# содержание

[содержание 3](#_Toc73669504)

[ВВЕДЕНИЕ 4](#_Toc73669505)

[1. Постановка задачи 5](#_Toc73669506)

[2. Разработка алгоритма 6](#_Toc73669507)

[3. Разработка программы 10](#_Toc73669508)

[3.1 Выбор среды разработки 10](#_Toc73669509)

[3.2. Спецификация программы 11](#_Toc73669510)

[3.3. Разработка модулей 13](#_Toc73669511)

[4. Тестирование программы 15](#_Toc73669512)

[4.1. Описание входных и выходных данных 15](#_Toc73669513)

[4.2. Результаты тестирования программы 16](#_Toc73669514)

[Заключение 31](#_Toc73669515)

[Список использованной литературы 32](#_Toc73669516)

**ПРИЛОЖЕНИЕ А ТЕКСТ ПРОГРАММЫ**

**ПРИЛОЖЕНИЕ Б ГРАФИЧЕСКИЙ МАТЕРИАЛ**

# ВВЕДЕНИЕ

Целью этой курсовой работы стала разработка программы, ориентированной на обработку информации о мировых рекордах в беге на 100 метров, данные о которых можно будет вводить как с клавиатуры, так и с текстового файла.

Используя данное ПО пользователю будет доступен инструмент по созданию и изменению базы данных мировых рекордов забегов на время. Основной функционал программы был написан в дополнительно подключаемой пользовательской библиотеке, что позволяет дополнить функционал программы в случае того если это потребуется.

Создание данной программы дало мне ценнейший опыт по разработке курсовых проектов и написании больших программ, в которых требуется разбитие на несколько модулей. Мною были получены новые знания о работе с текстовыми файлами, применении сортировок на практике, хранении информации в массивах структур, формировании меню выбора действий.

Программа позволяет реализовать спектр действий для работы с информацией, с помощью которого заинтересованные в этом пользователи смогут формировать таблицы с рекордами в беге на 100 метров, располагать рекорды по убыванию в шкале «результат», изменять записи в случае необходимости и многое другое.

Данная программа представляет собой мини-базу данных, в которой можно увидеть результат рекордсмена, его ФИО, год установления рекорда, возраст рекордсмена на момент установления рекорда, а также его пол.

# Постановка задачи

Задачей курсового проекта является написание программного обеспечения, которое позволяет работать с массивом структурированных данных «Мировые рекорды в беге 100 метров». Программа должна иметь следующий функционал:

1. Ввод информации из текстового файла в массив указателей на записи
2. Добавление новых элементов в конец массива
3. Просмотр всех элементов массива
4. Вывод информации из массива в текстовый файл
5. Корректировка полей выбранного элемента
6. Удаление выбранного элемента
7. Сортировка массива по числовому полю
8. Вставка нового элемента перед выбранным элементом
9. Вставка нового элемента перед выбранным элементом

В качестве среды разработки программы был использован продукт компании Microsoft - Visual Studio 2019. Языком программирования для написания программного обеспечения курсового проекта был выбран C++.

# Разработка алгоритма

Процесс разработки алгоритма программ и функций существенно усложняется, когда разработчик не придерживается определённого пути разработки с самого начала, не придерживается некоторой дисциплины, которая позволит ему на следующем этапе разработки чётко наметить шаги, цели, которые необходимо достичь для написания правильного решения программы. Метод пошаговой разработки как раз и является данной дисциплиной, которой пришлось придерживатся во время разработки решения данного курсового проекта. Сутью данного метода разработки является движение снизу-вверх, то есть достижение определённых подцелей, которые с каждым шагом всё более детализируют алгоритмы и приближают разработчика к полному построения программного обеспечения, которое решает поставленную проблему. Таким образом, получается последовательность все более детальных спецификаций алгоритма, приближающихся к окончательной версии программы.

Разработка алгоритма это одна из важнейших частей процесса написания программного решения какой-либо проблемы, ведь от выбора типов алгоритма и от тщательной проработки работы программы в теоретическом плане, будет зависеть быстродействие и удобство в работе с практической стороной программы.

В данном курсовом проекте областью исследования являются мировые рекорды в беге 100 метров. Пользователь имеет доступ к набору функций по созданию и изменению базы данных мировых рекордов забегов на время. Используя различные функции данной программы, пользователь будет обрабатывать информацию о рекордах в забеге на 100 метров.

Каждый рекорд будет представлен как элемент массива структур типа “Рекорд”. Поля в массиве структур содержат следующую информацию о рекорде:

* Результат рекорда;
* ФИО рекордсмена;
* Год установления рекорда;
* Возраст спортсмена;
* Пол.

При написании программы была использована модульность языка программирования С++. Модульность заключается в возможности помещать определённую часть функционала программы в подключаемую пользовательскую библиотеку, внутри которой также можно подключать другие стандартные, либо нестандартные (пользовательские), библиотеки Си. В данном случае, почти весь функционал программы написан в подключаемой библиотеке, что позволит расширить его в случае необходимости.

Во время запуска программы, её выполнение начинается с выполнения главной функции «main». Краткое описание главной функции программы: при запуске программы от пользователя потребуется ввести количество элементов массива, после чего начинается ввод каждого отдельного поля всех элементов по очереди. Вся информация будет записана в массив в отведённую для него динамическую память при помощи стандартных функций выделения памяти в C++.

После записи первых элементов выводится таблица всех элементов, где можно убедиться в правильности ввода. Также вместе с выводом таблицы результатов рекордов в забеге на 100 метров, выводится меню действий, где можно выбрать действие, которое необходимо выполнить над результатами, будь это удаление элемента, изменение, добавление, сортировка и т.д.

После выполнения какого-либо действия, которое изменяет результаты рекордов в забеге на 100 метров, таблица выводится ещё раз, после чего открывается меню, и таким образом будет происходить выполнение программы в цикле, пока пользователь не выберет пункт меню выхода из программы.

Функции, с помощью которых обрабатывается массив данных «мировые рекорды в беге 100 метров», описаны в отдельно подключаемой пользовательской библиотеке, заголовочный файл которой имеет название “kurs\_functions.h”. Краткое описание алгоритмов функций пользовательской библиотеки:

Алгоритм считывания файлов:

1. Ввод пути к файлу и названия файла;
2. Проверка на возможность открытия файла:
3. если открыть файл не удалось, функция открытия файла завершается;
4. если файл был открыт, то будет создан дополнительный массив для хранения в нём переменных такого же типа, как и в массиве структур «мировые рекорды в беге 100 метров», после чего во временный массив будут считаны данные в соответствующие поля структуры до конца файла, после чего временный массив будет удалён, а элементы из него помещены в оригинальный массив.

Алгоритм добавления элемента в конец массива:

1. Создание временного массива, размером на 1 больше основного, содержащего все те же элементы, что и в оригинальном массиве;
2. Ввод с клавиатуры полей нового элемента на последнее место во временном массиве;
3. Копирование временного массива в главный, с предшествующим этому увеличением кол-ва элементов на 1.

Алгоритм вывода данных на экран:

1. Создание цикла, который будет проходить столько раз, сколько элементов находится в нашем массиве структур «мировые рекорды в беге 100 метров».
2. Прохождение в цикле по каждому элементу массива структур, во время которого будет произведён вывод полей каждого отдельного элемента в окно консоли.

Алгоритм вывода информации в файл:

1. Открыть поток для записи в файл;
2. В цикле записывать поля каждого элемента в виде таблицы в файл с новой строки;
3. Закрыть поток.

Алгоритм корректировки полей элемента:

1. Ввод номера рекорда;
2. Ввод с клавиатуры значений полей для изменяемого элемента массива.

Алгоритм удаления рекорда из массива структур:

1. Создание временного массива, размером на 1 меньше основного;
2. Ввод с клавиатуры нового элемента на последнее место во временном массиве;
3. Копирование временного массива в главный, кроме удаляемого элемента, уменьшение кол-ва элементов.

Алгоритм вставки нового элемента перед выбранным элементом:

1. Ввод записи, перед которой следует вставить элемент;
2. Копирование элементов главного массива в новый, временный массив:
3. если элементы находятся до добавляемого, запись остаётся на месте;
4. если элементы находятся после добавленного, они передвигаются вправо на 1 (включая равный ему по позиции);
5. Ввод элемента на пустое место временного массива;
6. Копирование временного массива в главный, очистка временного массива.

Алгоритм вставки нового элемента после выбранного элемента:

1. Ввод записи, после которой следует вставить элемент;
2. Копирование элементов главного массива в новый, временный массив:
3. если элементы находятся до добавляемого, запись остаётся на месте (включая равный ему по позиции);
4. если элементы находятся после добавленного, они передвигаются вправо на 1;
5. Ввод элемента на пустое место временного массива;
6. Копирование временного массива в главный, очистка временного массива.

Алгоритм сортировки рекордов по числовому полю:

1. Создание временной переменной для хранения данных одной записи;
2. Устанавливается начальное значение минимального индекса;
3. В цикле происходит поиск минимального элемента, который потом становится на место первого, левая граница области поиска каждый проход перемещается вправо на 1, так до конца пока массив не будет отсортирован.

Алгоритм завершения работы программы:

1. Освобождение памяти, занимаемой массивом структур;
2. Закрытие программы.

# Разработка программы

## Выбор среды разработки

В качестве среды разработки программы был использован продукт компании Microsoft - Visual Studio 2019. Visual Studio имеет множество самых разных функций и возможностей, необходимых при написании программы. Эта программа является средой разработки проффесионального уровня, имеет полный набор инструментов для всех этапов разработки, от начального замысла до финального развертывания, а также является бесплатной при условии написания некомерчесских проектов, именно поэтому была выбрана она.

Сама программа написана на языке программирования высокого уровня С++, так как он является безопасным, имеется возможность писать обобщённый код с помощью шаблонов, также имеется возможность использовать объектно-ориентированный подход, возможно упрощение кода за счёт перегрузки функций и операторов, более простая обработка ошибок за счёт исключений (что в конечном итоге приводит к уменьшению времени, требуемого для написания кода).

Все перечисленные выше преимущества относятся именно к языку С++, что и стало причиной моего выбора данного языка программирования как основного при написании данного курсового проекта.

В данном программном обеспечении были также использованы другие подключаемые библиотеки, содержащие функции, упрощающие работу по разработке программы. Они были подключены в самой пользовательской библиотеке с заголовочным файлом «kurs\_functions.h», ведь для работы простого меню не требуется большой функционал. Используемыми в данном курсовом проекте по обработке массива «мировые рекорды в беге 100 метров» являются следующие заголовочные файлы:

1. IOSTREAM – заголовочный файл с классами, функциями и переменными для организации ввода-вывода в языке программирования C++. Он включён в стандартную библиотеку C++. Название образовано от Input/Output Stream («поток ввода-вывода»).;
2. IOMANIP – Для использования манипуляторов с параметрами в программу необходимо включить заголовочный файл iomanip.h. Манипуляторы могут использоваться в составе выражений ввода/вывода.;
3. String.h, string – эти заголовочные файлы производят организацию работы со строками. String.h - Заголовочный файл стандартной библиотеки языка Си ++, содержащий функции для работы с нуль-терминированными строками и различными функциями работы с памятью;
4. Stdio.h, windows – данные заголовочные файлы использованы для произведения установки кодировок и языка консоли чтобы отображалась кириллица. Stdio.h - заголовочный файл, содержащий определения макросов, константы и объявления функций и типов, используемых для различных операций стандартного ввода и вывода;
5. FSTREAM – заголовочный файл из стандартной библиотеки C++, включающий набор классов, методов и функций, которые предоставляют интерфейс для чтения/записи данных из/в файл;
6. Algorithm – организация замены нижних подчёркиваний на пробелы при записи из файла в массив.

## 3.2. Спецификация программы

Программа выводит основную информацию о рекордах в виде таблицы, где каждому элементу соответствует отдельная строка, каждому полю элемента соответствует один столбик.

Информация в файле для открытия распологается таким образом: самым первым числом будет количество элементов массива в файле, после чего начинается перечисление полей элементов (результат-автор-год-возраст-пол).

Программа реализована в виде “дерева”, то есть в ней имеется меню, которое можно вызвать для работы с функциями, файлами и структурой. Краткое описание работы функций, которые можно вызвать с помощью выбора конкретных пунктов меню:

Функция void Openfile необходима для заполнения массива в программе с помощью открытия файла и чтения данных массива из этого файла. При вызове функции вводится полный путь к файлу указывая название файла и расширение .txt.

Функция void Addelement необходима для добавления новых элементов в конец массива вводя их с клавиатуры. При вызове функции с клавиатуры вводятся поля нового элемента, который добавляется в конец основного массива.

Функция void Outputall необходима для вывода информации о рекордах в виде таблицы со строками и столбцами, содержащими элементы и поля элементов массива.

Функция void Createfile необходима для вывода информации в файл в виде таблицы со строками и столбцами, содержащими элементы и поля элементов массива.

Функция void Editline необходима для изменения полей некоторого выбранного в этой функции с помощью клавиатуры элемента. При вызове происходит запрос номера записи, которую требуется изменить и происходит заполнение полей элемента с клавиатуры с заменой старых данных элемента.

Функция void Deleteelement необходима для удаления элемента массива. При вызове функции происходит ввод номера элемента, который требуется удалить, после чего элементы массива сдвигаются влево если они стоят после элемента, который удаляется.

Функция void Addbefore необходима для добавления нового элемента после какого-либо выбранного с помощью клавиатуры элемента. При вызове функции происходит ввод номера элемента, после чего элементы, стоящие после номера выбранной записи, сдвигаются на 1 шаг вправо, а на пустое место записывается новый элемент.

Функция void Addafter необходима для добавления нового элемента перед каким-либо выбранным с помощью клавиатуры элемента. При вызове функции происходит ввод номера элемента, после чего элементы, стоящие после номера выбранной записи, сдвигаются на 1 шаг вправо, включая выбранный элемент, а на пустое место записывается новый элемент.

Функция void Sortall необходима для сортировки массива по числовому полю. Массив сортируется по полю результата забега, с наименьшим результатом на первом месте, потом идут результаты по возрастанию времени забега.

Функции float getValue\_float, int getValue\_int, int getValue\_ArraySize, int getValue\_pol являются дополнительными функциями, необходимые замены стандартных методов ввода информации с клавиатуры cin. В этих функциях реализована проверка правильности ввода различных типов данных, необходимы для предотвращения получения ошибок от программы.

Функция void print\_menu необходима для вывода меню выбора действий после выполнения каждой функции.

## 3.3. Разработка модулей

При написании программного обеспечения используя высокоуровневый язык программирования С++, самые простые программы могут состоять из одной функции main и содержать малое количество действий, функций, и тому подобное.

Чуть более сложные программы, обрабатывающие большее количество информации и данных включают в себя другие функции, а также другое количество этих функций. По мере возрастания сложности и длины кода программы функций становится слишком много, что может привести к такой ситуации, когда в них уже становится слишком тяжело ориентироваться обычному программисту.

Выходом из данной, не совсем приятной, ситуации является разбиение функций на отдельные модули по их смысловому значению. А если написанием программы занимается несколько человек в небольшой команде, то разбиение функций на отдельные модули неизбежно.

Модульное программирование — это организация программы как совокупности небольших независимых блоков, называемых модулями, структура и поведение которых подчиняются определённым правилам.

Удобство использования модульной архитектуры заключается в возможности обновления (замены) модуля, без необходимости изменения остальной системы. То есть при использовании модулей имеется возможность дополнения функционала программы в будущем. Модуль — функционально законченный фрагмент программы. Во многих языках (но далеко не обязательно) оформляется в виде отдельного файла с исходным кодом или поименованной непрерывной её части. Некоторые языки предусматривают объединение модулей в пакеты.

Каждый отдельный программный модуль (библиотека) может быть разработан отдельно от других модулей. Он может представлять собой самостоятельный продукт и быть включенным в структуру разных программ, даже не обязательно для тех, которых находился в разработке. Такие свойства модульной системы и самих модулей показывает, что разработка библиотек (модулей) ПО призвана решать не только задачу упрощения программирования, но и борьбы с дублированием – никакого смысла нет в разработке того, что уже создано и может быть использовано в программе точно так же хорошо, как и другая функция идентичная по её действию.

В данной программе по обработке массива «мировые рекорды в беге 100 метров» все функции, которые вызываются с помощью меню действий, расположены в дополнительном пользовательском модуле, заголовочный файл которо называется «kurs\_functions.h». Также заголовочный файл «kurs\_functions.h» подключает другие дополнительные стандартные функции языка Си++ для корректного функционирования пользовательского модуля, такие как iostream, iomanip, string, fstream и др.

Главная функция программы «main» описана в основном файле программы, который имеет название source.cpp. Выполнение программы начинается именно с функции main, через которую уже вызываются все дополнительные функции.

# Тестирование программы

## 4.1. Описание входных и выходных данных

Первоначальными входными данными при запуске программы являются данные введённые с клавиатуры. После ввода первоначальных данных уже возможна их дальнейшая обработка, а также после ввода первоначальных данных имеется возможность чтения заранее записанных данных из подготовленного для этого файла, находящегося на носителе/жёстком диске компьютера, путём вызова функции открытия файла из меню программы (функция вызывается через выбор пользователем с клавиатуры пункта меню под номером 1).

Для полноценного успешного открытия файла с записанными в него данными, информация в файле должна располагаться строго определённым образом, а также в определённом порядке, иначе могут возникнуть ошибки при попытке чтения и в память массива будет записано нечитаемое человеком значение. Чтобы этого избежать, следует соблюдать следующий порядок записи информации в файл:

1. Самым первым считываемым значением в файле выступает число количества элементов массива «мировые рекорды в беге 100 метров», количество так называемых «рекордов», которое следует считывать последовательно из этого файла.
2. Далее после числа элементов массива «мировые рекорды в беге 100 метров» идёт информация о самих элементах, поля одного элемента размещаются на одной отдельной строке, а каждый новый элемент начинается новой строкой. В каждой строке содержится один элемент, поля с информацией которого расположены следующим образом:

А) результат забега (число типа float),

Б) ФИО спортсмена (латинским алфавитом, вместо пробелов используется нижнее подчёркивание, которое позже после обработки функциями программы, описанными в дополнительно подключаемой пользовательской библиотеке, заменится на пробел),

В) год установки рекорда (целочисленное значение, число типа int, максимум 5 знаков),

Г) возраст рекордсмена (целочисленное значение, число типа int, максимум 3 знака),

Д) Пол (вводится число либо 1, либо 2, при вызове функции вывода информации на экран заменяются на «мужской» при значении 1, либо на «женский» при значении 2).

При правильном написании полного пути к файлу, полного имени файла, а также расширения txt, программа выведет на экран сообщение о успешном считывании информации из файла, после чего можно будет начать полноценную обработку информации массива «мировые рекорды в беге 100 метров» с помощью функций программы.

Пример правильного написания входного файла с вводными данными для записи в массив указателей указан ниже на рис. 4.1.

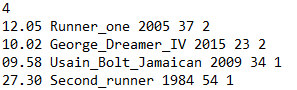


Рис. 4.1 – пример данных в файле для ввода (input.txt)

## 4.2. Результаты тестирования программы

Тестирование программы было произведено на персональном компьютере, имеющем следующие характеристики:

1. Процессор - AMD Athlon(tm) II X3 450 Processor 3.21 GHz.
2. Устройство обработки видео – NVIDIA GeForce GT710.
3. Количество оперативной памяти - 8,00 ГБ.
4. Количество видеопамяти – 2,00 ГБ.

В качестве операционной системы на компьютере, выбранном для тестирования программы по обработке массива «мировые рекорды в беге 100 метров» установлена 64-разрядная операционная система Windows.

В самом начале работы с программой первое что выводится на экран и видит пользователь данного программного обеспечения, это номер варианта выданного задания курсовой работы, название обрабатываемого массива, а также выводится информация о требовании произвести ввод количества первоначальных записей массива «мировые рекорды в беге 100 метров». При запуске программы от пользователя требуется ввести количество элементов массива, после чего начинается ввод каждого отдельного поля всех элементов по очереди

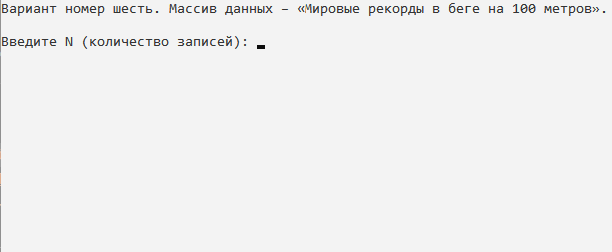


Рис. 4.2 – Первоначальный ввод

После ввода количества записей массива «мировые рекорды в беге 100 метров» начинается ввод самих полей записей.

Ввод информации о каждом элементе данного массива производится в таком же порядке, как при написании файла, из которого происходит чтение информации. Данный порядок является таковым:

1. вводится результат забега;
2. вводится ФИО рекордсмена;
3. вводятся данные о годе установки рекорда;
4. следует ввести 1, либо 2, для выбора пола рекордсмена (1 – мужской, 2 - женский).

Данный порядок ввода можно увидеть на рис 4.3.

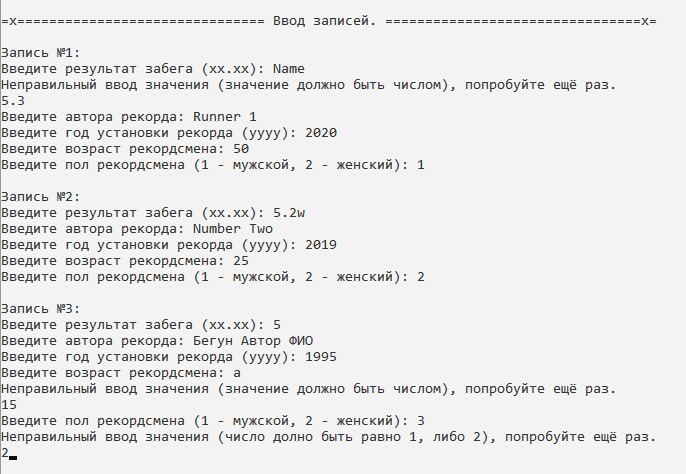


Рис. 4.3 – Ввод первоначальных записей

После того как ввод первоначальной информации о каждом поле всех элементов закончен, открывается меню для выбора различных действий над элементами массива «мировые рекорды в беге 100 метров», а также выводится таблица со всеми введёнными записями и их полями.

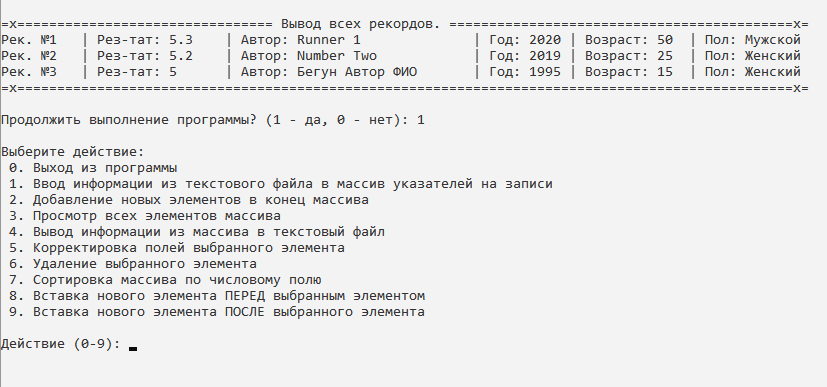


Рис. 4.4 – Меню и таблица рекордов

При выборе первого пункта меню выводится текст, в котором содержится просьба ввода пути и названия файла для открытия. В данном примере открывается файл input.txt, содержимое которого можно увидеть на листе 13.



Рис. 4.5.1 – Выбор первого пункта меню

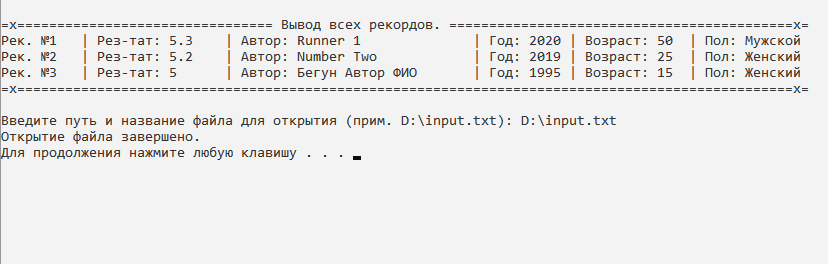


Рис. 4.5.2 – Ввод пути к файлу для открытия

После успешного открытия файла, на экран будет выведено сообщение об успешном открытии файла.

Информация из открытого файла будет записана в массив данных «мировые рекорды в беге 100 метров», на таблице информация будет заменена на информацию, записанную в файл.

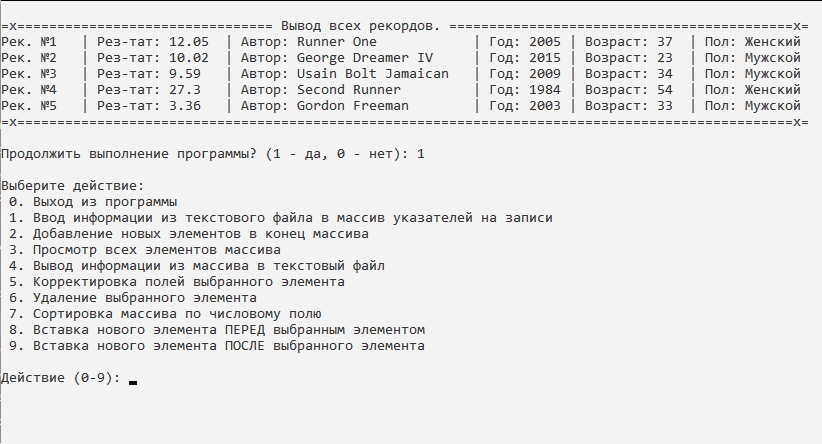


Рис. 4.6 – Файл открыт

При выборе пункта 2 (добавление элемента в конец массива) открывается ввод элемента в новую строку таблицы.



Рис. 4.7.1 – Выбор второго пункта меню

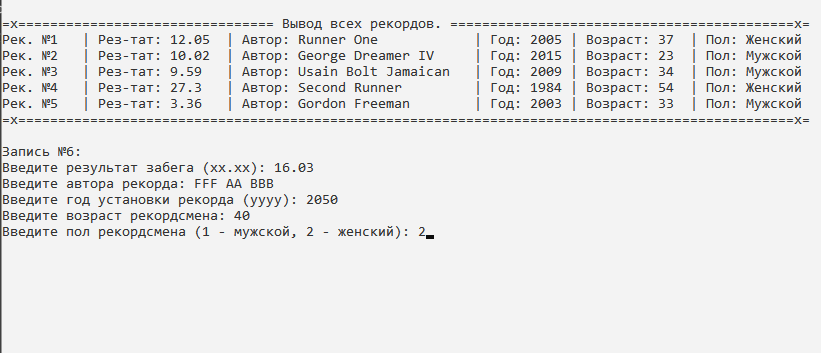


Рис 4.7.2 – Ввод нового элемента

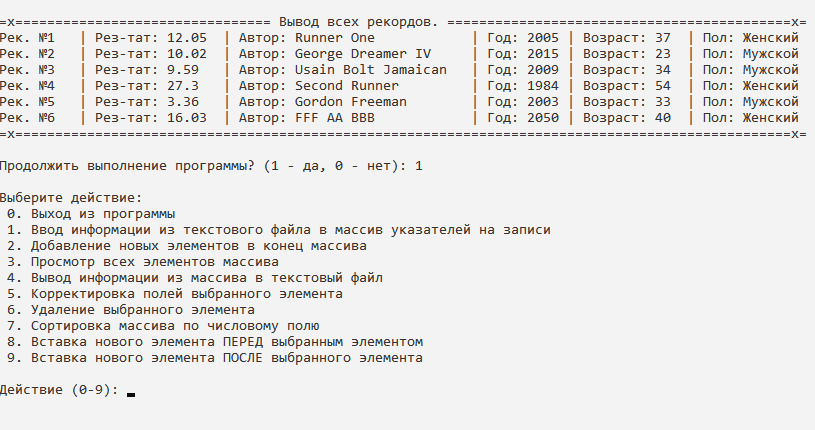


Рис. 4.7.3 – На последней строке таблицы отображается добавленная запись

При выборе пункта меню 3 происходит вывод точно такой же таблицы, как и после изменения какого-либо элемента.

Выводится вся информация о каждом элементе массива «мировые рекорды в беге 100 метров» содержащаяся в полях элементов массива.



Рис. 4.8.1 – Выбор третьего пункта меню

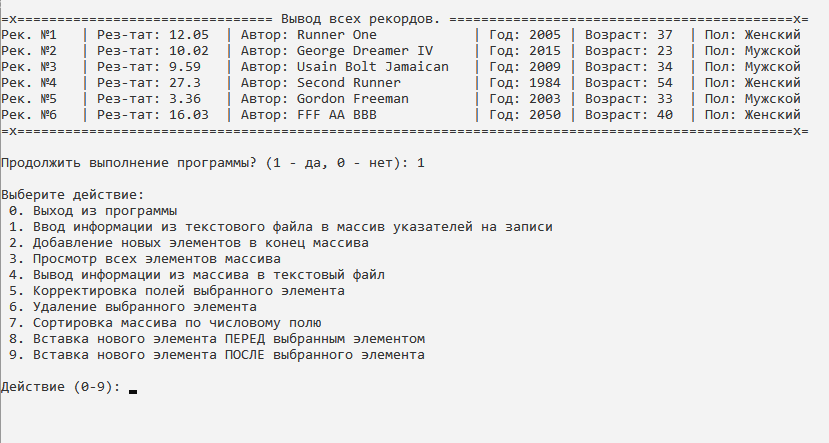


Рис. 4.8.2 – Вывод информации на экран

При выборе четвёртого пункта меню происходит ввод пути и названия файла для вывода, в который выведется таблица с информацией о рекордах.



Рис. 4.8.1 – Выбор четвёртого пункта меню

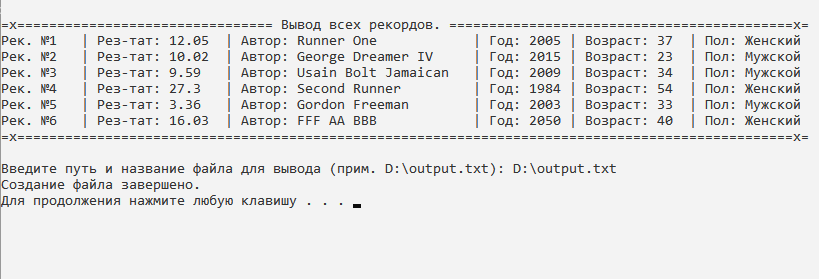


Рис. 4.8.2 – Ввод пути к файлу и успешное создание

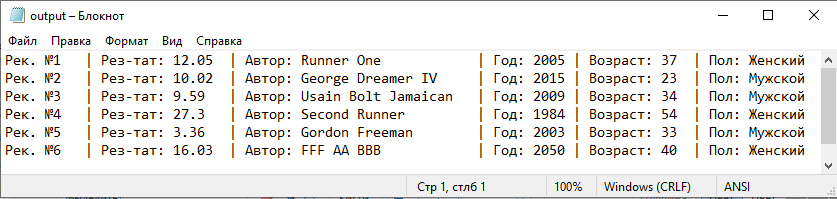


Рис. 4.8.3 – Содержимое созданного файла

При выборе пятого пункта меню будет произведён ввод номера записи, поля которой пользователь желает отредактировать и начнётся ввод информации.



Рис. 4.9.1 – Выбор пятого пункта меню

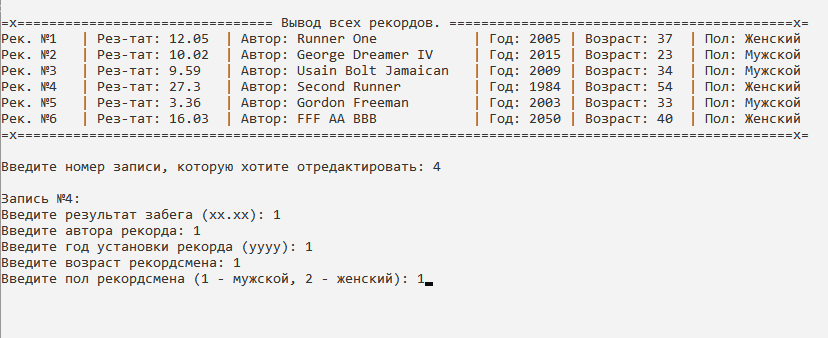


Рис. 4.9.2 – Редактирование полей с информацией

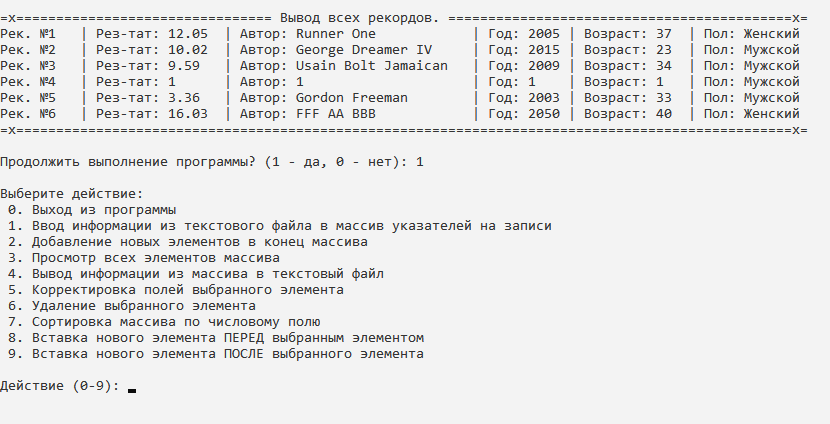


Рис. 4.9.3 – Отредактированное поле

При выборе пункта меню 6 будет выполнен ввод номера записи, которую требуется удалить, после чего запись исчезнет из таблицы.



Рис. 4.10.1 – Выбор шестого пункта меню

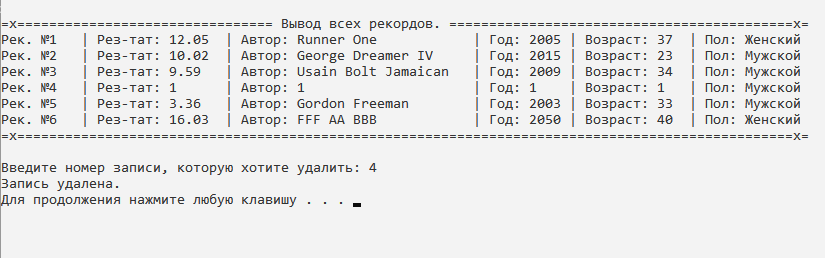


Рис. 4.10.2 – Выбор 4 записи на удаление

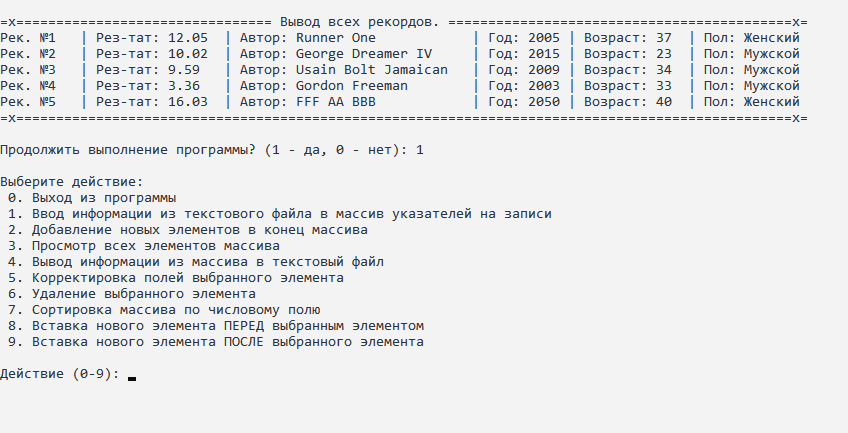


Рис. 4.10.3 – Запись 4 исчезла из таблицы, отображение меню

После выбора седьмого пункта будет произведена сортировка массива по числовому полю. В данном случае это поле значения результата забега.



Рис. 4.11.1 - Выбор седьмого пункта меню

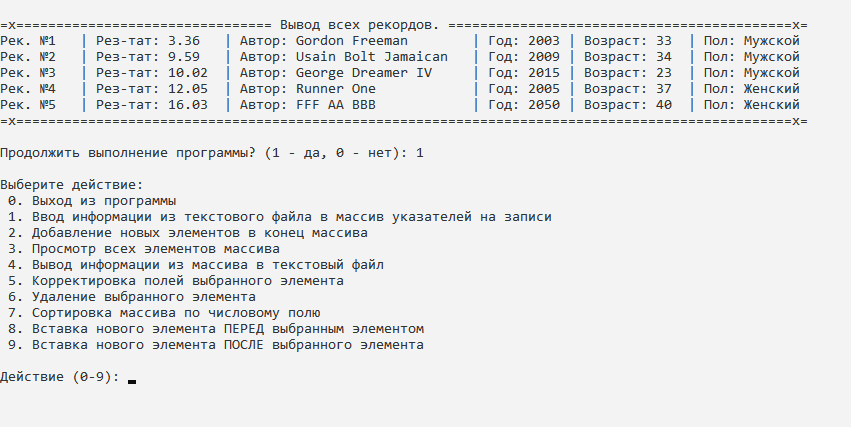


Рис. 4.11.2 – Результаты поменялись местами из-за сортировки

При выборе 8 пункта меню будет произведён выбор записи перед которой будет вставлен новый элемент массива, а также пользователь введёт сами поля элемента.



Рис. 4.12.1 – Выбор восьмого пункта меню

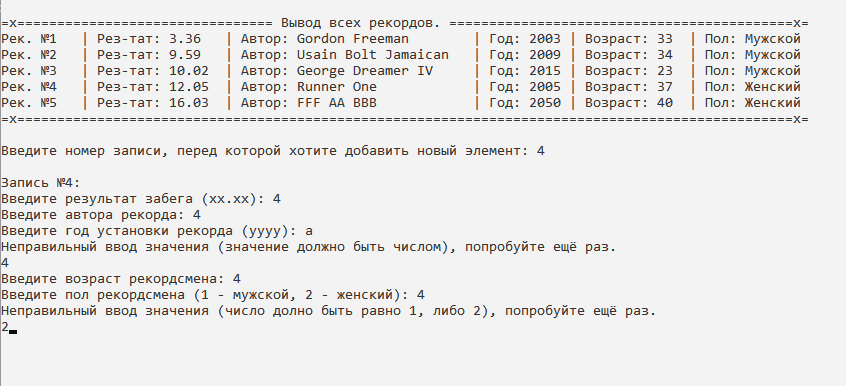


Рис. 4.12.2 – Ввод нового элемента

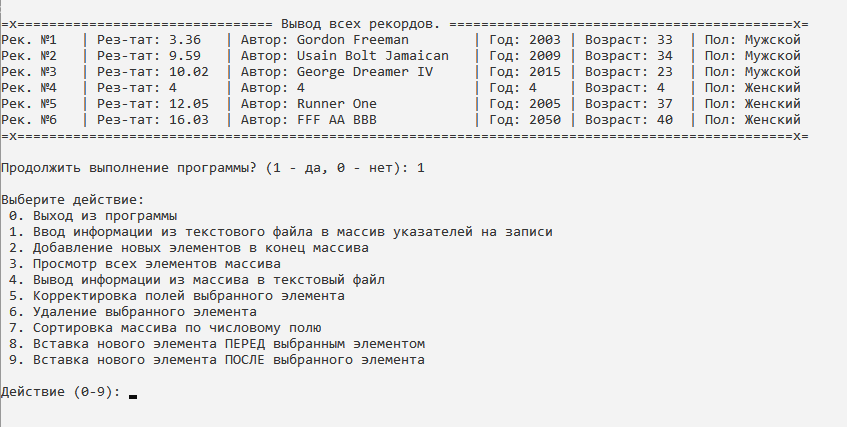


Рис. 4.12.3 – Добавление перед 4 записью нового элемента

При выборе пункта меню 9 будет введён номер записи, после которой будет добавлен новый элемент, а также будет произведён номер самих полей новой записи.



Рис. 4.13.1 – Выбор девятого пункта меню

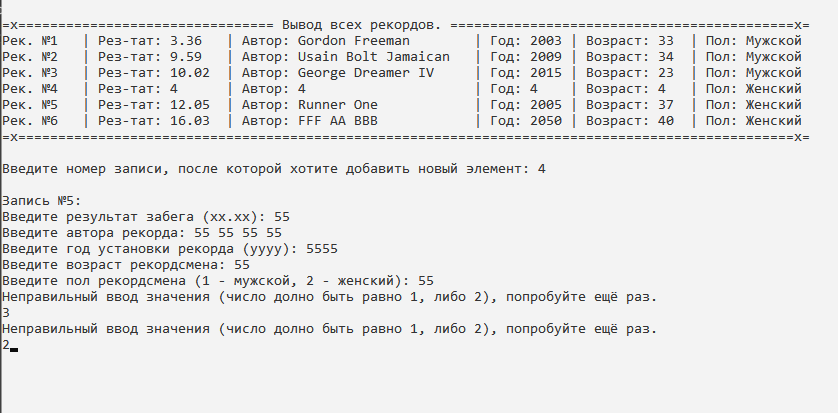


Рис. 4.13.1 – Меню ввода новой записи после выбранной

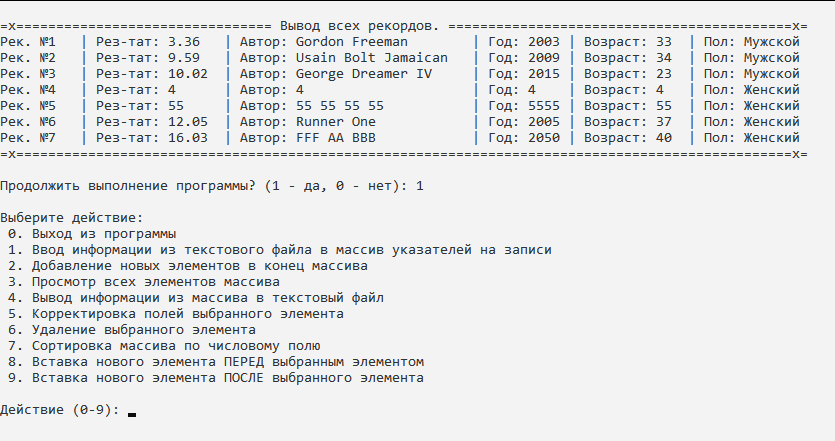


Рис. 4.13.2 – После четвёртой записи появилась новая запись, которая содержит введённую ранее информацию в своих полях

При выборе нулевого пункта меню программа прекращает выполнение каких – либо действий, очищает память от массива и закрывает консоль.



Рис. 4.14.1 – Выбор нулевого пункта меню, после чего происходит закрытие программы

# Заключение

В результате работы над курсовым проектом была написана программа, обрабатывающая массив структур «Мировые рекорды в беге 100 метров». Программа работает корректно и все нужные и указанные выше функции выполняет без ошибок. Из данного курсового проекта мною был получен ценный опыт разработки больших программ, была написана полезная программа, которую можно применить на практике для упрощения работы над большим чем обычно объёмом информации.

# Список использованной литературы

1. Павловская, Т.А. С/ С++. Программирование на языке высокого уровня / Т.А. Павловская. – СПб.: Питер, 2011. – 461 с.
2. Сидорина, Т. Самоучитель Microsoft Visual Studio C++ и MFC / Т. Сидорина. – Издательство: BHV, 2009. – 848 с.
3. Пахомов, Б. C/C++ и MS Visual C++ 2012 для начинающих / Б. Пахомов. - Москва: СИНТЕГ, 2015. - 518 c.