

Занятие 5

Базовый файловый ввод-вывод.

Пример 1. Запись в файл.

Помимо вывода информации на экран можно также её записывать и в файл. Сначала необходимо подключить заголовочный файл `fstream`, в котором декларированы операции с файловыми потоками. Для записи в файл вводится соответствующий поток вывода `ofstream` (output file stream) и ему дается имя, см. строку 25 в примере. Затем этот поток открывается и при этом задается имя файла, который будет создан, см. строку 26. Затем информация выводится в файл также, как и на экран, но вместо стандартного потока вывода `cout` используется открытый файловый поток, см. строки 27-30. После завершения вывода поток необходимо закрыть, см. строку 31.

```
1 #include <iostream>
2 #include <string>
3 #include <fstream> // для файлового ввода-вывода
4
5 int main(void) {
6     using namespace std;
7     string automobile;
8     int year;
9     double a_price, d_price;
10    cout << "Enter the make and model of automobile: ";
11    getline(cin, automobile);
12    cout << "Enter the model year: ";
13    cin >> year;
14    cout << "Enter the original asking price: ";
15    cin >> a_price;
16    d_price = 0.913 * a_price;
17
18    // Вывод в поток cout
19    cout << "Make and model: " << automobile << endl;
20    cout << "Year: " << year << endl;
21    cout << "Was asking $" << a_price << endl;
22    cout << "Now asking $" << d_price << endl;
23
24    // Вывод той же информации в файл вместо cout
25    ofstream outFile1; // имя потока вывода
26    outFile1.open("carinfo.txt"); // открыть поток с данным именем файла
27    outFile1 << "Make and model: " << automobile << endl;
28    outFile1 << "Year: " << year << endl;
29    outFile1 << "Was asking $" << a_price << endl;
30    outFile1 << "Now asking $" << d_price << endl;
31    outFile1.close(); // завершить работу с файлом
32    return 0;
33 }
```

Пример 2. Чтение файла.

Для чтения из файла сначала открывается поток ввода `ifstream` (input file stream), используя имя требуемого файла, см. строку 13. После попытки открыть файл, можно проверить состояние потока ввода, строка 14. Если файл открыть не удалось (например, если такого файла нет), то операция `.is_open()` возвращает 0 (или false). Если поток открыт,

то можно осуществлять чтение информации, так же как при вводе с клавиатуры, но вместо стандартного потока ввода `cin` использовать открытый файловый поток. При чтении можно осуществлять проверку считывания с помощью операции `.good`, см. строку 23. Если считывание информации прервалось, то можно проверить причину этого с помощью команд `.eof()` (достигнут конец файла, end of file) или `.fail()` (несоответствие типа данных и типа переменной).

Создайте текстовый файл, который содержит числа, например,

```
12.2 23.4 56.6 56.99
12.34 12.1 22.9
```

Используйте созданный файл для чтения в следующей программе.

```
1 #include <iostream>
2 #include <string>
3 #include <fstream> // поддержка файлового ввода-вывода
4 #include <cstdlib> // поддержка exit()
5
6 int main(void) {
7     using namespace std;
8     string filename;
9     ifstream inFile; // объект для обработки файлового ввода
10
11     cout << "Enter name of data file: "; // запрос имени файла данных
12     getline(cin, filename);             // чтение имени файла
13     inFile.open(filename);              // ассоциирование inFile с файлом
14     if (!inFile.is_open()){ // если не удалось открыть файл
15         cout << "Could not open the file " << filename << endl;
16         cout << "Program terminating.\n";
17         exit(EXIT_FAILURE); // завершение программы с ошибкой
18     }
19
20     double value, sum = 0.0;
21     int count = 0; // количество прочитанных элементов
22     inFile >> value; // ввод первого значения
23     while (inFile.good()){ // пока ввод успешен и не достигнут EOF
24         ++count; // еще один элемент прочитан
25         sum += value; // вычисление текущей суммы
26         inFile >> value; // ввод следующего значения
27     }
28
29     if (inFile.eof()) // если достигнут конец файла
30         cout << "End of file reached.\n";
31     else if (inFile.fail()) // или несоответствие типа данных
32         cout << "Input terminated by data mismatch.\n";
33     else // иначе неизвестная причина
34         cout << "Input terminated for unknown reason.\n";
35
36     if (count == 0) // данные для обработки отсутствуют
37         cout << "No data processed.\n";
38     else {
39         cout << "Items read:" << count << endl;
40         cout << "Sum: " << sum << endl;
41         cout << "Average: " << sum / count << endl;
42     }
43     inFile.close(); // закрыть файл
```

```
44 |
45 |     return 0;
46 | }
```

Задания

Задание 5.1. Напишите программу, которая подсчитывает количество заданных символов в файле. Программа должна запрашивать у пользователя имя файла и требуемый символ. Результат подсчета должен выводиться на экран. Используйте файл `text_v1.txt` из папки с заданиями.