисел и последующей работы с ними (определения количества, вычисления суммы, произведения и т. д.), нахождение слов, являющихся палиндромами, а также создания нового файла путём преобразования исходного, понимать условие конкретной задачи и выбирать подходящий метод для её решения;

готовность использовать основные методы, способы и средства получения, хранения, переработки информации, готовность работать с компьютером как средством управления информацией (ОК-8) — методы: абстрагирование, анализ, синтез, логический метод, восхождение от абстрактного к конкретному; способы: чтение соответствующей литературы, работа в библиотеках, работа в информационно-поисковых системах; средства: печатные издания, компьютер.

Кроме того, задачи 11-30 формируют способность работать с информацией в глобальных компьютерных сетях (ОК-9) — студенты самостоятельно находят в поисковых системах необходимые сведения: способы определения длины строк файла, создания нового файла на основе исходного.

Все задачи по данной теме подразделяются на 3 группы:

I. Задачи для ситуации полной информационной определённости:

1. Дан текстовый файл. Определить, каких букв больше, d или f.

2. Дан текстовый файл, каждая строка которого представляет число. Найти произведение чисел.

3. Дан текстовый файл. Найти слово минимальной длины в каждой чётной строке. Если в искомых строках есть несколько самых коротких слов, то вывести последнее из них.

4. Дан текстовый файл, содержащий символы латинского алфавита и числа. Определить количество нечётных чисел каждой строки исходного файла.

5. Дан текстовый файл. Определить количество слов-палиндромов в этом файле.

6. Дан текстовый файл, каждая строка которого представляет число. Найти среднее арифметическое чисел, расположенных в нечётных строках.

7. Дан текстовый файл (не менее 3 строк). Вывести палиндромы, встречающиеся в первой и последней строках.

8. Дан текстовый файл (не менее 6 строк), содержащий символы латинского алфавита и числа. Вычислить среднее арифметическое чисел каждой третьей строки.

9. Дан текстовый файл. Найти слово максимальной длины в строке, номер которой задаётся пользователем. Если в строке есть несколько самых длинных слов, то вывести последнее из них.

10. Дан текстовый файл. Определить, каких букв меньше, g или t.

II. Задачи для ситуации частичной информационной неопределённости:

11. Дан текстовый файл. В каждой чётной строке определить минимальную длину слов.

12. Дан текстовый файл (не менее 3 строк). Удалить из него 2 последних строки.

13. Дан текстовый файл. Удалить из него строки, начиная со строки, номер которого задаётся пользователем.

14. Дан текстовый файл. В каждой нечётной строке определить максимальную длину слов.

15. Дан текстовый файл. Определить, сколько раз стоящие рядом 2 слова начинаются на одну и ту же букву.

16. Дан текстовый файл. В строке, номер которой задаётся пользователем, определить среднюю длину слов.

17. Дан текстовый файл. Определить среднюю длину его строк.

18. Дан текстовый файл. Удалить из каждой строки последнее слово.

19. Дан текстовый файл (не менее 6 слов). Удалить из каждой третьей строки первое слово.

20. Дан текстовый файл. Определить, сколько раз встречается в файле введённое пользователем слово.

III. Задачи для ситуации полной информационной неопределённости:

21. Дан текстовый файл. Создать новый файл, утроив все буквы р исходного файла.

22. Дан текстовый файл (не менее 4 строк). Создать новый файл, удвоив каждый третий элемент второй и предпоследней строк исходного файла через пробел.

23. Дан текстовый файл. Заменить в нём все идущие подряд одинаковые символы на один символ.

24. Даны 2 текстовых файла. Создать новый текстовый файл, являющийся объединением содержимого первого файла в первоначальном виде и содержимого второго файла, символы которого записаны в обратном порядке в каждой строке.

25. Дан текстовый файл. Создать новый файл, содержащий строки первого файла, длины которых больше средней.

26. Дан текстовый файл. Создать новый файл, включающий абзацы первого файла, содержащие нечётное число строк (абзацы отделены друг от друга одной или несколькими пустыми строками).

27. Дан текстовый файл. Удалить из строки, номер которой задаётся пользователем, все знаки препинания, при этом сохраняя пробелы.

28. Даны 2 текстовых файла. Добавить после каждой строки первого файла противоположную строку второго (т. е. после первой строки первого файла последнюю строку второго, после второй строки первого файла предпоследнюю строку второго и т. д.). Если первый файл короче второго, добавлять в него строки второго столько раз, сколько строк содержит первый файл.

29. Дан текстовый файл и целое число N. Создать новый файл, включающий строки, длина которых не превосходит N.

30. Дан текстовый файл. Создать новый файл, включающий строки первого файла, начинающиеся с гласной буквы.

Теперь рассмотрим пример подробного решения трёх задач из первой группы.

1. Дан тестовый файл. Заменить в нём все нули на единицы и наоборот.

Разместим необходимые компоненты на форме. Нам понадобятся одна метка (Label1) для записи условия задачи, 2 кнопки (Button1 для преобразования файла и Button2 для окончания работы программы), 2 многострочных текстовых окна (Memo1 для отображения исходного файла и Memo2 для отображение преобразованного файла), диалоговое окно открытия файла (OpenDialog1), диалоговое окно сохранения файла (SaveDialog1).

В компоненте Label1 разместим условие задачи. Настроим компоненты Memo1 и Memo2. Для кнопки Button1 создадим надпись Решение, для Button2 – надпись Выход. Настроим компоненты OpenDialog1 и SaveDialog1.

Окончательный вид формы показан на рис. 1.

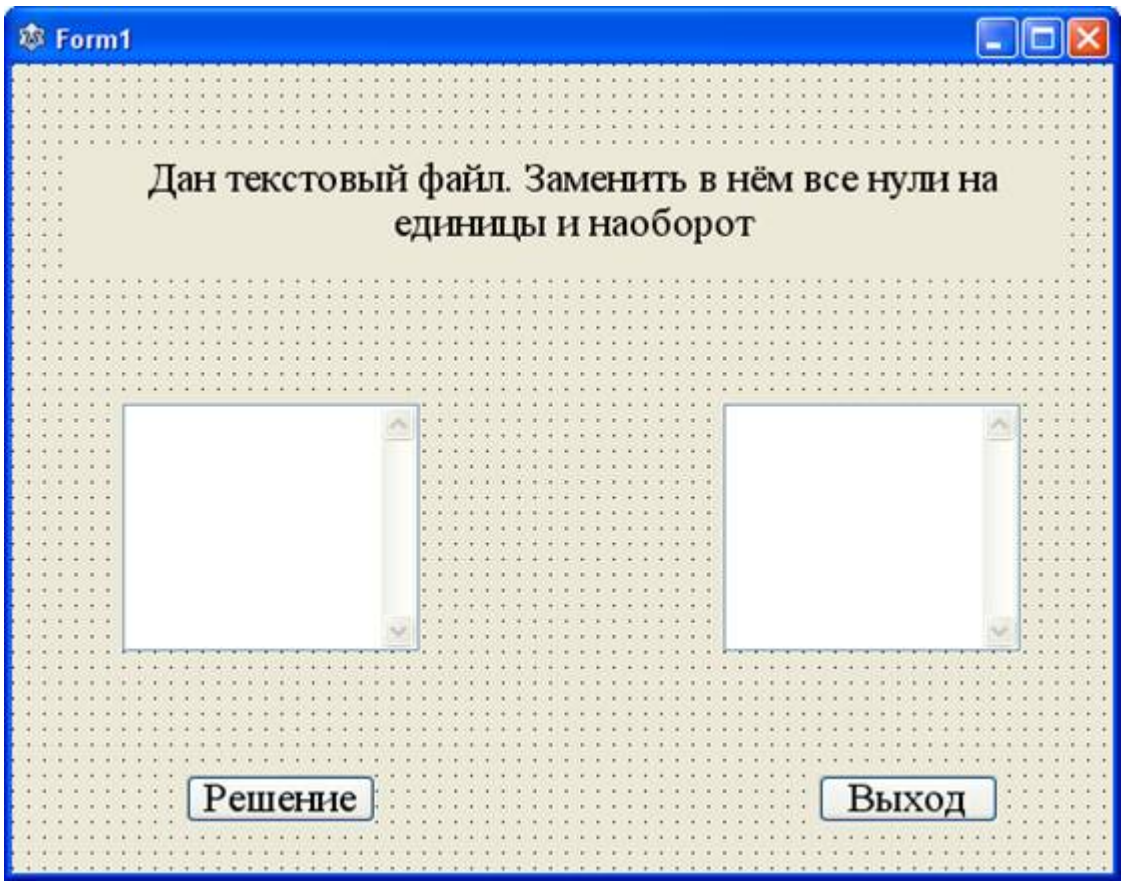

Рис. 1

Опишем процедуру Button1Click:

```
var s: string;
i, j: integer; //i – номер строки в Memo1, j – номер сим-вола в строке
begin
if OpenDialog1.Execute and FileExists (OpenDialog1.FileName) then
begin
Memo1.Lines.LoadFromFile(OpenDialog1.FileName);
Memo2.Clear;
for i:=0 to Memo1.Lines.Count-1 do
begin
s:=Memo1.Lines.Strings[i];
for j:=1 to length(s) do
if s[j]='1' then
s[j]:='0'
```

```
else
if s[j]='0' then
s[j]:='1';
Memo2.Lines.Add(s);
end; //конец цикла
end; //конец ветвления
if SaveDialog1.Execute then
Memo2.Lines.SaveToFile(SaveDialog1.FileName);
end; //конец процедуры
```

Осталось запрограммировать кнопку Выход.

Программа решения задачи готова.

2. Дан текстовый файл, каждая строка которого представляет число. Найти сумму чисел, сохранить результат в отдельном файле.

Разместим необходимые компоненты на форме. Нам понадобятся одна метка (Label1) для записи условия задачи, 2 кнопки (Button1 для подсчёта суммы чисел и Button2 для окончания работы программы), 2 многострочных текстовых окна (Memo1 для отображения файла и Memo2 для вывода суммы строк), диалоговое окно открытия файла (OpenDialog1), диалоговое окно сохранения файла (SaveDialog1).

В компоненте Label1 разместим условие задачи. Настроим компоненты Memo1 и Memo2. Для кнопки Button1 создадим надпись Решение, для Button2 – надпись Выход. Настроим компоненты OpenDialog1 и SaveDialog1.

Окончательный вид формы показан на рис. 2.

Рис. 2

Опишем процедуру Button1Click:

var s: **string**; //текущая строка файла

i, n, code, sum: integer; //i – номер строки в Memo1, code – номер ошибочного символа, sum – сумма чисел, n – вспомогательная переменная

begin

if OpenDialog1.Execute **and** FileExists (OpenDialog1.FileName) **then**

begin

Memo1.Lines.LoadFromFile(OpenDialog1.FileName);

Memo2.Clear;

sum:=0;

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

begin

s:=Memo1.Lines.Strings[i];

val(s, n, code);

if code=0 **then**

sum:=sum+n

else //если при преобразовании s в число произошла ошибка

MessageDlg('Строка должна представлять число', mtError, [mbOK], 0);

end; //конец цикла

Memo2.Lines.Add(IntToStr(sum));

end; //конец ветвления

if SaveDialog1.Execute **then**

Memo2.Lines.SaveToFile(SaveDialog1.FileName);

end; //конец процедуры

Осталось запрограммировать кнопку Выход.

Программа решения задачи готова.

3. Дан текстовый файл. Найти слово максимальной длины в каждой строке. Если в какой-то строке есть несколько самых длинных слов, то вывести последнее из них.

Разместим необходимые компоненты на форме. Нам понадобятся одна метка (Label1) для записи условия задачи, 2 кнопки (Button1 для поиска искомых слов и Button2 для окончания работы программы), 2 многострочных текстовых окна (Memo1 для отображения файла и Memo2 для вывода самых коротких слов), диалоговое окно открытия файла (OpenDialog1), диалоговое окно сохранения файла (SaveDialog1).

В компоненте Label1 разместим условие задачи. Настроим компоненты Memo1 и Memo2. Для кнопки Button1 создадим надпись Решение, для Button2 – надпись Выход. Настроим компоненты OpenDialog1 и SaveDialog1.

Окончательный вид формы показан на рис. 3.

Опишем процедуру Button1Click:

var s, s1, a: **string**; //s – текущая строка файла, s1 – текущее слово в строке, a – вспомогательная переменная

i, j, max: integer; //i – номер строки в Memo1, j – номер символа в строке, max – длина самого длинного слова

c: **set of** char; //множество знаков препинания

begin

if OpenDialog1.Execute **and** FileExists (OpenDialog1.FileName) **then**

begin

Memo1.Lines.LoadFromFile(OpenDialog1.FileName);

```
Memo2.Clear;
c:=['.', ',', '!', ':', ';', '?', '-', ' '];
for i:=0 to Memo1.Lines.Count-1 do
begin
s:=Memo1.Lines.Strings[i]+' ';
a:='';
```

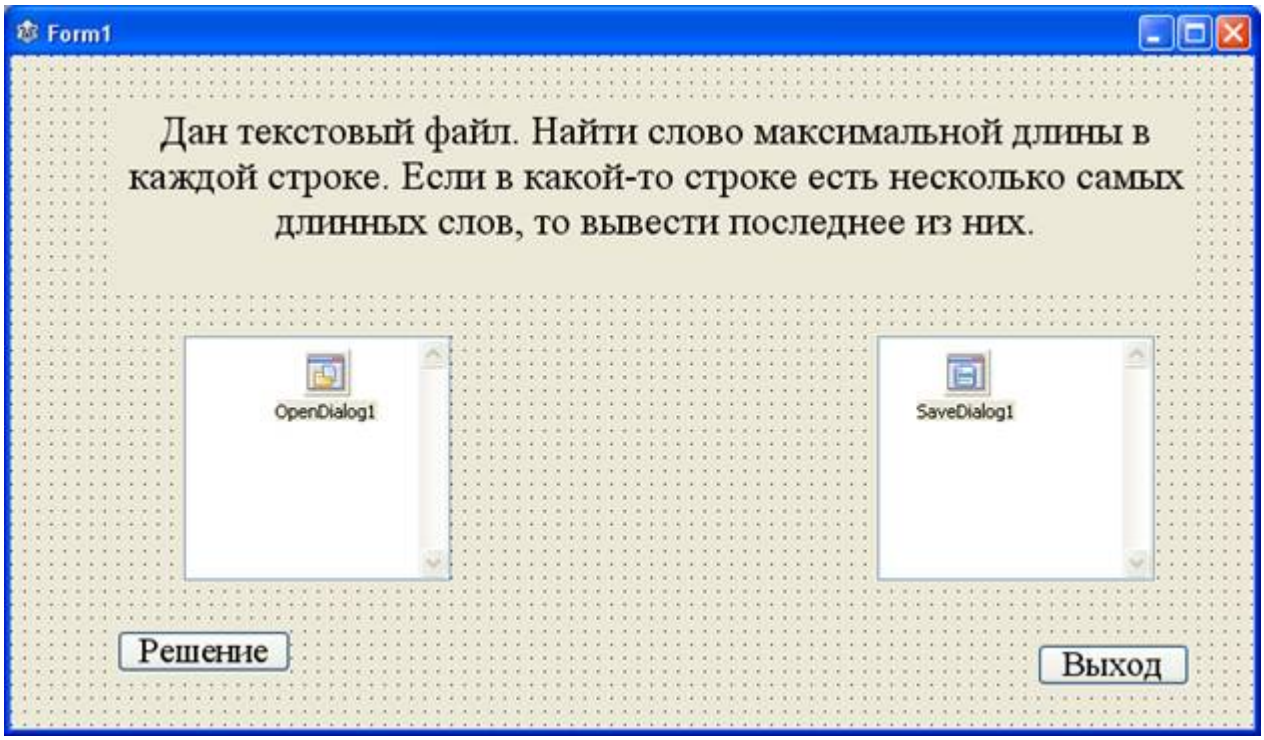

Рис. 3

```
s1:='';
max:=0;
for j:=1 to length(s) do
if not(s[j] in c) then
s1:=s1+s[j]
else //если s[j] – знак препинания
begin
if length(s1)>=max then
begin
max:=length(s1);
```

```
a:=s1;
end; //конец ветвления
s1:='';
end; //конец ветвления
Memo2.Lines.Add(a);
end; //конец цикла
end; //конец ветвления
if SaveDialog1.Execute then
Memo2.Lines.SaveToFile(SaveDialog1.FileName);
end; //конец процедуры
```

Осталось запрограммировать кнопку Выход.

Программа решения задачи готова.