Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

Лабораторная работа 3

По дисциплине «Структуры данных»

На тему «Двоичные и текстовые файлы»

Выполнила:

Студентка 1 курса 2 группы

Глухова Д.В.

Преподаватель: Север А.С.

2023, Минск

Файл – именованный набор байтов, который может быть сохранен на некотором накопителе.

В одной директории не могут находиться файлы с одинаковыми именами. Под именем файла понимается не только его название, но и расширение, например: file.txt и file.dat**-**разные файлы.

Полное имя файлов – это полный адрес к директории файла с указанием имени файла.

Файлы:

Текстовые. Текстовый поток – последовательность символов. Организуются по строкам, каждая заканчивается символом «конца строки». Конец файла - символом «конца файла». При записи информации в текстовый файл, просмотреть который можно с помощью любого текстового редактора, все данные преобразуются к символьному типу и хранятся в символьном виде. При передаче символов из потока на экран, часть из них не выводится (например, символ возврата каретки, перевода строки). В текстовом режиме каждый разделительный символ строки автоматически преобразуется в пару (возврат каретки – переход на новую строку).

Двоичные. Двоичный поток – это последовательность байтов, которые однозначно соответствуют тому, что находится на внешнем устройстве. В двоичных файлах информация считывается и записывается в виде блоков определенного размера, в которых могут храниться данные любого вида и структуры.

Открытый файл представляется как последовательность считываемых или записываемых данных. При открытии файла с ним связывается поток ввода-вывода. Выводимая информация записывается в поток, вводимая информация считывается из потока.

Основные операции над файлами:

1. Определение файла
2. Открытие файла
3. Работа с файлом
4. Закрытие файла

Вся информация хранится в компьютере в виде 0 и 1, т. е. в двоичном виде. Двоичные файлы отличаются от текстовых только методами работы с ними.

Для работы с файлами необходимо подключить заголовочный файл <fstream>**.**В <fstream> определены несколько классов и подключены заголовочные файлы <ifstream>**—**файловый ввод и <ofstream>**—**файловый вывод.

Файловый ввод/вывод аналогичен стандартному вводу/выводу, единственное отличие – это то, что ввод/вывод выполнятся не на экран, а в файл.

Функция is\_open() возвращает целые значения: 1 — если файл был успешно открыт, 0 — если файл открыт не был.

Режимы открытия файлов можно устанавливать непосредственно при создании объекта или при вызове функции open(). Режимы открытия файлов можно комбинировать с помощью поразрядной логической операции **или** |,например: ios\_base::out | ios\_base::trunc — открытие файла для записи, предварительно очистив его.

| **Константа** | **Описание** |
| --- | --- |
| **ios\_base::in** | открыть файл для чтения |
| **ios\_base::out** | открыть файл для записи |
| **ios\_base::ate** | при открытии переместить указатель в конец файла |
| **ios\_base::app** | открыть файл для записи в конец файла |
| **ios\_base::trunc** | удалить содержимое файла, если он существует |
| **ios\_base::binary** | открытие файла в двоичном режиме |

Написать функции :

1. Создание и запись данных в файл, название которого вводится с клавиатуры;

2. Чтение файла(название файла не нужно вводить с клавиатуры);

3. Удаление содержимого в файле;

4. Удаление файла.

|  |
| --- |
| /\*  Написать функции :  1. Создание и запись данных в файл, название которого вводится с клавиатуры;  2. Чтение файла(название файла не нужно вводить с клавиатуры);  3. Удаление содержимого в файле;  4. Удаление файла.  \*/  #include <iostream>  #include <fstream>  #include <windows.h>  #include <string>  using namespace std;  void create(string fileName);  void read(string fileName);  void delInFile(string fileName);  void delFile();  int main() {  int nswitch;  setlocale(LC\_ALL, ".1251");  SetConsoleCP(1251);  SetConsoleOutputCP(1251);  string fileName;  cout << "введите имя файла, с которым будут происходить действия: ";  cin >> fileName;  do {  cout << "\nвыберите действие: \n";  cout << "1 - создание и запись в файл\n";  cout << "2 - чтение файла\n";  cout << "3 - удаление содержимого файла\n";  cout << "4 - удаление файла\n";  cout << "0 - выход\n";  cin >> nswitch;  switch (nswitch) {  case 1: create(fileName); break;  case 2: read(fileName); break;  case 3: delInFile(fileName); break;  case 4: delFile(); break;  }  } while (nswitch != 0);  }  void create(string fileName) {  string strFile;  ofstream file(fileName);  if (!file.is\_open())  cout << "Файл не может быть открыт!\n";  else {  cout << "введите строку, которую хотите напечать в файл: ";  cin >> strFile;  file << strFile;  getline(cin, strFile);  file << strFile;  file.close();  cout << "\nданные записаны в файл\n";  }  }  void read(string fileName) {  string strFile2;  ifstream file(fileName);  if (!file.is\_open())  cout << "Файл не может быть открыт!\n";  else {  file >> strFile2;  cout << strFile2;  getline(file, strFile2);  file >> strFile2;  cout << strFile2;  if (strFile2 == "\0")  cout << "\nфайл пуст\n";  else  cout << "\nданные выведены\n";  file.close();  }  }  void delInFile(string fileName) {  ofstream file(fileName, ios\_base::trunc);  if (!file.is\_open())  cout << "Файл не может быть открыт!\n";  else {  cout << "\nсодержимое файла было успешно удалено\n";  file.close();  }  }  void delFile() {  remove("text.txt");  cout << "\nфайл text.txt был удален\n";  } |
| Данные из консоли |
| Файл был создан в папке с проектом    Информация с клавиатуры записана в файл    Вывел данные из файла в консоль    Удалим содержимое файла    Содержимое файла было удалено      Удалим файл    Файл был удален из папки с проектом |