

1. Цель работы

Целью работы является изучение файловых операций ввода-вывода.

2. Задание

Согласно варианту №1, требуется написать программу, которая считывает из текстового файла три предложения и записывает их в другой файл в обратном порядке.

3. Описание созданных функций

Для реализации задания были использованы следующие функции:

Имя: `is_znak`.

Назначение: проверка входного символа на принадлежность к `.!?`;

Входные данные: `char c`;

Выходные данные:

- принадлежит/не принадлежит поступивший символ знаку

Побочный эффект: отсутствует.

Прототип: `bool is_znak(char c);`

Алгоритм:

- псевдокод

Взять поступивший символ и сравнить его с заданным

Имя: `is_space`.

Назначение: проверка поступившего знака на принадлежность к управляющим знакам.

Входные данные:

- `char c`.

Выходные данные:

- принадлежит/не принадлежит поступивший символ управляющему знаку

Побочный эффект: отсутствует.

Прототип: bool is_space(char c);

Алгоритм:

- псевдокод

Взять поступивший символ и сравнить его с заданным

Имя: readSentence.

Назначение: считывание символов до заданного знака.

Входные данные:

char *s - символ поступивший из файла

- s – переменная типа char.

Выходные данные:

Побочный эффект: отсутствует.

Прототип: readSentence(char *s);

Алгоритм:

- псевдокод

Записывать в переменную s знаки до тех пор, пока не будет достигнут знак проверки.

Имя: ReadFile.

Назначение: считывание символов из файла.

Входные данные:

char *s1 - массив символов, хранящий в себе первое предложение;

char *r1 - массив символов, хранящий в себе разделители;

char *s2- массив символов, хранящий в себе второе предложение;

char *r2 - массив символов, хранящий в себе разделители;

char *s3 - массив символов, хранящий в себе третье предложение.

Выходные данные:

Побочный эффект: отсутствует.

Прототип: void ReadFile(char *s1, char *r1, char *s2, char *r2, char *s3);

Алгоритм:

- псевдокод

Открыть файл;

Если он не открыт, то вывести ошибку;

Прочитать первое предложение в первый массив символов s1 пока не встретим управляющий символ;

Управляющий символ записать в разделители r1;

Повторяем действия ещё 2 раза;

4. Листинг программы

4.1. С использованием ifstream

```
1. #include <iostream>
2. #include <fstream>
3. #include <windows.h>

4. using namespace std;

5. ofstream fout;
6. ifstream fin;

7. bool is_znak(char c)
8. {
9.     if (c == '.' || c == '?' || c == '!')
10.    return true;
11.    return false;
12. }

13. bool is_space(char c)
14. {
15.     if (c == ' ' || c == '\t' || c == '\n')
16.    return true;
17.    return false;
18. }

19. // Считываем предложение до знака .?! или до конца файла
20. void readSentence(char *s)
21. {
22.     int i = 0;
23.     while (1)
24.     {
25.         s[i] = fin.get();
26.         if (is_znak(s[i]))
27.             break;

28.         if (fin.eof())
29.         {
30.             s[i] = '\0';
31.             break;
32.         }

33.         i++;
34.     }
35.     s[i + 1] = '\0';

36. }

37. // s1, s2, s3 - три предложения,
38. // r1, r2 - разделители между ними (группы пробельных символов)
39. void ReadFile(char *s1, char *r1, char *s2, char *r2, char *s3)
40. {
41.     const char *name = "text.txt";
42.     fin.open(name);

43.     if (!fin.is_open())
44.     {
45.         cout << "Ошибка открытия файла!" << endl;
46.         return;
47.     }

48.     readSentence(s1);

49.     int i = 0;
```

```

50. char c = fin.get();
51. while (is_space(c)) // пока пробельный символ
52. {
53. r1[i++] = c; // заносим в строку разделитель
54. c = fin.get();
55. }
56. r1[i] = '\0';

57. s2[0] = c;
58. readSentence(s2 + 1);

59. i = 0;
60. c = fin.get();
61. while (is_space(c)) // пока пробельный символ
62. {
63. r2[i++] = c; // заносим в строку разделитель
64. c = fin.get();
65. }
66. r2[i] = '\0';

67. s3[0] = c;
68. readSentence(s3 + 1);

69. fin.close();
70. }

71. void Output(char *s1, char *r1, char *s2, char *r2, char *s3)
72. {
73. fout.open("newtext.txt");

74. fout << s3;
75. if (*r1)
76. fout << r1;
77. fout << s2;

78. if (*r2)
79. fout << r2;

80. fout << s1;
81. fout << endl;

82. fout.close();
83. }

84. int main()
85. {
86. int MAX_SIZE = 300;
87. char *s1 = new char[MAX_SIZE];
88. char *s2 = new char[MAX_SIZE];
89. char *s3 = new char[MAX_SIZE];
90. char *r1 = new char[MAX_SIZE];
91. char *r2 = new char[MAX_SIZE];

92. ReadFile(s1, r1, s2, r2, s3);

93. Output(s1, r1, s2, r2, s3);

94. delete[] s1;
95. delete[] s2;
96. delete[] s3;
97. delete[] r1;
98. delete[] r2;

99. _CrtSetReportMode(_CRT_WARN, _CRTDBG_MODE_FILE);

```

```
100.     _CrtSetReportFile(_CRT_WARN, _CRTDBG_FILE_STDOUT);
101.     _CrtSetReportMode(_CRT_ERROR, _CRTDBG_MODE_FILE);
102.     _CrtSetReportFile(_CRT_ERROR, _CRTDBG_FILE_STDOUT);
103.     _CrtSetReportMode(_CRT_ASSERT, _CRTDBG_MODE_FILE);
104.     _CrtSetReportFile(_CRT_ASSERT, _CRTDBG_FILE_STDOUT);
105.     _CrtDumpMemoryLeaks();

106.     system("pause");
107.     return 0;
108. }
```

4.1. С использованием cstdio

```
1. #define _CRT_SECURE_NO_WARNINGS
2. #include <iostream>
3. #include <cstdio>
4. #include <fstream>
5. #include<windows.h>

6. using namespace std;

7. bool is_znak(char c)
8. {
9.     if (c == '.' || c == '?' || c == '!')
10.    return true;
11.    return false;
12. }

13. bool is_space(char c)
14. {
15.     if (c == ' ' || c == '\t' || c == '\n')
16.    return true;
17.    return false;
18. }

19. // Считываем предложение до знака .?! или до конца файла
20. void readSentence(FILE *fin, char *s)
21. {
22.     int i = 0;
23.     while (1)
24.     {
25.         s[i] = fgetc(fin);
26.         if (is_znak(s[i]))
27.             break;

28.         if (feof(fin))
29.         {
30.             s[i] = '\0';
31.             break;
32.         }

33.         i++;
34.     }
35.     s[i + 1] = '\0';

36. }

37. // s1, s2, s3 - три предложения,
38. // r1, r2 - разделители между ними (группы пробельных символов)
39. void ReadFile(char *s1, char *r1, char *s2, char *r2, char *s3)
40. {
41.     const char *name = "text.txt";
42.     FILE *fin = fopen(name, "r");

43.     if (!fin)
44.     {
45.         cout << "Ошибка открытия файла!" << endl;
46.         return;
47.     }

48.     readSentence(fin, s1);

49.     int i = 0;
50.     char c = fgetc(fin);
51.     while (is_space(c)) // пока пробельный символ
```

```

52. {
53. r1[i++] = c;    // заносим в строку разделитель
54. c = fgetc(fin);
55. }
56. r1[i] = '\0';

57. s2[0] = c;
58. readSentence(fin, s2 + 1);

59. i = 0;
60. c = fgetc(fin);
61. while (is_space(c)) // пока пробельный символ
62. {
63. r2[i++] = c;    // заносим в строку разделитель
64. c = fgetc(fin);
65. }
66. r2[i] = '\0';

67. s3[0] = c;
68. readSentence(fin, s3 + 1);

69. fclose(fin);
70. }

71. void Output(char *s1, char *r1, char *s2, char *r2, char *s3)
72. {
73. FILE* fout = fopen("newtext.txt", "w");

74. fprintf(fout, "%s", s3);

75. if (*r1)
76. fprintf(fout, "%s", r1);
77. fprintf(fout, "%s", s2);

78. if (*r2)
79. fprintf(fout, "%s", r2);

80. fprintf(fout, "%s\n", s1);

81. fclose(fout);
82. }

83. int main()
84. {
85. setlocale(LC_ALL, "Russian");
86. SetConsoleCP(1251);
87. SetConsoleOutputCP(1251);

88. int MAX_SIZE = 300;
89. char *s1 = new char[MAX_SIZE];
90. char *s2 = new char[MAX_SIZE];
91. char *s3 = new char[MAX_SIZE];
92. char *r1 = new char[MAX_SIZE];
93. char *r2 = new char[MAX_SIZE];

94. ReadFile(s1, r1, s2, r2, s3);

95. Output(s1, r1, s2, r2, s3);

96. delete[] s1;
97. delete[] s2;
98. delete[] s3;
99. delete[] r1;
100. delete[] r2;

```



```
101.     _CrtSetReportMode(_CRT_WARN, _CRTDBG_MODE_FILE);
102.     _CrtSetReportFile(_CRT_WARN, _CRTDBG_FILE_STDOUT);
103.     _CrtSetReportMode(_CRT_ERROR, _CRTDBG_MODE_FILE);
104.     _CrtSetReportFile(_CRT_ERROR, _CRTDBG_FILE_STDOUT);
105.     _CrtSetReportMode(_CRT_ASSERT, _CRTDBG_MODE_FILE);
106.     _CrtSetReportFile(_CRT_ASSERT, _CRTDBG_FILE_STDOUT);
107.     _CrtDumpMemoryLeaks();

108.     return 0;
109. }
```

5. Пример выполнения программы

Ниже приведён пример выполнения программы

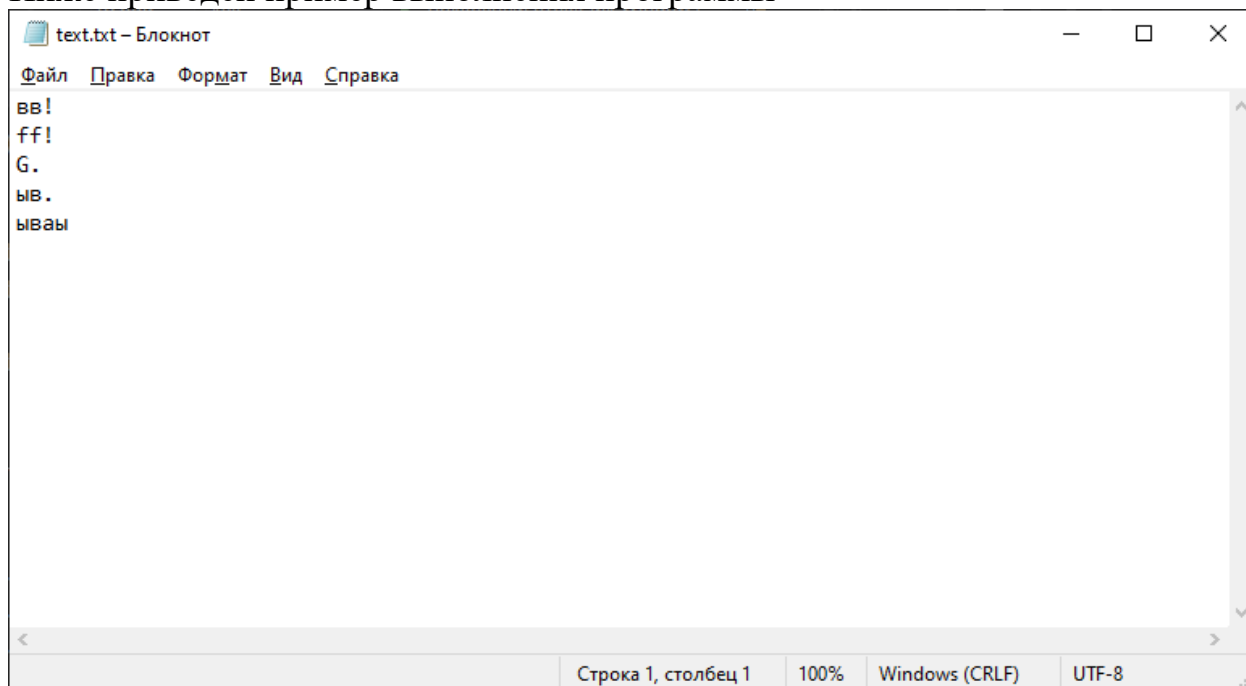


Рис 1. Пример выполнения программы fstream

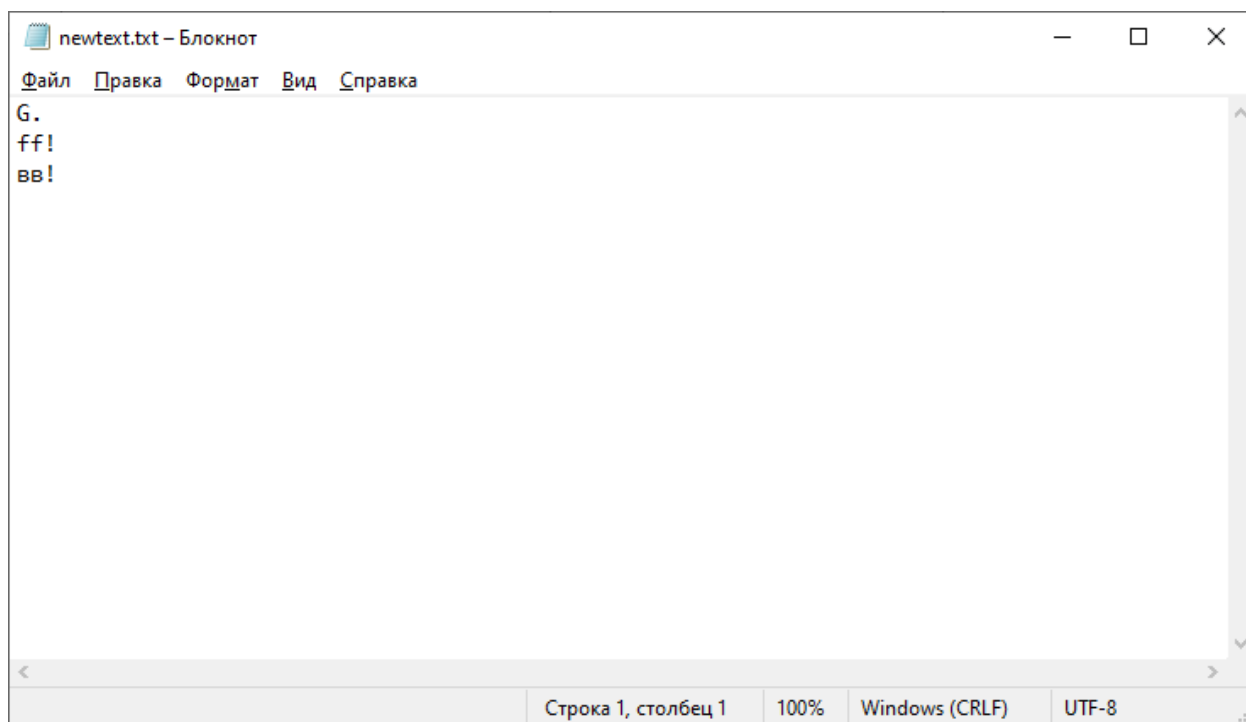


Рис 2. Пример выполнения программы fstream

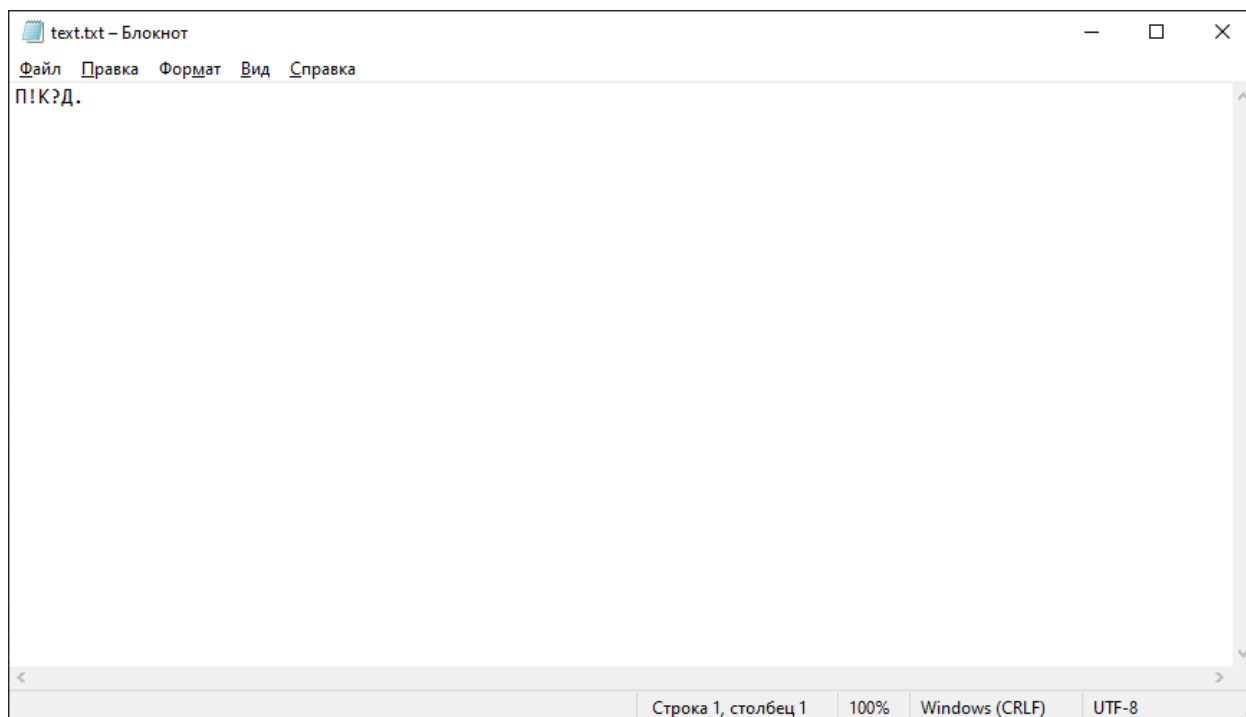


Рис 3. Пример выполнения программы cstdio

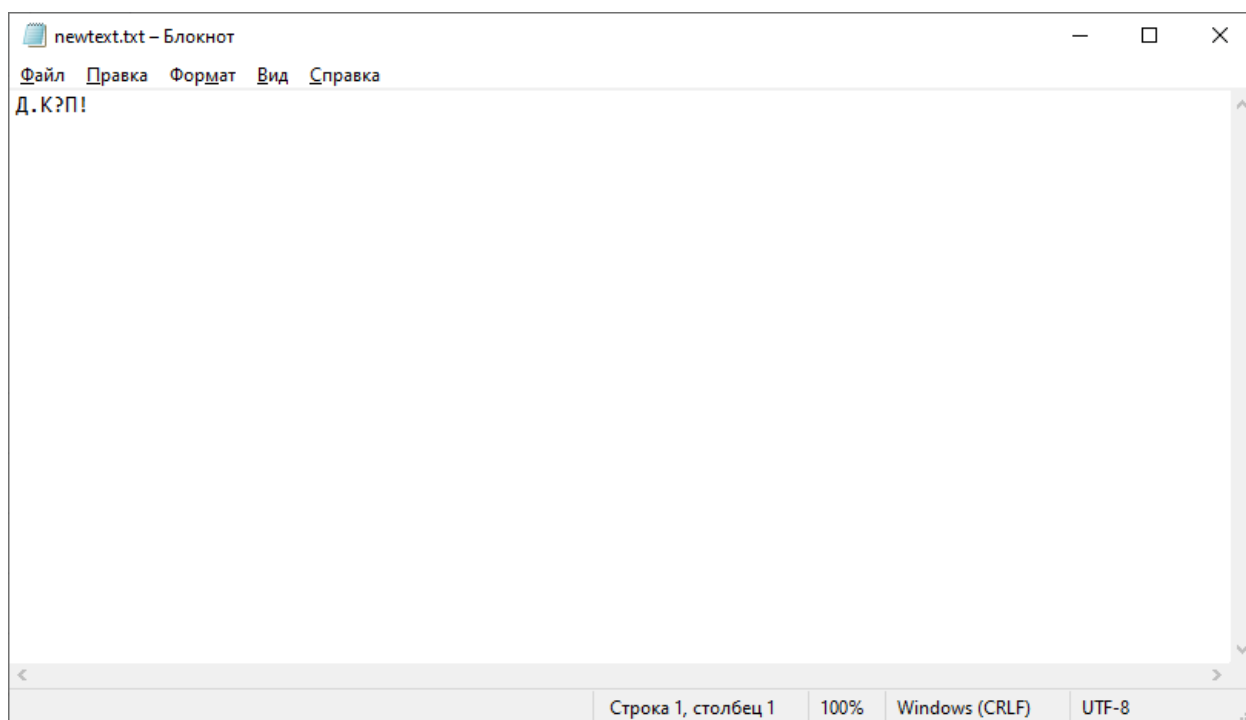


Рис 4. Пример выполнения программы cstdio