



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных
технологий

ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 2

по дисциплине

«Структуры и алгоритмы обработки данных»

Тема: «Внешние структуры данных: текстовый и двоичный файлы.»

Выполнил студент группы ИКБО-10-21

Черномуров С.А.

Принял преподаватель

Филатов А.С.

Лабораторная работа выполнена

«__»_____202__ г.

(подпись студента)

«Зачтено»

«__»_____202__ г.

(подпись руководителя)

Москва 2022

1. Цель работы

Получить навыки применения файловых потоков языка C++ (или файлов языка Си) по управлению текстовым и двоичным файлами.

2. Постановка задачи

Задание 1.

1. Разработать программу, управления текстовым файлом.
2. Реализация ввода-вывода на основе файловых потоков C++: ofstream, ifstream.
3. Имя физического файла вводится пользователем и передается в функции обработки через параметр.
4. Разработать функции для выполнения операций над текстовым файлом:
 - 4.1.создание текстового файла кодировки ASCII, содержащего десятичные числа по несколько чисел на строке;
 - 4.2.вывод содержимого текстового файла;
 - 4.3.добавление новой строки в конец файла;
 - 4.4.прочитать значение числа, указав его порядковый номер в файле, и вернуть его значение;
 - 4.5.определить количество чисел в файле.
5. Разработать приложение и выполнить тестирование всех функций. Приложение должно содержать диалоговый интерфейс на основе текстового меню.
6. Приложение должно осуществлять контроль открытия и существования файла в основной программе перед вызовом функций. Перед закрытием файла, проверять отсутствие ошибок ввода и вывода (метод good).
7. Создать модуль и перенести в него все отлаженные функции. Исключить функции из приложения. Отладить приложение, подключив к нему модуль с функциями.

8. Разработать функции для реализации дополнительных операций, определенных вариантом и сохранить их в модуле с остальными функциями.
9. Выполнить тестирование приложения в полном объеме.

Задание 2.

Разработать программу управления двоичными файлами с записями фиксированной длины. Общие требования: файл состоит из записей определенной структуры, согласно варианту. Записи имеют ключ, уникальный в пределах файла.

1. Разработать структуру записи двоичного файла согласно варианту задания.
2. Подготовить тестовые данные в текстовом файле с кодировкой ASCII, в соответствии со структурой записи варианта. При открытии файла выполнить контроль его существования и открытия.
3. Имя файла вводит пользователь.
4. При открытии файла обеспечить контроль существования и открытия файла.
5. При применении механизма прямого доступа к записи файла выполнить контроль присутствия записи с заданным номером в файле.
6. Разработать функции для выполнения операций:
 - 6.1. преобразование тестовых данных из текстового файла в двоичный файл;
 - 6.2. преобразование данных из двоичного файла в текстовый;
 - 6.3. вывод всех записей двоичного файла;
 - 6.4. доступ к записи по ее порядковому номеру в файле, используя механизм прямого доступа к записи в двоичном файле;
 - 6.5. удаление записи с заданным значением ключа, выполнить путем замены на последнюю запись.
 - 6.6. манипулирование записями в двоичном файле согласно дополнительным операциям, определенным в варианте;

7. Сохраните функции в новом модуле.
8. Разработать приложение, демонстрирующее выполнение всех операций, подключив к нему модуль с функциями.
9. Выполнить тестирование приложения, продемонстрировав выполнение всех операций.

Вариант №27. Условие задания:

Упражнение 1	Создать новый файл из значений исходного, представить в нем на двух отдельных строках значения для заполнения двух массивов: на первой строке нечетные числа, а на второй четные. В начале каждой строки указать количество чисел в строке.
Упражнение 2	<p>Структура записи:</p> <p>Учет техосмотра автомобилей. Структура записи об автомобиле: Номер (код региона, цифровой код, буквенный код), Модель, Цвет, Сведения о владельце (Фамилия, Имя, Адрес), дата последнего техосмотра.</p> <p>Доп. Операция:</p> <ol style="list-style-type: none"> 1) Заменить дату техосмотра у автомобиля с заданным номером на новую дату. 2) Сформировать список авто одной модели, прошедших техосмотр в заданную дату.

3. Решение

Инструменты ввода-вывода, разработанные для языка С, не соответствуют сегодняшним стандартам безопасности, так как активно используют указатели, поэтому в С++ были разработаны стандартные классы ввода-вывода.

Для вышеперечисленных целей применяются классы `ofstream`, `ifstream`, `fstream`, которые являются наследниками класса `ios`. Для использования этих классов необходимо подключить библиотеку `<fstream>`.

Файлы можно открывать, используя разные флаги:

Флаг	Назначение
<code>ios::in</code>	Открыть для ввода (выбирается по умолчанию для <code>ifstream</code>)
<code>ios::out</code>	Открыть для вывода (выбирается по умолчанию для <code>ofstream</code>)
<code>ios::binary</code>	Открыть файл в бинарном виде
<code>ios::app</code>	Дозапись в конец файла
<code>ios::ate</code>	Установить файловый указатель на конец файла
<code>ios::trunc</code>	Уничтожить содержимое, если файл существует (выбирается по умолчанию, если флаг <code>out</code> указан, а флаги <code>ate</code> и <code>app</code> – нет)

Флаги можно комбинировать, перечисляя их через символ “|”.

Переопределённый в классах работы с файлами оператор включения (`<<`) записывает данные в файловый поток. Оператор извлечения (`>>`) производит чтение данных из файлового потока.

После окончания работы с файлом, для сохранения изменений, поток необходимо закрыть (метод `close()`). В случае незакрытия потока изменения в файле не будут сохранены (за исключением случаев, когда у объекта потока, явно или неявно, будет вызван деструктор, тогда поток закроется автоматически).

Объяснение алгоритмов применяемых функций

Задание 1

Открывается поток `fout` для записи в файл `file_name`, после чего в поток записывается строка `data` и поток `fout` закрывается.

```
//Функция записи в файл
void fileInput(string file_name, string data) {
    ofstream fout;
    fout.open(file_name);

    if (fout.is_open()) {
        if (fout.good()) fout << data;
    }
    if (fout.good()) fout.close();
}
}
```

Открывается поток fout для дозаписи в конец файла file_name, после чего в поток записывается строка data и поток fout закрывается.

```
//Функция дозаписи в конец файла
void fileAppend(string file_name, string data) {
    ofstream fout;
    fout.open(file_name, ios::app);

    if (fout.is_open()) {
        if (fout.good())
            fout << data;
    }

    if (fout.good()) fout.close();
}
}
```

Открывается поток fin для чтения из файла file_name, после чего в строку res считываются данные из потока fin и поток fin закрывается, функция возвращает значение res.

```
//Функция чтения файла
string fileOutput(string file_name) {
    ifstream fin;
    fin.open(file_name);
    char chr;
    string res = "";
    if (fin.is_open()) {
        while (fin.get(chr) && fin.good()) {
            res+=chr;
        }
    }

    if (fin.eof())
        fin.close();

    return res;
}
}
```

Формируется текст finalStr, который состоит из случайного количества строк случайной длины, в которых содержатся случайные числа, далее этот текст записывается в файл file_name.

```
//Функция заполнения файла десятичными числами
```

```

void filePasteNumbers(string file_name){
    int strNumber = rand() % 11+1;
    string finalStr = "";
    for (int i = 0; i < strNumber; i++) {
        int digitsNumber = rand() % 11 + 1;
        for (int j = 0; j < digitsNumber; j++) {
            finalStr += to_string(rand() % 90+10);
            if (j != digitsNumber - 1) finalStr += " ";
        }
        if (i != strNumber - 1) finalStr += "\n";
    }
    fileInput(file_name, finalStr);}

```

Формируется строка finalStr случайной длины, состоящая из случайных чисел, после чего эта строка дозаписывается в конец файла file_name.

```

//Функция добавления строки в файл
void fileAppendString(string file_name){
    string finalStr = "";
    int digitsNumber = rand() % 11 + 1;
    for (int j = 0; j < digitsNumber; j++) {
        finalStr += to_string(rand() % 90 + 10);
        if (j != digitsNumber - 1) finalStr += " ";
    }
    finalStr = "\n" + finalStr;

    fileAppend(file_name, finalStr);
}

```

В строку finalStr считывается файл file_name, далее эта строка обрабатывается, после чего в ней по значению index находится искомое число num.

```

//Функция чтения числа по его индексу
int fileReadNumber(string file_name, int index){

    string finalStr=fileOutput(file_name);

    for (int i = 0; i < finalStr.length(); i++) {
        if (finalStr[i] == '\n') finalStr[i] = ' ';
    }

    string res = "";
    res += finalStr[3 * (index - 1)];
    res += finalStr[3 * (index - 1)+1];
    int num = atoi(res.c_str());

    return num;
}

```

В строку finalStr считывается файл file_name, далее эта строка обрабатывается, после чего в строке finalStr считается количество пробелов count и функция возвращает инкрементированное значение count.

```
//Функция подсчета количества чисел
int fileCountNumber(string file_name){

    string finalStr = fileOutput(file_name);

    for (int i = 0; i < finalStr.length(); i++) {
        if (finalStr[i] == '\n') finalStr[i] = ' ';
    }
    int count = 0;
    for (int i = 0; i < finalStr.length(); i++) {
        if (finalStr[i] == ' ') count++;
    }
    return count+1;
}
```

В строку strToInp считывается файл file_name_1, в finalStr копируется значение strToInp, далее строка finalStr обрабатывается, четные и нечетные числа сортируются в отведенные им строки even и odd, далее к строкам even и odd вначале приписывается количество чисел в этих строках и строки strToInp, odd, even конкатенируются и записываются в файл file_name_2.

```
//Функция добавления массивов в конец файла
void fileAppendArrays(string file_name_1, string file_name_2) {
    string strToInp = fileOutput(file_name_1);
    string finalStr = strToInp;
    for (int i = 0; i < finalStr.length(); i++) {
        if (finalStr[i] == '\n') finalStr[i] = ' ';
    }

    string even = "";
    string odd = "";

    for (int i = 0; i < finalStr.length(); i+=3) {
        string s = "";
        s += finalStr[i];
        s += finalStr[i + 1];
        int num = atoi(s.c_str());

        if (num % 2 == 0) even = even + " " + s;
        else odd = odd + " " + s;
    }
    even = to_string(even.length() / 3) + even;
    odd = to_string(odd.length() / 3) + odd;

    strToInp = strToInp + "\n\n" + odd + "\n" + even;
    fileInput(file_name_2, strToInp);
}
```

Задание 2

В функции открываются два потока fin и fout (один для чтения, другой для записи в бинарном режиме) и привязываются к файлам file_name_txt и file_name_bin соответственно, далее из потокового буфера fin данные

записываются в поток fout и оба потока закрываются в случае отсутствия ошибок.

```
//Функция конвертации текстового файла в бинарный
void textToBin(string file_name_txt, string file_name_bin) {
    fstream fin = fstream(file_name_txt, ios::in);
    fstream fout = fstream(file_name_bin, ios::binary | ios::out);
    if (fin.is_open() && fout.is_open()) {
        fout << fin.rdbuf();
        if (fin.eof())
            fin.close();
        if (fout.fail() != ios::failbit)
            fout.close();
    }
}
```

В функции открываются два потока fin и fout (один для чтения в бинарном режиме, другой для записи в бинарном режиме) и привязываются к файлам file_name_bin и file_name_txt соответственно, далее из потокового буфера fin данные записываются в поток fout и оба потока закрываются в случае отсутствия ошибок.

```
//Функция конвертации бинарного файла в текстовый
void binToText(string file_name_txt, string file_name_bin) {
    fstream fin = fstream(file_name_bin, ios::binary | ios::in);
    fstream fout = fstream(file_name_txt, ios::binary | ios::out);
    if (fin.is_open() && fout.is_open()) {

        fout << fin.rdbuf();

        if (fin.eof())
            fin.close();
        if (fout.fail() != ios::failbit)
            fout.close();
    }
}
```

В функции открывается поток fin для чтения в бинарном режиме файла file_name, далее данные из потокового буфера fin передаются в поток вывода cout, и поток fin закрывается в случае отсутствия ошибок.

```
//Функция вывода содержимого бинарного файла
void fileOutputBin(string file_name) {
    fstream fin(file_name, ios::binary | ios::in);

    if (fin.is_open()) cout << fin.rdbuf();

    if (fin.eof())
        fin.close();
}
```

В функции открывается поток `fin` для чтения в бинарном режиме и привязывается к файлу `file_name`, далее открывается аналогичный поток `fin2`, данные из потока `fin2` записываются в строку `str`, поток `fin2` закрывается. Далее, с помощью механизма прямого доступа к байту по его номеру, указатель потока `fin` ставится в нужное место и с этого места считывается строка `line`, которая и является результатом работы функции. Далее, в случае отсутствия ошибок, поток `fin` закрывается.

```
//Функция получения записи по ее индексу
string fileGet(string file_name, int index) {
    ifstream fin(file_name, ios::in | ios::binary);
    string line = "";
    string str = "";
    if (fin.is_open()) {
        ifstream fin2(file_name, ios::in | ios::binary);
        while (fin2) {
            str += fin2.get();
        }
        fin2.close();

        if (index * 37 <= str.size() + 10) {
            fin.seekg(index * 37 * sizeof(char));
            getline(fin, line);
        }
    }
    if (fin.fail() != ios::failbit)
        fin.close();
    return line;
}
```

В функции открывается поток `fin` в бинарном формате в режиме как чтения, так и записи и привязывается к файлу `file_name`, далее данные из потока `fin` считываются в переменные `num`, `mark`, `color`, `fam`, `no`, `data`, находится запись, содержащая правильный номер автомобиля, дата прохождения техосмотра в этой записи заменяется на значение `date`, поток `fin` закрывается в случае отсутствия ошибок.

```
//Функция замены даты техосмотра
void switchTI(string file_name, string number, string date){
    fstream fin(file_name, ios::in | ios::binary | ios::out);
    string num, mark, color, fam, no, data;
    string hash;
    string index="-1";
    while (fin) {
        fin >> hash;
        fin >> num;
        fin >> mark;
        fin >> color;
        fin >> fam;
```

```

        fin >> no;
        fin >> data;
        if (number == num) { index = hash; break; }
    }
    if (index != "-1") {
        fin.seekp(atoi(index.c_str()) * 37);
        data = date;
        fin << hash << " ";
        fin << num << " ";
        fin << mark << " ";
        fin << color << " ";
        fin << fam << " ";
        fin << no << " ";
        fin << data;
    }
    if (fin.fail() != ios::failbit)
        fin.close();
}

```

В функции открывается поток `fin` для чтения в бинарном формате файла `file_name`, далее выводятся записи, в которых марка автомобиля и дата техосмотра совпадают со значением `car` и `date`, и поток `fin` закрывается в случае отсутствия ошибок.

```

//Функция вывода списка автомобилей определенной марки с определенной
//датой техосмотра
void checkCars(string file_name, string car, string date){
    fstream fin(file_name, ios::in | ios::binary);
    string num, mark, color, fam, no, data;
    string hash;
    while (!fin.eof()) {
        fin >> hash;
        fin >> num;
        fin >> mark;
        fin >> color;
        fin >> fam;
        fin >> no;
        fin >> data;
        if (car == mark && date == data)
            cout << hash << " " << num << " " << mark << " " << color << "
" << fam << " " << no << " " << data<<"\n";
    }
    if (fin.fail() != ios::failbit)
        fin.close();
}
}

```

В функции в строку `str` записываются данные файла `file_name`, далее в строку `strToCopy` записывается последняя запись из файла, и этой записью заменяется искомая запись, далее последняя запись удаляется, так как была перемещена на место удаленной, строка `str` записывается в файл `file_name`.

```

//Функция удаления записи по ее индексу
void fileDelete(string file_name, int index) {
    string str = fileOutput(file_name);
    string strToCopy="";
}

```

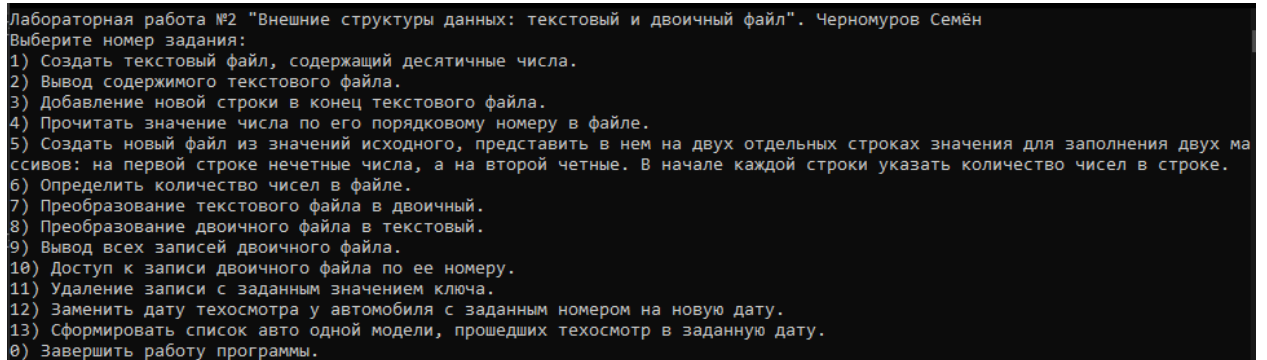
```

    for (int i = str.rfind("\n") + 1; i < str.length(); i++) {
        strToCopy += str[i];
    }
    for (int i = index * 37; i < index * 37 + strToCopy.length(); i++) {
        str[i] = strToCopy[i - index * 37];
    }
    str[index * 37+1] = index + '0';
    str=str.erase(str.rfind("\n"), strToCopy.length());
    fileInput(file_name, str);
}

```

Описание работы пользовательского интерфейса

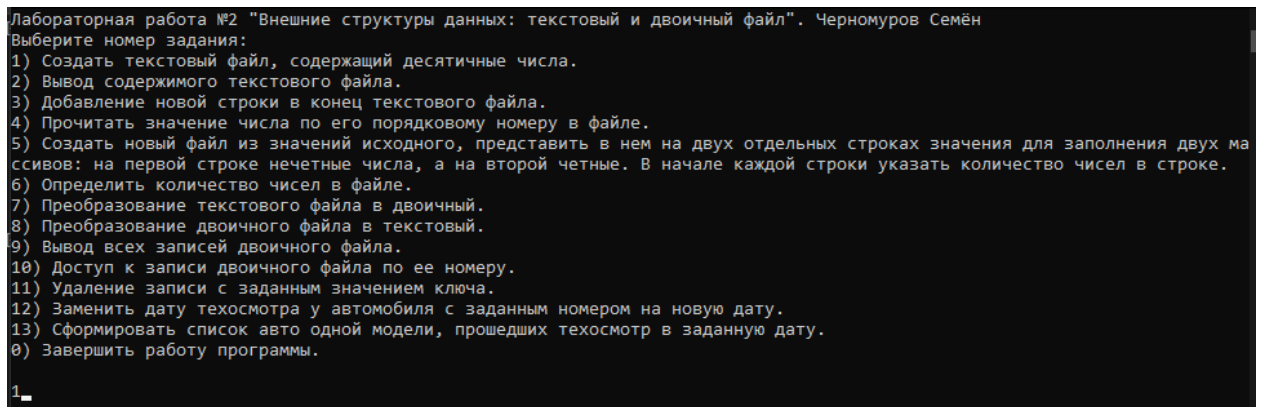
Сначала выводится выбор задания, пользователь вводом с клавиатуры выбирает задание. После чего консоль очищается и отображается ввод в зависимости от выбранного задания. Введенные данные обрабатываются, результат обработки выводится на экран, программа автоматически перезапускается.



Лабораторная работа №2 "Внешние структуры данных: текстовый и двоичный файл". Черномуров Семён
 Выберите номер задания:
 1) Создать текстовый файл, содержащий десятичные числа.
 2) Вывод содержимого текстового файла.
 3) Добавление новой строки в конец текстового файла.
 4) Прочитать значение числа по его порядковому номеру в файле.
 5) Создать новый файл из значений исходного, представить в нем на двух отдельных строках значения для заполнения двух массивов: на первой строке нечетные числа, а на второй четные. В начале каждой строки указать количество чисел в строке.
 6) Определить количество чисел в файле.
 7) Преобразование текстового файла в двоичный.
 8) Преобразование двоичного файла в текстовый.
 9) Вывод всех записей двоичного файла.
 10) Доступ к записи двоичного файла по ее номеру.
 11) Удаление записи с заданным значением ключа.
 12) Заменить дату техосмотра у автомобиля с заданным номером на новую дату.
 13) Сформировать список авто одной модели, прошедших техосмотр в заданную дату.
 0) Завершить работу программы.

Рисунок 1. Интерфейс программы

4. Тестирование



Лабораторная работа №2 "Внешние структуры данных: текстовый и двоичный файл". Черномуров Семён
 Выберите номер задания:
 1) Создать текстовый файл, содержащий десятичные числа.
 2) Вывод содержимого текстового файла.
 3) Добавление новой строки в конец текстового файла.
 4) Прочитать значение числа по его порядковому номеру в файле.
 5) Создать новый файл из значений исходного, представить в нем на двух отдельных строках значения для заполнения двух массивов: на первой строке нечетные числа, а на второй четные. В начале каждой строки указать количество чисел в строке.
 6) Определить количество чисел в файле.
 7) Преобразование текстового файла в двоичный.
 8) Преобразование двоичного файла в текстовый.
 9) Вывод всех записей двоичного файла.
 10) Доступ к записи двоичного файла по ее номеру.
 11) Удаление записи с заданным значением ключа.
 12) Заменить дату техосмотра у автомобиля с заданным номером на новую дату.
 13) Сформировать список авто одной модели, прошедших техосмотр в заданную дату.
 0) Завершить работу программы.
 1_

Рисунок 2. Выбор задания 1

```
Введите название файла: 1.txt
Файл успешно создан.
```

Рисунок 3. Указание имени файла

```
83 18 76
86 65 36 65 99 68 61 68 38
86 83 23 39 48 12
```

Рисунок 4. Результат работы программы (задание 1)

Создан файл 1.txt, в него записано случайно сгенерированное число строк случайной длины, содержащих случайные числа (рис. 4).

```
Лабораторная работа №2 "Внешние структуры данных: текстовый и двоичный файл". Черномуров Семён
Выберите номер задания:
1) Создать текстовый файл, содержащий десятичные числа.
2) Вывод содержимого текстового файла.
3) Добавление новой строки в конец текстового файла.
4) Прочитать значение числа по его порядковому номеру в файле.
5) Создать новый файл из значений исходного, представить в нем на двух отдельных строках значения для заполнения двух массивов: на первой строке нечетные числа, а на второй четные. В начале каждой строки указать количество чисел в строке.
6) Определить количество чисел в файле.
7) Преобразование текстового файла в двоичный.
8) Преобразование двоичного файла в текстовый.
9) Вывод всех записей двоичного файла.
10) Доступ к записи двоичного файла по ее номеру.
11) Удаление записи с заданным значением ключа.
12) Заменить дату техосмотра у автомобиля с заданным номером на новую дату.
13) Сформировать список авто одной модели, прошедших техосмотр в заданную дату.
0) Завершить работу программы.
2_
```

Рисунок 5. Выбор задания 2

```
Введите название файла: 1.txt

Содержимое выбранного файла:

83 18 76
86 65 36 65 99 68 61 68 38
86 83 23 39 48 12
```

Рисунок 6. Вывод содержимого указанного файла

Открыт файл 1.txt в формате чтения, данные с него считаны и выведены на экран.

Лабораторная работа №2 "Внешние структуры данных: текстовый и двоичный файл". Черномуров Семён
 Выберите номер задания:
 1) Создать текстовый файл, содержащий десятичные числа.
 2) Вывод содержимого текстового файла.
 3) Добавление новой строки в конец текстового файла.
 4) Прочитать значение числа по его порядковому номеру в файле.
 5) Создать новый файл из значений исходного, представить в нем на двух отдельных строках значения для заполнения двух массивов: на первой строке нечетные числа, а на второй четные. В начале каждой строки указать количество чисел в строке.
 6) Определить количество чисел в файле.
 7) Преобразование текстового файла в двоичный.
 8) Преобразование двоичного файла в текстовый.
 9) Вывод всех записей двоичного файла.
 10) Доступ к записи двоичного файла по ее номеру.
 11) Удаление записи с заданным значением ключа.
 12) Заменить дату техосмотра у автомобиля с заданным номером на новую дату.
 13) Сформировать список авто одной модели, прошедших техосмотр в заданную дату.
 0) Завершить работу программы.
 3

Рисунок 7. Выбор задания 3

Введите название файла: 1.txt
 Строка успешно добавлена к файлу.

Рисунок 8. Указание имени файла

```

83 18 76
86 65 36 65 99 68 61 68 38
86 83 23 39 48 12
92 41 29 32
  
```

Рисунок 9. Результат работы программы (задание 3)

В программе сгенерирована строка (92 41 29 32), которая дозаписана в конец файла 1.txt

Лабораторная работа №2 "Внешние структуры данных: текстовый и двоичный файл". Черномуров Семён
 Выберите номер задания:
 1) Создать текстовый файл, содержащий десятичные числа.
 2) Вывод содержимого текстового файла.
 3) Добавление новой строки в конец текстового файла.
 4) Прочитать значение числа по его порядковому номеру в файле.
 5) Создать новый файл из значений исходного, представить в нем на двух отдельных строках значения для заполнения двух массивов: на первой строке нечетные числа, а на второй четные. В начале каждой строки указать количество чисел в строке.
 6) Определить количество чисел в файле.
 7) Преобразование текстового файла в двоичный.
 8) Преобразование двоичного файла в текстовый.
 9) Вывод всех записей двоичного файла.
 10) Доступ к записи двоичного файла по ее номеру.
 11) Удаление записи с заданным значением ключа.
 12) Заменить дату техосмотра у автомобиля с заданным номером на новую дату.
 13) Сформировать список авто одной модели, прошедших техосмотр в заданную дату.
 0) Завершить работу программы.
 4_

Рисунок 10. Выбор задания 4

Введите название файла: 1.txt
 Введите индекс искомого элемента: 5
 Число с указанным индексом: 65

Рисунок 11. Указание имени файла, индекса числа и последующий вывод искомого числа

Программа открывает файл 1.txt, находит в нем число с номером 5 (83 18 76 86 65 36...) и выводит его на экран.

Лабораторная работа №2 "Внешние структуры данных: текстовый и двоичный файл". Черномуров Семён
 Выберите номер задания:
 1) Создать текстовый файл, содержащий десятичные числа.
 2) Вывод содержимого текстового файла.
 3) Добавление новой строки в конец текстового файла.
 4) Прочитать значение числа по его порядковому номеру в файле.
 5) Создать новый файл из значений исходного, представить в нем на двух отдельных строках значения для заполнения двух массивов: на первой строке нечетные числа, а на второй четные. В начале каждой строки указать количество чисел в строке.
 6) Определить количество чисел в файле.
 7) Преобразование текстового файла в двоичный.
 8) Преобразование двоичного файла в текстовый.
 9) Вывод всех записей двоичного файла.
 10) Доступ к записи двоичного файла по ее номеру.
 11) Удаление записи с заданным значением ключа.
 12) Заменить дату техосмотра у автомобиля с заданным номером на новую дату.
 13) Сформировать список авто одной модели, прошедших техосмотр в заданную дату.
 0) Завершить работу программы.
 5_

Рисунок 12. Выбор задания 5

Введите название первого файла: 1.txt
 Введите название второго файла: 2.txt

Рисунок 13. Указание имен файлов

```
83 18 76
86 65 36 65 99 68 61 68 38
86 83 23 39 48 12
92 41 29 32

10 83 65 65 99 61 83 23 39 41 29
12 18 76 86 36 68 68 38 86 48 12 92 32
```

Рисунок 14. Результат работы программы (задание 5)

Программа открывает файл 1.txt в формате чтения, файл 2.txt в формате записи, записывает в файл 2.txt данные файла 1.txt и две строки, состоящих из 10 и 12 чисел соответственно. В первой строке содержатся нечетные числа файла 1.txt, во второй – четные.

Лабораторная работа №2 "Внешние структуры данных: текстовый и двоичный файл". Черномуров Семён
 Выберите номер задания:
 1) Создать текстовый файл, содержащий десятичные числа.
 2) Вывод содержимого текстового файла.
 3) Добавление новой строки в конец текстового файла.
 4) Прочитать значение числа по его порядковому номеру в файле.
 5) Создать новый файл из значений исходного, представить в нем на двух отдельных строках значения для заполнения двух массивов: на первой строке нечетные числа, а на второй четные. В начале каждой строки указать количество чисел в строке.
 6) Определить количество чисел в файле.
 7) Преобразование текстового файла в двоичный.
 8) Преобразование двоичного файла в текстовый.
 9) Вывод всех записей двоичного файла.
 10) Доступ к записи двоичного файла по ее номеру.
 11) Удаление записи с заданным значением ключа.
 12) Заменить дату техосмотра у автомобиля с заданным номером на новую дату.
 13) Сформировать список авто одной модели, прошедших техосмотр в заданную дату.
 0) Завершить работу программы.
 6_

Рисунок 15. Выбор задания 6

Введите название файла: 1.txt
 Количество чисел в файле: 22

Рисунок 16. Указание файла и результат работы программы (задание 6)

Программа открывает файл 1.txt в формате чтения, считает количество чисел в этом файле и выводит их количество.

```
Лабораторная работа №2 "Внешние структуры данных: текстовый и двоичный файл". Черномуров Семён
Выберите номер задания:
1) Создать текстовый файл, содержащий десятичные числа.
2) Вывод содержимого текстового файла.
3) Добавление новой строки в конец текстового файла.
4) Прочитать значение числа по его порядковому номеру в файле.
5) Создать новый файл из значений исходного, представить в нем на двух отдельных строках значения для заполнения двух массивов: на первой строке нечетные числа, а на второй четные. В начале каждой строки указать количество чисел в строке.
6) Определить количество чисел в файле.
7) Преобразование текстового файла в двоичный.
8) Преобразование двоичного файла в текстовый.
9) Вывод всех записей двоичного файла.
10) Доступ к записи двоичного файла по ее номеру.
11) Удаление записи с заданным значением ключа.
12) Заменить дату техосмотра у автомобиля с заданным номером на новую дату.
13) Сформировать список авто одной модели, прошедших техосмотр в заданную дату.
0) Завершить работу программы.
7
```

Рисунок 17. Выбор задания 7

```
Введите название файла: data.txt
Введите название бинарного файла: databin.bin
Конвертация произошла успешно.
```

Рисунок 18. Указание имен файлов

```
00 A129BC HON R PUPKIN VS 28.08.2003
01 A130BC NIS R ZHUKOV VS 28.08.2003
02 A131BC TOY R DURKIN VS 28.08.2003
03 A135BC TOY R DURKIN VS 28.08.2003
04 A134BC HON R PUPKIN VS 28.08.2003
05 A134BC NIS R GASHIK MT 28.08.2003
```

Рисунок 19. Содержимое текстового файла

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	UTF-8 or .	
000000	30	30	20	41	31	32	39	42	43	20	48	4F	4E	20	52	20	00 A129B	C HON R
000010	50	55	50	48	49	4E	20	56	53	20	32	38	2E	30	38	2E	PUPKIN V	S 28.08.
000020	32	30	30	33	0A	30	31	20	41	31	33	30	42	43	20	4E	2003 \n01	A130BC N
000030	49	53	20	52	20	5A	48	55	48	4F	56	20	56	53	20	32	IS R ZHU	KOV VS 2
000040	38	2E	30	38	2E	32	30	30	33	0A	30	32	20	41	31	33	8.08.200	3 \n02 A13
000050	31	42	43	20	54	4F	59	20	52	20	44	55	52	48	49	4E	1BC TOY	R DURKIN
000060	20	56	53	20	32	38	2E	30	38	2E	32	30	30	33	0A	30	VS 28.0	8.2003 \n0
000070	33	20	41	31	33	35	42	43	20	54	4F	59	20	52	20	44	3 A135BC	TOY R D
000080	55	52	48	49	4E	20	56	53	20	32	38	2E	30	38	2E	32	URKIN VS	28.08.2
000090	30	30	33	0A	30	34	20	41	31	33	34	42	43	20	48	4F	003 \n04 A	134BC NO
0000A0	4E	20	52	20	50	55	50	48	49	4E	20	56	53	20	32	38	N R PUPK	IN VS 28
0000B0	2E	30	38	2E	32	30	30	33	0A	30	35	20	41	31	33	34	.08.2003	\n05 A134
0000C0	42	43	20	4E	49	53	20	52	20	47	41	53	48	49	48	20	BC NIS R	GASHIK
0000D0	4D	54	20	32	38	2E	30	38	2E	32	30	30	33				MT 28.08	.2003

Рисунок 20. Содержимое бинарного файла

Программа открывает файлы data.txt в режиме чтения, databin.bin в бинарном режиме (для записи), и записывает содержимое файла data.txt в файл databin.bin

```
Лабораторная работа №2 "Внешние структуры данных: текстовый и двоичный файл". Черномуров Семён
Выберите номер задания:
1) Создать текстовый файл, содержащий десятичные числа.
2) Вывод содержимого текстового файла.
3) Добавление новой строки в конец текстового файла.
4) Прочитать значение числа по его порядковому номеру в файле.
5) Создать новый файл из значений исходного, представить в нем на двух отдельных строках значения для заполнения двух массивов: на первой строке нечетные числа, а на второй четные. В начале каждой строки указать количество чисел в строке.
6) Определить количество чисел в файле.
7) Преобразование текстового файла в двоичный.
8) Преобразование двоичного файла в текстовый.
9) Вывод всех записей двоичного файла.
10) Доступ к записи двоичного файла по ее номеру.
11) Удаление записи с заданным значением ключа.
12) Заменить дату техосмотра у автомобиля с заданным номером на новую дату.
13) Сформировать список авто одной модели, прошедших техосмотр в заданную дату.
0) Завершить работу программы.
8_
```

Рисунок 21. Выбор задания 8

```
Введите название бинарного файла: databin.bin
Введите название файла: 2.txt
Конвертация произошла успешно.
```

Рисунок 22. Указание имен файлов

```
00 A129BC HON R PUPKIN VS 28.08.2003
01 A130BC NIS R ZHUKOV VS 28.08.2003
02 A131BC TOY R DURKIN VS 28.08.2003
03 A135BC TOY R DURKIN VS 28.08.2003
04 A134BC HON R PUPKIN VS 28.08.2003
05 A134BC NIS R GASHIK MT 28.08.2003
```

Рисунок 23. Результат выполнения программы (задание 8)

Программа открывает файлы databin.bin в режиме бинарного чтения, 2.txt для записи в бинарном формате, и записывает содержимое файла databin.bin в файл 2.txt

```

Лабораторная работа №2 "Внешние структуры данных: текстовый и двоичный файл". Черномуров Семён
Выберите номер задания:
1) Создать текстовый файл, содержащий десятичные числа.
2) Вывод содержимого текстового файла.
3) Добавление новой строки в конец текстового файла.
4) Прочитать значение числа по его порядковому номеру в файле.
5) Создать новый файл из значений исходного, представить в нем на двух отдельных строках значения для заполнения двух массивов: на первой строке нечетные числа, а на второй четные. В начале каждой строки указать количество чисел в строке.
6) Определить количество чисел в файле.
7) Преобразование текстового файла в двоичный.
8) Преобразование двоичного файла в текстовый.
9) Вывод всех записей двоичного файла.
10) Доступ к записи двоичного файла по ее номеру.
11) Удаление записи с заданным значением ключа.
12) Заменить дату техосмотра у автомобиля с заданным номером на новую дату.
13) Сформировать список авто одной модели, прошедших техосмотр в заданную дату.
0) Завершить работу программы.
9_

```

Рисунок 24. Выбор задания 9

```

Введите название файла: databin.bin

Содержимое выбранного файла:

00 A129BC HON R PUPKIN VS 28.08.2003
01 A130BC NIS R ZHUKOV VS 28.08.2003
02 A131BC TOY R DURKIN VS 28.08.2003
03 A135BC TOY R DURKIN VS 28.08.2003
04 A134BC HON R PUPKIN VS 28.08.2003
05 A134BC NIS R GASHIK MT 28.08.2003

```

Рисунок 25. Указание файла и его содержимое

Программа открывает файл databin.bin в режиме бинарного чтения и выводит его содержимое на экран.

```

Лабораторная работа №2 "Внешние структуры данных: текстовый и двоичный файл". Черномуров Семён
Выберите номер задания:
1) Создать текстовый файл, содержащий десятичные числа.
2) Вывод содержимого текстового файла.
3) Добавление новой строки в конец текстового файла.
4) Прочитать значение числа по его порядковому номеру в файле.
5) Создать новый файл из значений исходного, представить в нем на двух отдельных строках значения для заполнения двух массивов: на первой строке нечетные числа, а на второй четные. В начале каждой строки указать количество чисел в строке.
6) Определить количество чисел в файле.
7) Преобразование текстового файла в двоичный.
8) Преобразование двоичного файла в текстовый.
9) Вывод всех записей двоичного файла.
10) Доступ к записи двоичного файла по ее номеру.
11) Удаление записи с заданным значением ключа.
12) Заменить дату техосмотра у автомобиля с заданным номером на новую дату.
13) Сформировать список авто одной модели, прошедших техосмотр в заданную дату.
0) Завершить работу программы.
10

```

Рисунок 26. Выбор задания 10

```

Введите название бинарного файла: databin.bin
Введите индекс искомого элемента: 2
Запись с указанным индексом: 02 A131BC TOY R DURKIN VS 28.08.2003

```

Рисунок 27. Указание файла и индекса элемента, результат выполнения программы (задание 10)

Программа открывает файл databin.bin в формате бинарного чтения, находит по заданному индексу запись и выводит ее на экран.

```
Введите название бинарного файла: databin.bin
Введите индекс искомого элемента: 199
Запись с указанным индексом: Не найдена
```

Рисунок 28. Попытка обратиться к несуществующей записи

При попытке обращения к несуществующей записи программа оповещает пользователя о том, что искомая запись не найдена.

```
Лабораторная работа №2 "Внешние структуры данных: текстовый и двоичный файл". Черномуров Семён
Выберите номер задания:
1) Создать текстовый файл, содержащий десятичные числа.
2) Вывод содержимого текстового файла.
3) Добавление новой строки в конец текстового файла.
4) Прочитать значение числа по его порядковому номеру в файле.
5) Создать новый файл из значений исходного, представить в нем на двух отдельных строках значения для заполнения двух массивов: на первой строке нечетные числа, а на второй четные. В начале каждой строки указать количество чисел в строке.
6) Определить количество чисел в файле.
7) Преобразование текстового файла в двоичный.
8) Преобразование двоичного файла в текстовый.
9) Вывод всех записей двоичного файла.
10) Доступ к записи двоичного файла по ее номеру.
11) Удаление записи с заданным значением ключа.
12) Заменить дату техосмотра у автомобиля с заданным номером на новую дату.
13) Сформировать список авто одной модели, прошедших техосмотр в заданную дату.
0) Завершить работу программы.
11
```

Рисунок 29. Выбор задания 11

```
Введите название бинарного файла: databin.bin
Введите индекс удаляемого элемента: 3
Запись успешно удалена.
```

Рисунок 30. Указание файла и индекса удаляемого элемента

```
Введите название файла: databin.bin

Содержимое выбранного файла:

00 A129BC HON R PUPKIN VS 28.08.2003
01 A130BC NIS R ZHUKOV VS 28.08.2003
02 A131BC TOY R DURKIN VS 28.08.2003
03 A134BC NIS R GASHIK MT 28.08.2003
04 A134BC HON R PUPKIN VS 28.08.2003
```

Рисунок 31. Результат работы программы (задание 11)

Программа открывает файл databin.bin в режиме бинарного чтения, считывает содержимое файла в строку, удаляет из строки нужную запись, после чего перезаписывает содержимое файла databin.bin на полученную строку.

```

Лабораторная работа №2 "Внешние структуры данных: текстовый и двоичный файл". Черномуров Семён
Выберите номер задания:
1) Создать текстовый файл, содержащий десятичные числа.
2) Вывод содержимого текстового файла.
3) Добавление новой строки в конец текстового файла.
4) Прочитать значение числа по его порядковому номеру в файле.
5) Создать новый файл из значений исходного, представить в нем на двух отдельных строках значения для заполнения двух массивов: на первой строке нечетные числа, а на второй четные. В начале каждой строки указать количество чисел в строке.
6) Определить количество чисел в файле.
7) Преобразование текстового файла в двоичный.
8) Преобразование двоичного файла в текстовый.
9) Вывод всех записей двоичного файла.
10) Доступ к записи двоичного файла по ее номеру.
11) Удаление записи с заданным значением ключа.
12) Заменить дату техосмотра у автомобиля с заданным номером на новую дату.
13) Сформировать список авто одной модели, прошедших техосмотр в заданную дату.
0) Завершить работу программы.
12_

```

Рисунок 32. Выбор задания 12

```

Введите название бинарного файла: databin.bin
Введите номер нужного автомобиля: A129BC
Введите изменяемую дату: 10.10.2010
Дата успешно изменена.

```

Рисунок 33. Указание имени файла, номера авто, новой даты техосмотра

```

Введите название файла: databin.bin

Содержимое выбранного файла:

00 A129BC HON R PUPKIN VS 10.10.2010
01 A130BC NIS R ZHUKOV VS 28.08.2003
02 A131BC TOY R DURKIN VS 28.08.2003
03 A134BC NIS R GASHIK MT 28.08.2003
04 A134BC HON R PUPKIN VS 28.08.2003

```

Рисунок 34. Результат выполнения программы (задание 12)

Программа открывает файл databin.bin в бинарном режиме (для чтения и для записи), находит запись, содержащую введенный номер автомобиля, и заменяет дату прохождения техосмотра в этой записи на введенную.

```

Лабораторная работа №2 "Внешние структуры данных: текстовый и двоичный файл". Черномуров Семён
Выберите номер задания:
1) Создать текстовый файл, содержащий десятичные числа.
2) Вывод содержимого текстового файла.
3) Добавление новой строки в конец текстового файла.
4) Прочитать значение числа по его порядковому номеру в файле.
5) Создать новый файл из значений исходного, представить в нем на двух отдельных строках значения для заполнения двух массивов: на первой строке нечетные числа, а на второй четные. В начале каждой строки указать количество чисел в строке.
6) Определить количество чисел в файле.
7) Преобразование текстового файла в двоичный.
8) Преобразование двоичного файла в текстовый.
9) Вывод всех записей двоичного файла.
10) Доступ к записи двоичного файла по ее номеру.
11) Удаление записи с заданным значением ключа.
12) Заменить дату техосмотра у автомобиля с заданным номером на новую дату.
13) Сформировать список авто одной модели, прошедших техосмотр в заданную дату.
0) Завершить работу программы.
13

```

Рисунок 35. Выбор задания 13

```
Введите название бинарного файла: databin.bin
Введите код автомобиля: TOY
Введите дату: 28.08.2003
Сформированный список:
02 A131BC TOY R DURKIN VS 28.08.2003
```

Рисунок 36. Указание имени файла, марки автомобиля, даты техосмотра и результат выполнения программы (задание 13)

Программа открывает файл databin.bin в режиме бинарного чтения, и выводит на экран записи, содержащие введенные код автомобиля и дату прохождения техосмотра. Программа вывела список автомобилей TOY, которые прошли техосмотр 28.08.2003.

```
Лабораторная работа №2 "Внешние структуры данных: текстовый и двоичный файл". Черномуров Семён
Выберите номер задания:
1) Создать текстовый файл, содержащий десятичные числа.
2) Вывод содержимого текстового файла.
3) Добавление новой строки в конец текстового файла.
4) Прочитать значение числа по его порядковому номеру в файле.
5) Создать новый файл из значений исходного, представить в нем на двух отдельных строках значения для заполнения двух массивов: на первой строке нечетные числа, а на второй четные. В начале каждой строки указать количество чисел в строке.
6) Определить количество чисел в файле.
7) Преобразование текстового файла в двоичный.
8) Преобразование двоичного файла в текстовый.
9) Вывод всех записей двоичного файла.
10) Доступ к записи двоичного файла по ее номеру.
11) Удаление записи с заданным значением ключа.
12) Заменить дату техосмотра у автомобиля с заданным номером на новую дату.
13) Сформировать список авто одной модели, прошедших техосмотр в заданную дату.
0) Завершить работу программы.
0_
```

Рисунок 37. Выбор завершения работы программы

```
Консоль отладки Microsoft Visual Studio
C:\Users\KLINY-ПК\Desktop\data\Учёба\3 семестр\СИА0Д\Zadanie2\Zadanie2\Debug\Zadanie2.exe (процесс 13520) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" -> "Параметры" -> "Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно:
```

Рисунок 38. Результат выбора завершения работы программы

При выборе функции завершения работы программы, программа прекращает свою работу.

```
Введите название файла: adssgfsagsa
Указанного файла не существует.
```

Рисунок 39. Попытка открытия несуществующего файла

При попытке открытия несуществующего файла программа оповестит пользователя (эта проверка универсальна для всех заданий).

Из результатов выполнения программы видно, что программа работает корректно.

5. Вывод

В результате выполнения работы я:

1. Получил навыки применения файловых потоков языка C++ по управлению текстовыми и двоичными файлами.
2. Закрепил знания по тестированию корректности работы программы.
3. Закрепил навыки создания пользовательских интерфейсов.

6. Исходный код программы

Файл `main.cpp` (основной алгоритм программы)

```
#include "functions1.h"
#include "functions2.h"

#include <iostream>
#include <string>
#include <fstream>
using namespace std;

int main() {
    setlocale(LC_ALL, "");
    srand(time(NULL));
    cout << "Лабораторная работа №2 \\"Внешние структуры данных: текстовый и двоичный файл\\". Черномуров Семён\n";

    cout << "Выберите номер задания:\n1) Создать текстовый файл, содержащий десятичные числа.\n" <<
        "2) Вывод содержимого текстового файла.\n" <<
        "3) Добавление новой строки в конец текстового файла.\n" <<
        "4) Прочитать значение числа по его порядковому номеру в файле.\n" <<
        "5) Создать новый файл из значений исходного, представить в нем на двух отдельных строках значения для заполнения двух массивов: на первой строке нечетные числа, а на второй четные. В начале каждой строки указать количество чисел в строке.\n" <<
        "6) Определить количество чисел в файле.\n" <<
        "7) Преобразование текстового файла в двоичный.\n" <<
        "8) Преобразование двоичного файла в текстовый.\n" <<
        "9) Вывод всех записей двоичного файла.\n" <<
        "10) Доступ к записи двоичного файла по ее номеру.\n" <<
        "11) Удаление записи с заданным значением ключа.\n" <<
        "12) Заменить дату техосмотра у автомобиля с заданным номером на новую дату.\n" <<
        "13) Сформировать список авто одной модели, прошедших техосмотр в заданную дату.\n" <<
        "0) Завершить работу программы.\n\n";

    int choice1;
    do {
        cin >> choice1;
```



```

        if (choice1 != 1 && choice1 != 2 && choice1 != 3 && choice1 != 4 &&
choice1 != 5 && choice1 != 6 && choice1 != 7 && choice1 != 8 && choice1 != 9 &&
choice1 != 10 && choice1 != 11 && choice1 != 0 && choice1 != 12 && choice1 != 13)
cout << "Введено неверное значение, попробуйте снова.\n";
    } while (choice1 != 1 && choice1 != 2 && choice1 != 3 && choice1 != 4 &&
choice1 != 5 && choice1 != 6 && choice1 != 7 && choice1 != 8 && choice1 != 9 &&
choice1 != 10 && choice1 != 11 && choice1 != 0 && choice1 != 12 && choice1 != 13);

    system("cls");

    switch (choice1)
    {
    case 1: {
        cout << "Введите название файла: ";
        string file_name;
        cin >> file_name;
        filePasteNumbers(file_name);
        cout << "Файл успешно создан.";
        break;
    }
    case 2: {
        cout << "Введите название файла: ";
        string file_name;
        cin >> file_name;
        if (fstream(file_name, ios::_Nocreate)) {
            cout << "\nСодержимое выбранного файла:\n\n";
            cout << fileOutput(file_name);
        }
        else cout << "\nУказанного файла не существует.\n";
        break;
    }
    case 3: {
        cout << "Введите название файла: ";
        string file_name;
        cin >> file_name;
        if (fstream(file_name, ios::_Nocreate)) {
            fileAppendString(file_name);
            cout << "Строка успешно добавлена к файлу.";
        }
        else cout << "\nУказанного файла не существует.\n";
        break;
    }
    case 4: {
        cout << "Введите название файла: ";
        string file_name;
        cin >> file_name;

        if (fstream(file_name, ios::_Nocreate)) {
            cout << "Введите индекс искомого элемента: ";
            int index;
            cin >> index;
            int res=fileReadNumber(file_name, index);
            cout << "Число с указанным индексом: " << res;
        }
        else cout << "\nУказанного файла не существует.\n";

        break;
    }
    case 5: {
        cout << "Введите название первого файла: ";
        string file_name_1;
        cin >> file_name_1;
        if (fstream(file_name_1, ios::_Nocreate)) {
            cout << "Введите название второго файла: ";

```

```

        string file_name_2;
        cin >> file_name_2;
        fileAppendArrays(file_name_1, file_name_2);
    }
    break;
}
case 6: {
    cout << "Введите название файла: ";
    string file_name;
    cin >> file_name;
    if (fstream(file_name, ios::_Nocreate)) {
        cout << "Количество чисел в файле: " <<
fileCountNumber(file_name);
    }
    else cout << "\nУказанного файла не существует.\n";
    break;
}
case 7: {
    cout << "Введите название файла: ";
    string file_name;
    cin >> file_name;
    if (fstream(file_name, ios::_Nocreate)) {
        cout << "Введите название бинарного файла: ";
        string file_name_bin;
        cin >> file_name_bin;
        textToBin(file_name, file_name_bin);
        cout << "Конвертация произошла успешно.";
    }
    else cout << "\nУказанного файла не существует.\n";
    break;
}
case 8: {
    cout << "Введите название бинарного файла: ";
    string file_name_bin;
    cin >> file_name_bin;
    if (fstream(file_name_bin, ios::_Nocreate)) {
        cout << "Введите название файла: ";
        string file_name;
        cin >> file_name;
        binToText(file_name, file_name_bin);
        cout << "Конвертация произошла успешно.";
    }
    else cout << "\nУказанного файла не существует.\n";
    break;
}
case 9: {
    cout << "Введите название файла: ";
    string file_name;
    cin >> file_name;
    if (fstream(file_name, ios::_Nocreate)) {
        cout << "\nСодержимое выбранного файла:\n\n";
        fileOutputBin(file_name);
    }
    else cout << "\nУказанного файла не существует.\n";
    break;
}
case 10: {
    cout << "Введите название бинарного файла: ";
    string file_name;
    cin >> file_name;

    if (fstream(file_name, ios::_Nocreate)) {
        cout << "Введите индекс искомого элемента: ";
        int index;
        cin >> index;
    }
}

```



```

        string res =fileGet(file_name, index)==" " ? "Не найдена":
fileGet(file_name, index);
        cout << "Запись с указанным индексом: " << res;
    }
    else cout << "\nУказанного файла не существует.\n";

    break;
}
case 11: {
    //void fileDelete(string file_name, int index);
    cout << "Введите название бинарного файла: ";
    string file_name;
    cin >> file_name;

    if (fstream(file_name, ios::_Nocreate)) {
        cout << "Введите индекс удаляемого элемента: ";
        int index;
        cin >> index;
        fileDelete(file_name, index);
        cout << "Запись успешно удалена.";
    }
    else cout << "\nУказанного файла не существует.\n";
    break;
}
case 12: {
    //void switchTI(string file_name, string number, string date);
    cout << "Введите название бинарного файла: ";
    string file_name;
    cin >> file_name;

    if (fstream(file_name, ios::_Nocreate)) {
        cout << "Введите номер нужного автомобиля: ";
        string number;
        cin >> number;
        cout << "Введите изменяемую дату: ";
        string date;
        cin >> date;
        switchTI(file_name, number, date);
        cout << "Дата успешно изменена.\n";
    }
    else cout << "\nУказанного файла не существует.\n";
    break;
}
case 13: {
    cout << "Введите название бинарного файла: ";
    string file_name;
    cin >> file_name;

    if (fstream(file_name, ios::_Nocreate)) {
        cout << "Введите код автомобиля: ";
        string car;
        cin >> car;
        cout << "Введите дату: ";
        string date;
        cin >> date;
        cout << "Сформированный список:\n";
        checkCars(file_name, car, date);
    }
    else cout << "\nУказанного файла не существует.\n";
    break;
}
case 0:
    return 0;
}
cout << "\n\n";

```

```
    main();  
}
```

Файл functions1.h (прототипы разработанных функций задания 1)

```
#pragma once  
#include <iostream>  
#include <string>  
using namespace std;  
  
void fileInput(string file_name, string data);  
void fileAppend(string file_name, string data);  
string fileOutput(string file_name);  
  
void filePasteNumbers(string file_name);  
void fileAppendString(string file_name);  
int fileReadNumber(string file_name, int index);  
int fileCountNumber(string file_name);  
  
void fileAppendArrays(string file_name_1, string file_name_2);
```

Файл functions1.cpp (тела разработанных функций задания 1)

```
#include <iostream>  
#include <fstream>  
#include <string>  
#include "random"  
using namespace std;  
  
//Функция записи в файл  
void fileInput(string file_name, string data) {  
    ofstream fout;  
    fout.open(file_name);  
  
    if (fout.is_open()) {  
        if (fout.good()) fout << data;  
    }  
  
    if (fout.good()) fout.close();  
}  
  
//Функция дозаписи в конец файла  
void fileAppend(string file_name, string data) {  
    ofstream fout;  
    fout.open(file_name, ios::app);  
  
    if (fout.is_open()) {  
        if (fout.good())  
            fout << data;  
    }  
  
    if (fout.good()) fout.close();  
}  
  
//Функция чтения файла  
string fileOutput(string file_name) {  
    ifstream fin;  
    fin.open(file_name);  
    char chr;  
    string res = "";  
    if (fin.is_open()) {  
        while (fin.get(chr) && fin.good()) {  
            res+=chr;  
        }  
    }  
}
```

```

        if (fin.eof())
            fin.close();

        return res;
    }
    //Функция заполнения файла десятичными числами
    void filePasteNumbers(string file_name){
        int strNumber = rand() % 11+1;
        string finalStr = "";

        for (int i = 0; i < strNumber; i++) {
            int digitsNumber = rand() % 11 + 1;
            for (int j = 0; j < digitsNumber; j++) {
                finalStr += to_string(rand() % 90+10);
                if (j != digitsNumber - 1) finalStr += " ";
            }
            if (i != strNumber - 1) finalStr += "\n";
        }
        fileInput(file_name, finalStr);
    }
    //Функция добавления строки в файл
    void fileAppendString(string file_name){
        string finalStr = "";
        int digitsNumber = rand() % 11 + 1;
        for (int j = 0; j < digitsNumber; j++) {
            finalStr += to_string(rand() % 90 + 10);
            if (j != digitsNumber - 1) finalStr += " ";
        }
        finalStr = "\n" + finalStr;

        fileAppend(file_name, finalStr);
    }
    //Функция чтения числа по его индексу
    int fileReadNumber(string file_name, int index){

        string finalStr=fileOutput(file_name);

        for (int i = 0; i < finalStr.length(); i++) {
            if (finalStr[i] == '\n') finalStr[i] = ' ';
        }

        string res = "";
        res += finalStr[3 * (index - 1)];
        res += finalStr[3 * (index - 1)+1];
        int num = atoi(res.c_str());

        return num;
    }
    //Функция подсчета количества чисел
    int fileCountNumber(string file_name){

        string finalStr = fileOutput(file_name);

        for (int i = 0; i < finalStr.length(); i++) {
            if (finalStr[i] == '\n') finalStr[i] = ' ';
        }
        int count = 0;
        for (int i = 0; i < finalStr.length(); i++) {
            if (finalStr[i] == ' ') count++;
        }
        return count+1;
    }
}

```

```

//Функция добавления массивов в конец файла
void fileAppendArrays(string file_name_1, string file_name_2) {
    string strToInp = fileOutput(file_name_1);
    string finalStr = strToInp;
    for (int i = 0; i < finalStr.length(); i++) {
        if (finalStr[i] == '\n') finalStr[i] = ' ';
    }

    string even = "";
    string odd = "";

    for (int i = 0; i < finalStr.length(); i+=3) {
        string s = "";
        s += finalStr[i];
        s += finalStr[i + 1];
        int num = atoi(s.c_str());

        if (num % 2 == 0) even = even + " " + s;
        else odd = odd + " " + s;
    }

    even = to_string(even.length() / 3) + even;
    odd = to_string(odd.length() / 3) + odd;

    strToInp = strToInp + "\n\n" + odd + "\n" + even;
    fileInput(file_name_2, strToInp);
}

```

Файл functions2.h (прототипы разработанных функций задания 2)

```

#pragma once
#include <string>
#include <iostream>
using namespace std;
void textToBin(string file_name_txt, string file_name_bin);
void binToText(string file_name_txt, string file_name_bin);
void fileOutputBin(string file_name);
string fileGet(string file_name, int index);
void fileDelete(string file_name, int index);
void switchTI(string file_name, string number, string date);
void checkCars(string file_name, string car, string date);

```

Файл functions2.cpp (тела разработанных функций задания 2)

```

#include "functions1.h"

#include <iostream>
#include <fstream>
#include <string>
using namespace std;
//Функция конвертации текстового файла в бинарный
void textToBin(string file_name_txt, string file_name_bin) {
    fstream fin = fstream(file_name_txt, ios::in);
    fstream fout = fstream(file_name_bin, ios::binary | ios::out);
    if (fin.is_open() && fout.is_open()) {

        fout << fin.rdbuf();

        if (fin.eof())
            fin.close();
        if (fout.fail() != ios::failbit)
            fout.close();
    }
}

```

```

    }
}
//Функция конвертации бинарного файла в текстовый
void binToText(string file_name_txt, string file_name_bin) {
    fstream fin = fstream(file_name_bin, ios::binary | ios::in);
    fstream fout = fstream(file_name_txt, ios::binary | ios::out);
    if (fin.is_open() && fout.is_open()) {

        fout << fin.rdbuf();

        if (fin.eof())
            fin.close();
        if (fout.fail() != ios::failbit)
            fout.close();
    }
}
//Функция вывода содержимого бинарного файла
void fileOutputBin(string file_name) {
    fstream fin(file_name, ios::binary | ios::in);

    if (fin.is_open()) cout << fin.rdbuf();

    if (fin.eof())
        fin.close();
}
//Функция получения записи по ее индексу
string fileGet(string file_name, int index) {
    ifstream fin(file_name, ios::in | ios::binary);
    string line = "";
    string str = "";
    if (fin.is_open()) {
        ifstream fin2(file_name, ios::in | ios::binary);
        while (fin2) {
            str += fin2.get();
        }
        fin2.close();

        if (index * 37 <= str.size() + 10) {
            fin.seekg(index * 37 * sizeof(char));
            getline(fin, line);
        }
    }
    if (fin.fail() != ios::failbit)
        fin.close();
    return line;
}
//Функция удаления записи по ее индексу
void fileDelete(string file_name, int index) {
    string str = fileOutput(file_name);
    string strToCopy="";
    for (int i = str.rfind("\n") + 1; i < str.length(); i++) {
        strToCopy += str[i];
    }
    for (int i = index * 37; i < index * 37 + strToCopy.length(); i++) {
        str[i] = strToCopy[i - index * 37];
    }

    str[index * 37+1] = index + '0';

    str=str.erase(str.rfind("\n"), strToCopy.length());
    fileInput(file_name, str);
}
//Функция замены даты техосмотра
void switchTI(string file_name, string number, string date){

```

```

fstream fin(file_name, ios::in | ios::binary | ios::out);
string num, mark, color, fam, no, data;
string hash;
string index="-1";
while (fin) {
    fin >> hash;
    fin >> num;
    fin >> mark;
    fin >> color;
    fin >> fam;
    fin >> no;
    fin >> data;
    if (number == num) { index = hash; break; }
}
if (index != "-1") {
    fin.seekp(atoi(index.c_str()) * 37);
    data = date;
    fin << hash << " ";
    fin << num << " ";
    fin << mark << " ";
    fin << color << " ";
    fin << fam << " ";
    fin << no << " ";
    fin << data;
}
if (fin.fail() != ios::failbit)
    fin.close();
}
//Функция вывода списка автомобилей определенной марки с определенной
//датой техосмотра
void checkCars(string file_name, string car, string date){
    fstream fin(file_name, ios::in | ios::binary);
    string num, mark, color, fam, no, data;
    string hash;
    while (!fin.eof()) {
        fin >> hash;
        fin >> num;
        fin >> mark;
        fin >> color;
        fin >> fam;
        fin >> no;
        fin >> data;
        if (car == mark && date == data)
            cout << hash << " " << num << " " << mark << " " << color << "
" << fam << " " << no << " " << data<<"\n";
    }
    if (fin.fail() != ios::failbit)
        fin.close();
}

```