

**С. В. Журавлев**

**МЕТОДИКА ПРОВЕДЕНИЯ ЗАНЯТИЙ ПО ДИСЦИПЛИНЕ  
«ПРАКТИКУМ ПО РЕШЕНИЮ ЗАДАЧ НА ЭВМ» НА ТЕМУ  
«СОРТИРОВКА» НА ОСНОВЕ КОМПЕТЕНТНОСТНОГО,  
ИНФОРМАЦИОННОГО И ДЕЯТЕЛЬНОСТНОГО ПОДХОДОВ (ЭТАП  
ПРИБЛИЖЕНИЯ)**

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 сортировки массивов и строк, учатся анализировать основные методы сортировки и определения максимальной длины упорядоченных элементов в произвольной последовательности, 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: дополнительную информацию о методах сортировки, способы нахождения максимальной длины упорядоченных элементов в произвольной последовательности.

Все задачи по данной теме подразделяются на 3 группы:

I. Задачи для ситуации полной информационной определённости:

1. Дан одномерный числовой массив. Расположить вначале нечётные элементы по возрастанию, а затем чётные по убыванию.
2. Дан одномерный числовой массив. Расположить вначале элементы с нечётными индексами по убыванию, а затем элементы с чётными индексами по возрастанию.
3. Дан двумерный числовой массив. Отсортировать столбец, номер которого запрашивается с клавиатуры, по убыванию.
4. Дан двумерный числовой массив. Отсортировать строку, номер которой запрашивается с клавиатуры, по возрастанию.

5. Дан двумерный числовой массив. Расположить элементы в строках с нечётными индексами по возрастанию, а в строках с чётными индексами — по убыванию.

6. Дан двумерный числовой массив. Упорядочить элементы столбцов, которые начинаются с минимального элемента в первой строке, по возрастанию, а которые начинаются с максимального элемента — по убыванию.

7. Дана последовательность строчных латинских букв. Отсортировать эту последовательность в алфавитном порядке.

8. Дан одномерный числовой массив. Расположить вначале отрицательные элементы по возрастанию, а затем положительные по убыванию.

9. Дан двумерный числовой массив. Расположить элементы в столбцах с чётными индексами по возрастанию, а в столбцах с нечётными — по убыванию.

10. Дана последовательность латинских букв. Расположить вначале прописные буквы в алфавитном порядке, а затем строчные в порядке, противоположном алфавитному.

II. Задачи для ситуации частичной информационной неопределённости:

11. Дан одномерный числовой массив, представляющий собой невозрастающую последовательность. Преобразовать массив таким образом, чтобы он превратился в убывающую последовательность.

12. Дан одномерный числовой массив из 50 элементов. Определить число элементов самой длинной упорядоченной по убыванию последовательности.

13. Дана квадратная матрица. Упорядочить элементы, расположенные на главной диагонали, по возрастанию.

14. Дана квадратная матрица. Упорядочить элементы, расположенные на побочной диагонали, по убыванию.

15. Дан одномерный числовой массив. Отсортировать отрицательные элементы в порядке убывания, а неотрицательные элементы оставить без изменения.

16. Дана последовательность из 25 строчных латинских букв. Определить число букв самой длинной упорядоченной по возрастанию последовательности.

17. Даны 2 числовых массива размерности  $N$ , упорядоченных по возрастанию. Сформировать из них третий массив так, чтобы он также был упорядочен по возрастанию.

18. Дана последовательность строчных латинских букв, представляющая собой неубывающую последовательность. Преобразовать строку таким образом, чтобы она превратилась в возрастающую последовательность.

19. Дана последовательность латинских букв. Отсортировать строчные буквы в алфавитном порядке, а прописные буквы оставить без изменения.

20. Дан одномерный числовой массив, упорядоченный по убыванию и число  $s$ . Вставить число в массив, сохраняя порядок.

III. Задачи для ситуации полной информационной неопределённости:

21. Дан список слов. Расставить слова в алфавитном порядке.

22. Дан список слов. Упорядочить слова по их длине.

23. Дан список слов. Расположить вначале слова с нечётным числом букв, упорядочив их по длине в порядке убывания, а затем слова с чётным числом букв в порядке, противоположном алфавитному.

24. Дан список слов. Определить максимальное число идущих подряд слов, расположенных в алфавитном порядке.

25. Дан список слов. Определить максимальное число идущих подряд слов, расположенных в порядке уменьшения их длин.

26. Дан двумерный массив строкового типа. Расположить слова в строках с нечётными индексами в порядке, противоположном алфавитному, а в строках с чётными индексами — в порядке увеличения их длин.

27. Дана квадратная матрица строкового типа. Отсортировать слова, расположенные на главной диагонали, в алфавитном порядке.

28. Дана квадратная матрица строкового типа. Отсортировать слова, расположенные на побочной диагонали, в порядке уменьшения их длин.

29. Дан список слов, расположенных в порядке увеличения длин и некоторое слово. Вставить слово в список, сохраняя порядок.

30. Даны 2 списка, состоящих из  $N$  слов, расположенных в алфавитном порядке. Сформировать из них третий список так, чтобы он также состоял из слов, расположенных в алфавитном порядке.

Теперь рассмотрим пример подробного решения трёх задач из первой группы.

1. Дан одномерный числовой массив. Отсортировать его по возрастанию.

Разместим необходимые компоненты на форме. Нам понадобятся одна метка (Label1) для записи условия задачи, 2 кнопки (Button1 для осуществления сортировки массива и Button2 для окончания работы программы), одна таблица строк (StringGrid1) для вывода исходного и отсортированного массива.

В компоненте Label1 разместим условие задачи. Для кнопки Button1 создадим надпись Решение, для Button2 – надпись Выход. Настроим компонент StringGrid1.

Окончательный вид формы показан на рис. 1.

Опишем процедуру FormCreate:

**begin**

randomize;

**end;** //конец процедуры

Теперь опишем процедуру Button1Click:

**var** A: **array of** real; //динамический одномерный массив вещественного типа

i, N: integer; //i – индекс элемента массива A, N – раз-мерность массива A

x: real; //вспомогательная переменная

**begin**

N:=StrToInt(InputBox('Размерность массива','Введите количество элементов массива','5'));

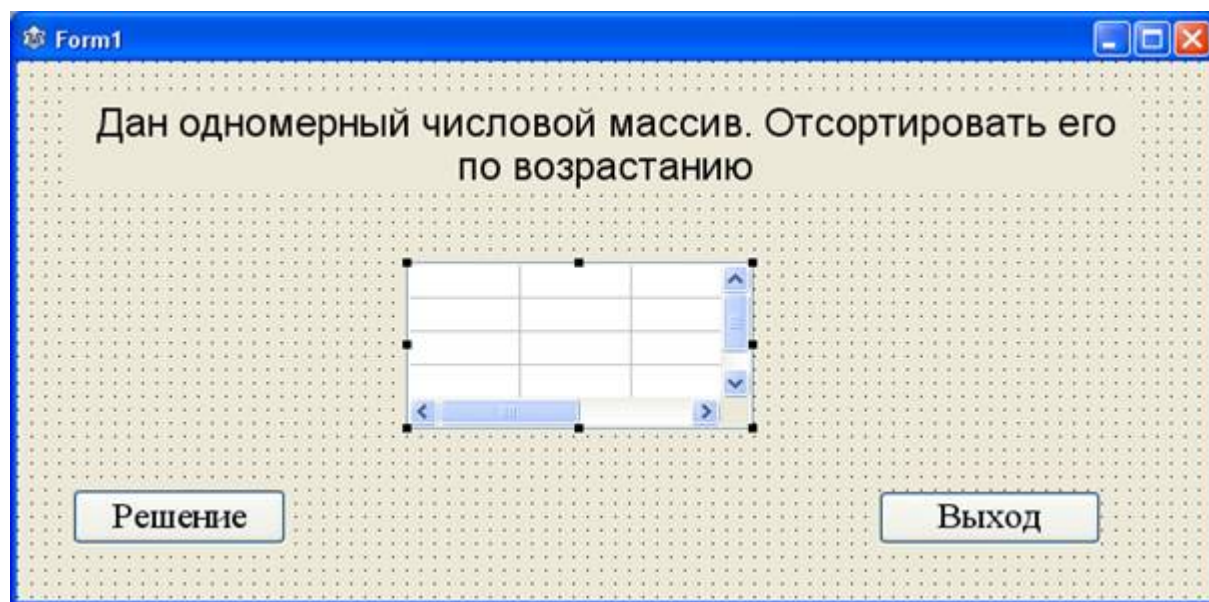


Рис. 1

**if** N>0 **then**

**begin**

SetLength(A, N);

StringGrid1.ColWidths[0]:=120;

```
StringGrid1.ColCount:=n+1;
StringGrid1.RowCount:=3;
StringGrid1.Cells[0, 0]:='Номер элемента';
StringGrid1.Cells[0, 1]:='Исходный массив';
StringGrid1.Cells[0, 2]:='Отсортированный массив';
for i:=low(A) to high(A) do
begin
  A[i]:=random*100-50; //заполняем массив случайными числами из отрезка
  [-50, 49]
  StringGrid1.Cells[i+1, 0]:=IntToStr(i+1);
  StringGrid1.Cells[i+1, 1]:=FloatToStrF(A[i], ffFixed, 5, 2);
end; //конец цикла
for i:=low(A) to high(A)-1 do
for j:=i+1 to high(A) do
if A[i]>A[j] then
begin
  x:=A[i];
  A[i]:=A[j];
  A[j]:=x;
end; //конец ветвления
for i:=low(A) to high(A) do
  StringGrid1.Cells[i+1, 2]:=FloatToStrF(A[i], ffFixed, 5, 2);
  Finalize(A);
end //конец ветвления
else //если N неположительно
  MessageDlg('Введено не натуральное число', mtError, [mbOK], 0);
end; //конец процедуры
Осталось запрограммировать кнопку Выход.
Программа решения задачи готова.
```

2. Дан двумерный числовой массив. Отсортировать элементы каждой строки по убыванию.

Разместим необходимые компоненты на форме. Нам понадобятся одна метка (Label1) для записи условия задачи, 2 кнопки (Button1 для осуществления сортировки массива и Button2 для окончания работы программы), 2 таблицы строк (StringGrid1 для вывода элементов исходного массива и StringGrid2 для вывода элементов отсортированного массива).

В компоненте Label1 разместим условие задачи. Для кнопки Button1 создадим надпись Решение, для Button2 – надпись Выход. Настроим компоненты StringGrid1 и StringGrid2.

Окончательный вид формы показан на рис. 2.

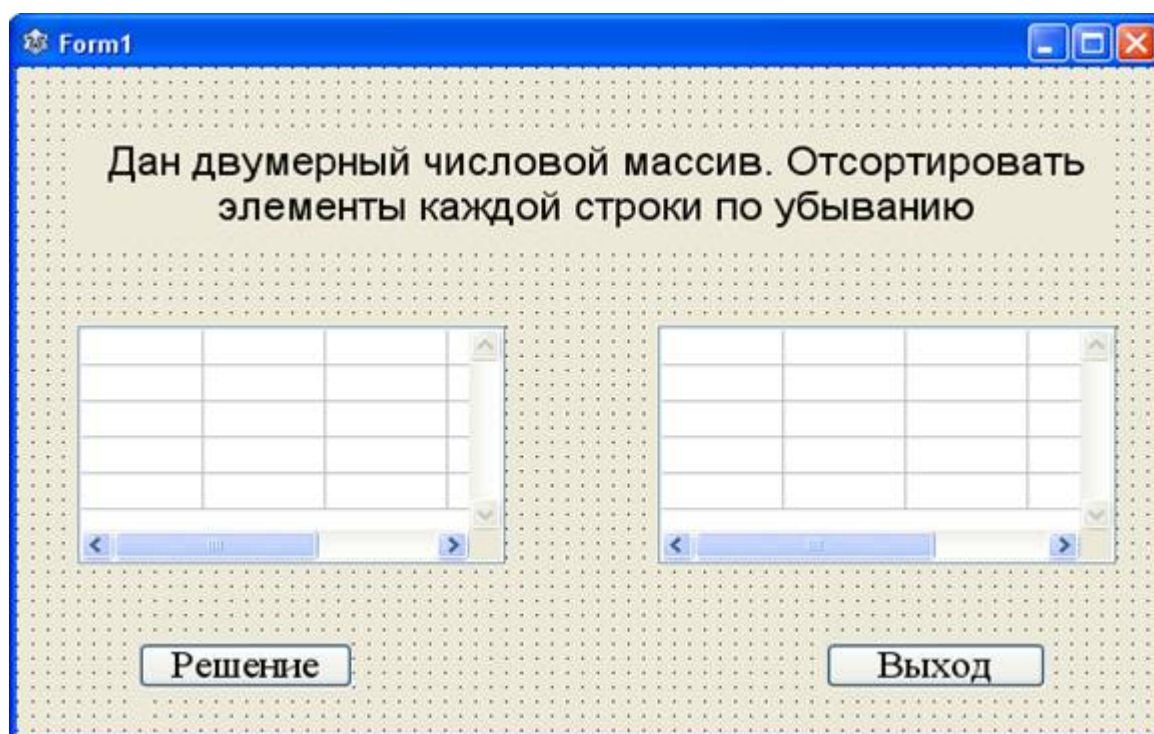


Рис. 2

Опишем процедуру FormCreate:

**begin**

randomize;

**end;** //конец процедуры

Теперь опишем процедуру Button1Click:

**var A: array of array of real;** //динамический двумерный массив вещественного типа

N, M, i, j, k, num: integer; //N – количество строк, M – количество столбцов, i – индекс строки, j – индекс столбца, k, num – вспомогательные переменные

max: real; //вспомогательная переменная

**begin**

N:=StrToInt(InputBox('Массив','Введите количество строк','2'));

M:=StrToInt(InputBox('Массив','Введите количество столбцов','2'));

**if** (N>1) **and** (M>1) **then**

**begin**

SetLength(A, N, M);

StringGrid1.ColCount:=M;

StringGrid1.RowCount:=N;

StringGrid2.ColCount:=M;

StringGrid2.RowCount:=N;

**for** i:=0 **to** N-1 **do**

**for** j:=0 **to** M-1 **do**

**begin**

A[i, j]:=random\*100-50; //заполняем массив случайными числами из отрезка [-50, 49]

StringGrid1.Cells[j, i]:=FloatToStrF(A[i, j], ffFixed, 5, 2);

**end;** //конец цикла

**for** i:=0 **to** N-1 **do**

**for** j:=0 **to** M-2 **do**

**begin**

max:=A[i, j];

**for** k:=j+1 **to** M-1 **do**

**if** A[i, k]>max **then**

**begin**

max:=A[i, k];

num:=k;

A[i, num]:=A[i, j];

A[i, j]:=max;

```
end; //конец ветвления
```

```
end; //конец цикла
```

```
for i:=0 to N-1 do
```

```
for j:=0 to M-1 do
```

```
StringGrid2.Cells[j, i]:=FloatToStrF(A[i, j], ffFixed, 5, 2);
```

```
Finalize(A);
```

```
end //конец ветвления
```

```
else //если  $N \leq 1$  или  $M \leq 1$ 
```

```
MessageDlg('Размерность массива недостаточна для наличия матрицы',  
mtError, [mbOK], 0);
```

```
end; //конец процедуры
```

Осталось запрограммировать кнопку Выход.

Программа решения задачи готова.

3. Дана последовательность прописных латинских букв. Отсортировать эту последовательность в порядке, противоположном алфавитному.

Разместим необходимые компоненты на форме. Нам понадобятся одна метка (Label1) для записи условия задачи, 2 текстовых окна (Edit1 для ввода исходной строки и Edit2 для вывода отсортированной строки), 2 кнопки (Button1 для осуществления сортировки строки и Button2 для окончания работы программы).

В компоненте Label1 разместим условие задачи. Очистим компоненты Edit1 и Edit2. Для кнопки Button1 создадим надпись Решение, для Button2 – надпись Выход.

Окончательный вид формы показан на рис. 3.

Опишем процедуру Button1Click:

```
var s, s1: string; //s – исходная строка, s1 – отсортированная строка
```

```
i, j, k, m, l: integer; //i – номер символа в строке, k – количество символов  
исходной строки, больших по алфавиту первого символа, m – количество  
символов исходной строки, совпадающих с первым символом, j, l –  
вспомогательные переменные
```

```
begin
```

```
Edit2.Text:="";
```

```
s:=Edit1.Text;
```



```
s1:=Edit1.Text;  
for i:=1 to length(s) do  
begin  
k:=0;  
m:=0;  
for j:=1 to length(s) do
```

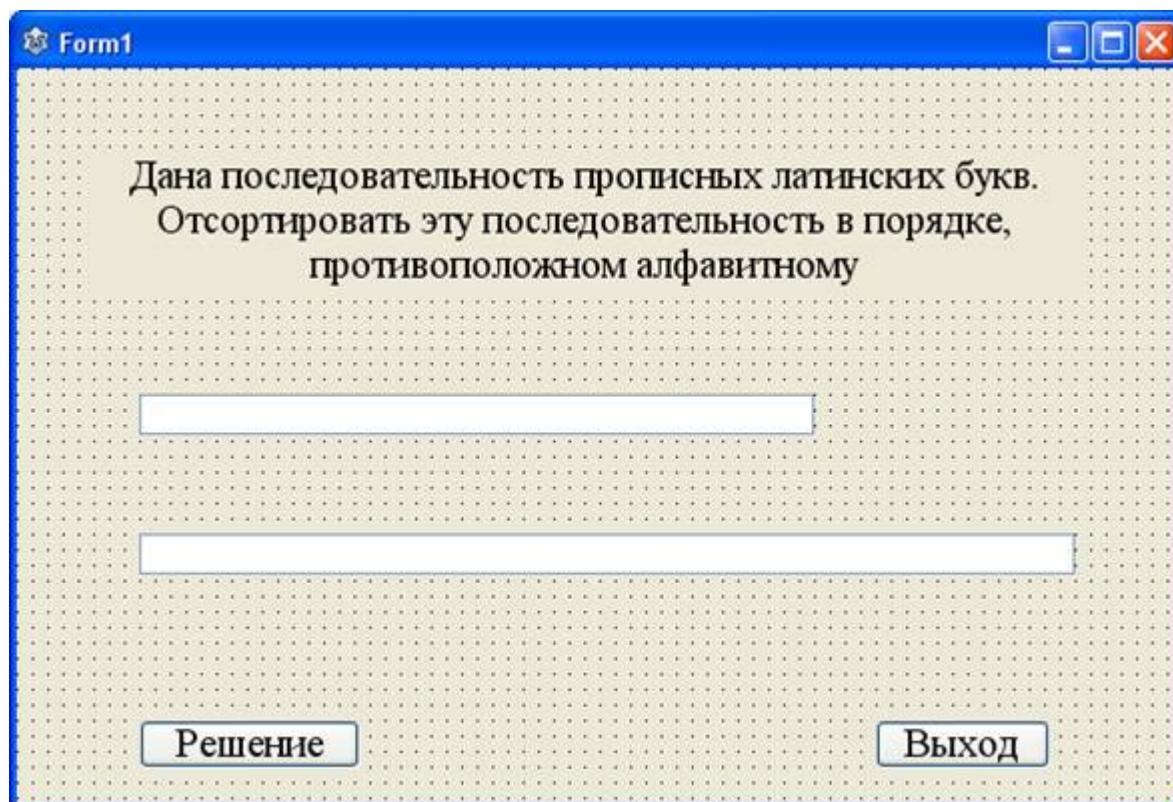


Рис. 3

```
begin  
if s[i]<s[j]>then</s[j]>  
k:=k+1;  
if s[i]=s[j] then  
m:=m+1;  
end; //конец цикла  
for l:=1 to m do  
begin  
delete(s1, k+l, 1);
```

```
insert(s[i], s1, k+1);
```

```
end; //конец цикла
```

```
end; //конец цикла
```

```
Edit3.Text:=s1;
```

```
end; //конец процедуры
```

Осталось запрограммировать кнопку Выход.

Программа решения задачи готова.