

Министерство образования Республики Беларусь
Учреждение образования «Белорусский государственный университет
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Дисциплина: Программирование на языках высокого уровня

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовому проекту
на тему

СВЕДЕНИЯ ОБ ИГРОКАХ ФУТБОЛЬНОЙ КОМАНДЫ

БГУИР КП 1-40 02 01 123 ПЗ

Студент: гр. 130501 Феденко Р. Ю.

Руководитель: Ковальчук А.М.

Минск 2022

Учреждение образования
«Белорусский государственный университет информатики
и радиоэлектроники»

Военный факультет

УТВЕРЖДАЮ
Заведующий кафедрой

(подпись)

2022 г.

ЗАДАНИЕ
по курсовому проектированию

Студенту Феденко Роману Юрьевичу

1. Тема проекта Сведения об игроках футбольной команды
2. Срок сдачи студентом законченного проекта 15 декабря 2022г.
3. Исходные данные к проекту текстовые файлы, в которых содержится информация: Член команды (Имя, Фамилия, Возраст), Футболист (Имя, Фамилия, Возраст, Игровой номер), Тренер (Имя, Фамилия, Возраст, Стаж работы), Защитник (Имя, Фамилия, Возраст, Игровой номер, Количество желтых карточек), Нападающий (Имя, Фамилия, Возраст, Игровой номер, Количество голов, Количество голевых передач), Тренер по Физической подготовке (Имя, Фамилия, Возраст, Стаж работы, Вид тренировки), Тренер вратарей (Имя, Фамилия, Возраст, Стаж работы, Время тренировки вратарей); Контейнер-Итератор: linklist, стандартные контейнеры: list, vector.
4. Содержание расчетно-пояснительной записки (перечень вопросов, которые подлежат разработке)
 1. Задание.
 2. Введение.
 3. Обзор литературы.
 - 3.1. Обзор методов и алгоритмов решения поставленной задачи.
 4. Функциональное проектирование.

4.1. Структура входных и выходных данных.

4.2. Разработка диаграммы классов.

4.3. Описание классов.

5. Разработка программных модулей.

5.1. Разработка схем алгоритмов(два наиболее важных метода).

5.2. Разработка алгоритмов (описание алгоритмов по шагам, для двух методов).

6. Результаты работы.

7. Заключение

8. Литература

9. Приложения

5. Перечень графического материала (с точным обозначением обязательных чертежей и графиков)

1. Диаграмма классов.

2. Схема алгоритма метода:

template<typename TYPE>

TYPE Algorithm<TYPE>::search1(Node<TYPE>* beg, int c)

3. Схема алгоритма метода:

template<class TYPE>

void Algorithm<TYPE>::sort(linklist<TYPE>& a)

6. Консультант по проекту (с обозначением разделов проекта) А.М. Ковальчук

7. Дата выдачи задания 15.09.2022 г.

8. Календарный график работы над проектом на весь период проектирования (с обозначением сроков выполнения и трудоемкости отдельных этапов):

1. Выбор задания. Разработка содержания пояснительной записки. Перечень графического материала – 15 %;

разделы 2, 3 – 10 %;

разделы 4 к – 20 %;

разделы 5 к – 35 %;

раздел 6,7,8 – 5 %;

раздел 9 к – 5%;

оформление пояснительной записки и графического материала к 15.12.22 – 10 %

Защита курсового проекта с 21.12 по 28.12.22г.

РУКОВОДИТЕЛЬ А.М. Ковальчук

(подпись)

Задание принял к исполнению Р. Ю. Феденко

(дата и подпись студента)

СОДЕРЖАНИЕ

1. ОБЗОР ЛИТЕРАТУРЫ.....	7
2. СТРУКТУРНОЕ ПРОЕКТИРОВАНИЕ	9
2.1 Блок приложения	9
2.1.1 Блок выбора типа объекта	10
2.1.2 Блок выбора метода для работы с объектом	10
2.1.3 Блок работы с STL	10
3. ФУНКЦИОНАЛЬНОЕ ПРОЕКТИРОВАНИЕ.....	11
3.1 Текстовый файл.....	11
3.1.1 Файл Fis_Trener.txt	12
3.1.2 Файл Vera_Trener.txt.....	12
3.1.3 Файл Defender.txt	12
3.1.4 Файл Forward.txt	13
3.1.5 Файл Trener.txt	13
3.1.6 Файл Footballist.txt	13
3.1.7 Файл Chlen_komandi.txt.....	13
3.2 Диаграмма классов	14
4.РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ.....	15
4.1 Схемы алгоритмов	16
5. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ.....	17
6. СПИСОК ЛИТЕРАТУРЫ	26
ЗАКЛЮЧЕНИЕ	27
ПРИЛОЖЕНИЕ А.....	28
ПРИЛОЖЕНИЕ Б	29
ПРИЛОЖЕНИЕ В.....	30
ПРИЛОЖЕНИЕ Г	31
ПРИЛОЖЕНИЕ Д.....	32

ВВЕДЕНИЕ

В данном курсовом проекте предполагается реализовать программное обеспечение по управлению библиотекой при помощи инструментальной среды программного обеспечения. Для реализации этой задачи я использовал язык программирования C++ который был разработан в начале 1980-х годов сотрудником фирмы Bell Labs Бьёрном Страуструпом.

C++ — это язык с C-подобным синтаксисом.

Страуструп решил дополнить язык с возможностями, имеющимися в языке Си. Из этого языка были позаимствованы возможности объектно-ориентированного программирования (возможность работы с классами и объектами). К 1983 году в язык были добавлены новые возможности, такие как виртуальные функции, перегрузка функций и операторов, ссылки, константы, пользовательский контроль над управлением свободной памятью, улучшенная проверка типов и новый стиль комментариев (//). Получившийся язык уже перестал быть просто дополненной версией классического C (“C с классами”) и был переименован в «C++». Его первый коммерческий выпуск состоялся в октябре 1985 года.

C++ уже давно поддерживает много полезных функций:

- инкапсуляция,
- наследование,
- полиморфизм,
- перегрузка операторов,
- статическая типизация.

При этом он всё ещё активно развивается, и с каждой новой версией появляется всё больше интересного — например лямбды, динамическое связывание, асинхронные методы и т.д.

Когда говорят C++, нередко имеют в виду технологии платформы .NET. И наоборот, когда говорят .NET, нередко имеют в виду C++. Однако, хотя эти понятия связаны, отождествлять их неверно. Язык C++ был создан специально для работы с фреймворком .NET, однако само понятие .NET несколько шире.

Когда-то Билл Гейтс сказал, что .NET — это лучшее, что сделала компания Microsoft. У него есть весомые основания так считать. Фреймворк .NET представляет мощную платформу для создания приложений.

Вот его киллер-фичи:

- Поддержка нескольких языков. В основе .NET — общезыковая среда исполнения Common Language Runtime (CLR), благодаря чему платформа поддерживает несколько языков: наряду с C# это VB.NET, C++, F#, а также различные диалекты других языков, привязанные к

.NET, например, Delphi.NET. Код на любом из этих языков компилируется в сборку на общем языке CIL (Common Intermediate Language) — своего рода ассемблер платформы .NET. Поэтому можно сделать отдельные модули одного приложения на разных языках.

- Мощная библиотека классов. .NET представляет единую для всех поддерживаемых языков библиотеку классов. Какое бы приложение мы ни собирались писать на C++ — текстовый редактор, чат или сложный веб-сайт — так или иначе мы задействуем библиотеку классов .NET.
- Разнообразие технологий. Общезыковая среда исполнения CLR и базовая библиотека классов — это основа для целого стека технологий, которые разработчики могут задействовать при создании разных приложений. Например, для баз данных в этом стеке имеется технология ADO.NET и Entity Framework Core. .NET долгое время развивался под названием .NET Framework — преимущественно как платформа для Windows. Но с 2019 она больше не развивается — последней версией этой платформы стала .NET Framework 4.8.

В 2014 Microsoft начал выпускать альтернативную платформу - .NET Core, которая должна была вобрать в себя все возможности устаревшего .NET Framework и добавить новую функциональность. Поэтому следует различать .NET Framework, который предназначен преимущественно для Windows, и кроссплатформенный .NET Core.

Существует множество способов для написания графических приложений Windows Forms, Qt Creator и т.д. В данном курсовом проекте используется одна из самых популярных IDE – WPF.

1. ОБЗОР ЛИТЕРАТУРЫ

Ниже представлены основные термины в программировании, так как без понимания их невозможно будет выучить ни один из языков программирования, а также проектирование ПО:

Программирование – процесс и искусство создания компьютерных программ с помощью языков программирования. В узком смысле слова, программирование рассматривается как кодирование – реализация одного или нескольких взаимосвязанных алгоритмов на некотором языке программирования. В более широком смысле, программирование – процесс создания программ, то есть разработка программного обеспечения. Большая часть работы программиста связана с написанием исходного кода на одном из языков программирования.

Программа – данные, предназначенные для управления конкретными компонентами системы обработки информации в целях реализации определенного алгоритма.

Программное обеспечение – совокупность программ системы обработки информации и программных документов, необходимых для эксплуатации этих программ.

Разработка программного обеспечения – это проектирование, написание, тестирование и поддержка компьютерных программ с целью решения задач для множества пользователей; это создание надежных защищенных решений, которые выдержат испытание временем и справятся с некоторыми не известными заранее задачами, лежащими в области, близкой к очевидным исходным задачам.

Программный модуль – программа или функционально заверченный фрагмент программы, предназначенный для хранения, трансляции, объединения с другими программными модулями для загрузки в оперативную память.

Объектно-ориентированное программирование – метод построения программ как совокупностей объектов и классов объектов, которые могут вызывать друг друга.

Компиляция – трансляция программы с языка высокого уровня в форму, близкую к программе на машинном языке.

Абстракция — мысленное отвлечение, обособление от тех или иных сторон, свойств или связей предметов и явлений для выявления существенных их признаков.

Диаграмма классов – структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их коопераций, атрибутов (полей), методов, интерфейсов и взаимосвязей между ними.

Целью создания диаграммы классов является графическое представление статической структуры декларативных элементов системы (классов, типов и т. п.) Она содержит в себе также некоторые элементы

поведения (например — операции), однако их динамика должна быть отражена на диаграммах других видов (диаграммах коммуникации, диаграммах состояний). Для удобства восприятия диаграмму классов можно также дополнить представлением пакетов, включая вложенные^[2].

При представлении сущностей реального мира разработчику требуется отразить их текущее состояние, их поведение и их взаимные отношения. На каждом этапе осуществляется абстрагирование от маловажных деталей и концепций, которые не относятся к реальности (производительность, инкапсуляция, видимость и т. п.). Классы можно рассматривать с позиции различных уровней. Как правило, их выделяют три основных: аналитический уровень, уровень проектирования и уровень реализации:

- на уровне анализа класс содержит в себе только набросок общих контуров системы и работает как логическая концепция предметной области или программного продукта.
- на уровне проектирования класс отражает основные проектные решения касательно распределения информации и планируемой функциональности, объединяя в себе сведения о состоянии и операциях.
- на уровне реализации класс дорабатывается до такого вида, в каком он максимально удобен для воплощения в выбранной среде разработки; при этом не воспрещается опустить в нём те общие свойства, которые не применяются на выбранном языке программирования.

1.1 Постановка задачи

Темой данного курсового проекта является «Сведения об игроках футбольной команды».

Данная система содержит программное обеспечение для персонального компьютера.

Цели курсового проектирования: овладеть практическими навыками проектирования и разработки законченного, отлаженного и протестированного программного продукта с использованием языка высокого уровня C++;

Минимальные требования к реализации программного продукта:

- хранение пользовательских данных
- запись, удаление и изменение информации
- простой и понятный интерфейс.

Задачами курсового проекта (работы) как этапа подготовки к дипломному проектированию являются:

- освоение, углубление и обобщение знаний, полученных студентами в процессе обучения;
- приобретение практических навыков и развитие творческих подходов к решению конкретной инженерной или инженерно-экономической задачи;

– формирование умений использовать справочную литературу, нормативную, правовую, нормативно-техническую документацию, осуществлять патентный поиск;

– приобретение навыков по оформлению текстовой и графической документации согласно требованиям государственных стандартов и стандарта предприятия. Дипломные проекты (работы).

2. СТРУКТУРНОЕ ПРОЕКТИРОВАНИЕ

Основной функцией компьютера является обработка информации. Программная обработка данных на компьютере реализуется следующим образом:

1. После запуска на выполнение программы, хранящейся во внешней долговременной памяти, она загружается в оперативную память.

2. Процессор последовательно считывает команды программы и выполняет их.

3. Необходимые для выполнения команды данные загружаются из внешней памяти в оперативную и над ними производятся необходимые операции. Данные, полученные в процессе выполнения команды, записываются процессором обратно в оперативную или внешнюю память.

4. В процессе выполнения программы процессор может запрашивать данные с устройств ввода информации и пересылать данные на устройства вывода информации.

Программная конфигурация ПК многоуровневая. Это связано с тем, что требования к программам, предназначенным для работы с устройствами, существенно отличаются от требований к программам, предназначенным для работы с людьми. Общий принцип такой: чем ниже уровень программ, тем больше они работают с устройствами и меньше с человеком. Этот принцип соблюдается во всей компьютерной технике от отдельного ПК до всемирной компьютерной сети Интернет.

Для разработки больших и сложных программ программисту необходимо овладеть специальными приемами получения рациональной структуры программы, которая обеспечивает почти двукратное сокращение объема программирования. Подчиненность модулей программы отражается в схеме, представленной в приложении А. Однако последняя не отражает порядок их вызова или функционирование программы.

2.1 Блок приложения

В приложении можно выделить следующие структурные блоки:

- блок выбора типа объекта;
- блок выбора метода для работы с объектом;
- блок работы с STL;

2.1.1 Блок выбора типа объекта

Данный блок предоставляет пользователю возможность выбора типа объекта для работы с ним в программе. Блок имеет удобный пользовательский интерфейс, в котором находится 7 пунктов. В первоначальном окне пользователь может выбрать тип объекта, также может выйти из программы.

2.1.2 Блок выбора метода для работы с объектом

Данный блок предоставляет пользователю, выбор метода, для работы с объектами. Блок имеет удобный пользовательский интерфейс, в котором находится 16 пунктов. В окне пользователь может выбрать метод, для работы с объектами, также может вернуться к блоку выбора типа объекта.

2.1.3 Блок работы с STL

Данный блок предоставляет пользователю возможность выбора метода для работы с STL. Блок имеет удобный пользовательский интерфейс, в котором находится 10 пунктов. В окне пользователь может выбрать метод, для работы с STL, также может вернуться к блоку выбора метода для работы с объектом.

3. ФУНКЦИОНАЛЬНОЕ ПРОЕКТИРОВАНИЕ

3.1 Текстовый файл

Текстовый файл — это файл, компонентами которого являются символьные строки переменной длины, заканчивающиеся специальным маркером конца строки.

Преимущества:

- Минимальный объём файла (при малом количестве текстовых данных).
- Универсальность — текстовый файл может быть прочитан (так или иначе) на любой системе или ОС, особенно если речь идёт об однобайтных кодировках вроде ASCII — они не подвержены многим проблемам, характерным для других форматов файлов — для них не важна разница в порядке байтов или длине машинного слова на разных платформах.
- Устойчивость — каждое слово и символ в таком файле самодостаточны, и если случится повреждение такого файла, то в этом случае обычно легче восстановить данные и продолжить обработку остального содержимого — повреждение любого из байтов такого файла никак не сказывается на сохранности остальных, в отличие от или сжатых или двоичных (напр. *.doc). Многие распространённые системы управления версиями (например в MS Word) рассчитаны на текстовые файлы, и с двоичными файлами могут работать только как с единым целым — при повреждении даже нескольких байтов такой файл восстановлению не подлежит.

Для работы с файлами я использовал классы `ofstream` и `ifstream`.

В таблице 1.1 представлены команды используемые в приложении для работы с файлами.

Таблица 1.1 – Используемые методы `fromFile` и `toFile`, `checkCount`, `clear`.

Тип метода	Метод	Описание метода
Методы доступа к файлу	<code>open()</code>	Этим методом можно открыть заданный файл, сопоставив его с одним из объектов потока. В зависимости от передаваемых аргументов, файл может быть открыт для чтения, для записи (либо для

Тип метода	Метод	Описание метода
		полной, либо для добавления данных), как бинарный, или как текстовый файл.
	<code>close()</code>	Функция закрывает файл, то есть прекращает доступ к нему, таким образом освобождая его для других функций или программ.

3.1.1 Файл Fis_Trener.txt

В файле Fis_Trener.txt находится основная информация о тренерах команды по физической подготовки. Более подробно рассмотрим поля, который хранятся в данном файле:

- поле `name`. Является именем тренера футбольной команды;
- поле `surname`. Является фамилией тренера футбольной команды;
- поле `year`. Содержит возраст тренера футбольной команды;
- поле `staj_work`. Содержит стаж работы тренера;
- поле `vid_treni`. Содержит основной вид тренировки, которую тренер проводит с игроками;

3.1.2 Файл Vera_Trener.txt

В файле Vera_Trener.txt находится основная информация о тренерах вратарей команды. Более подробно рассмотрим поля, который хранятся в данном файле:

- поле `name`. Является именем тренера футбольной команды;
- поле `surname`. Является фамилией тренера футбольной команды;
- поле `year`. Содержит возраст тренера футбольной команды;
- поле `staj_work`. Содержит стаж работы тренера;
- поле `vremia_trener`. Содержит время тренировки, которую тренер проводит с вратарями;

3.1.3 Файл Defender.txt

В файле Defender.txt находится основная информация о тренерах команды по физической подготовки. Более подробно рассмотрим поля, который хранятся в данном файле:

- поле `name`. Является именем игрока футбольной команды;
- поле `surname`. Является фамилией игрока футбольной команды;
- поле `year`. Содержит возраст игрока футбольной команды;
- поле `number`. Содержит номер игрока футбольной команды;
- поле `yellow_card`. Содержит количество желтых карточек у игрока;

3.1.4 Файл Forward.txt

В файле Forward.txt находится основная информация о тренерах вратарей команды. Более подробно рассмотрим поля, который хранятся в данном файле:

- поле name. Является именем игрока футбольной команды;
- поле surname. Является фамилией игрока футбольной команды;
- поле year. Содержит возраст игрока футбольной команды;
- поле number. Содержит номер игрока футбольной команды;
- поле goals. Содержит количество голов у игрока;
- поле assists. Содержит количество голевых передач у игрока;

3.1.5 Файл Trener.txt

В файле Trener.txt находится основная информация о тренерах команды по физической подготовки. Более подробно рассмотрим поля, который хранятся в данном файле:

- поле name. Является именем тренера футбольной команды;
- поле surname. Является фамилией тренера футбольной команды;
- поле year. Содержит возраст тренера футбольной команды;
- поле staj_work. Содержит стаж работы тренера;

3.1.6 Файл Footballist.txt

В файле Footballist.txt находится основная информация о тренерах вратарей команды. Более подробно рассмотрим поля, который хранятся в данном файле:

- поле name. Является именем игрока футбольной команды;
- поле surname. Является фамилией игрока футбольной команды;
- поле year. Содержит возраст игрока футбольной команды;
- поле number. Содержит номер игрока футбольной команды;

3.1.7 Файл Chlen_komandi.txt

В файле Chlen_komandi.txt находится основная информация о тренерах вратарей команды. Более подробно рассмотрим поля, который хранятся в данном файле:

- поле name. Является именем члена футбольной команды;
- поле surname. Является фамилией члена футбольной команды;
- поле year. Содержит возраст члена футбольной команды;

3.2 Диаграмма классов

Диаграмма классов представлена в приложении А

4.РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ

Программа начинает свою работу с созданием объекта класса и вызова метода запуска главного меню программы.

```
Fut_Team obj;  
obj.Start();
```

Во всем коде программы встречаются методы, с помощью которых производится работа с проектом.

```
template<typename TYPE>  
TYPE Algorithm<TYPE>::search1(Node<TYPE>* beg, int c)  
{  
    TYPE temp;  
    Node <TYPE>* rab = beg;  
    while (c > 1)  
    {  
        rab = rab->next;  
        c--;  
    }  
    cout << rab->data;  
    temp = rab->data;  
    return temp;  
}
```

Выше показан код при котором производится поиск члена команды по индексу.

Так же в программе реализован метод сортировки шаблона:

```
template<class TYPE>  
void Algorithm<TYPE>::sort(linklist<TYPE>& _a)  
{  
    Node<TYPE>* tmp = new Node<TYPE>;  
    Iterator<TYPE> it_1 = _a.Begin();  
    Iterator<TYPE> it_2 = _a.Begin()->next;  
    while (it_1 != NULL)  
    {  
        it_2 = it_1;  
        ++it_2;  
        while (it_2 != NULL)  
        {  
            if ((*it_1) > (*it_2))  
            {  
                tmp->data = *it_1;  
                *it_1 = *it_2;  
                *it_2 = tmp->data;  
            }  
            ++it_2;  
        }  
    }  
}
```

```

        ++it_1;
    }
}

```

4.1 Схемы алгоритмов

Схема алгоритма метода

```

template<typename TYPE>
TYPE Algorithm<TYPE>::search1(Node<TYPE>* beg, int c)

```

приведена в приложении Б – метод поиск объекта по индексу.

Схема алгоритма метода

```

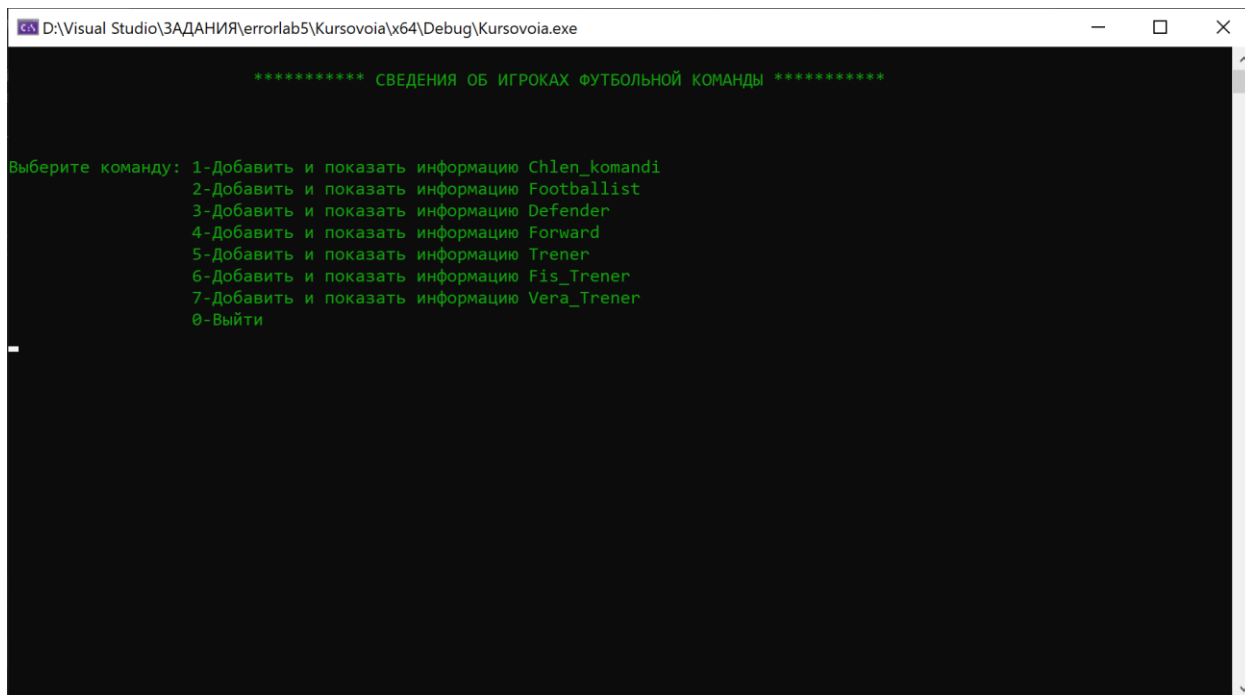
template<class TYPE>
void Algorithm<TYPE>::sort(linklist<TYPE>& _a)

```

приведена в приложении Б – метод сортировки шаблона.

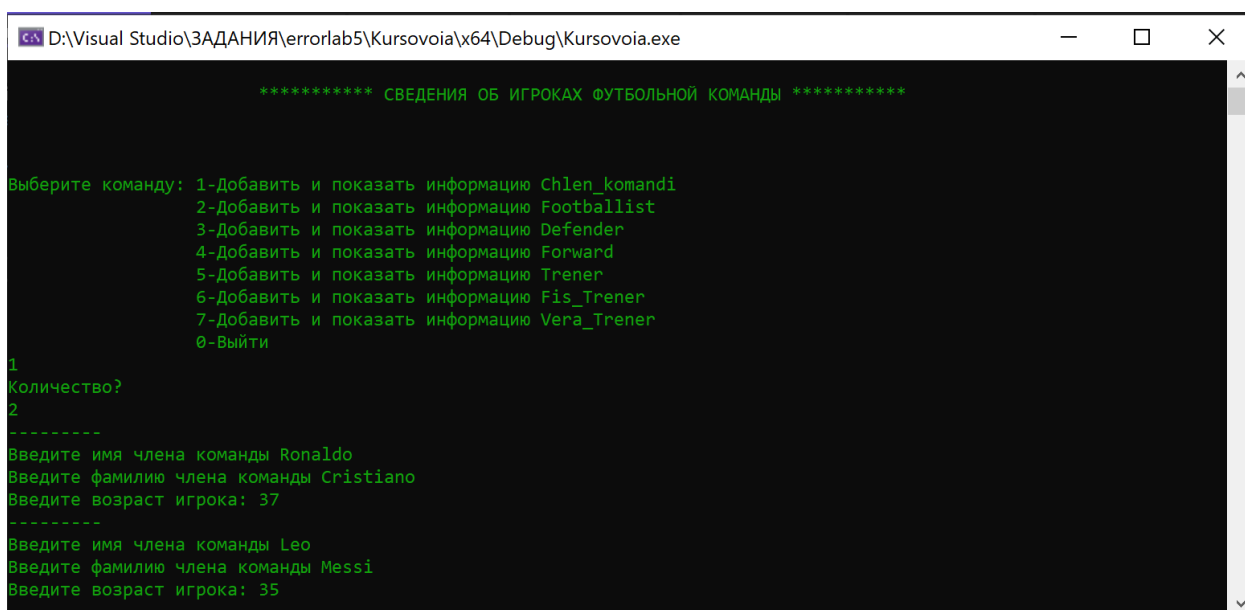
5. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

При запуске программы открывается окно



Где мы наблюдаем выбор, пользователь может работать с информацией про члена команды, с информацией про футболиста, с информацией про защитника, с информацией про нападающего, с информацией про тренера, с информацией про тренера по физической подготовке, с информацией про тренера вратарей. Также пользователь может закрыть программу

При выборе номера “1” открывается окно показанный ниже на рисунке.



Где пользователь вводит количество объектов и после чего вводит информацию про члена команды.

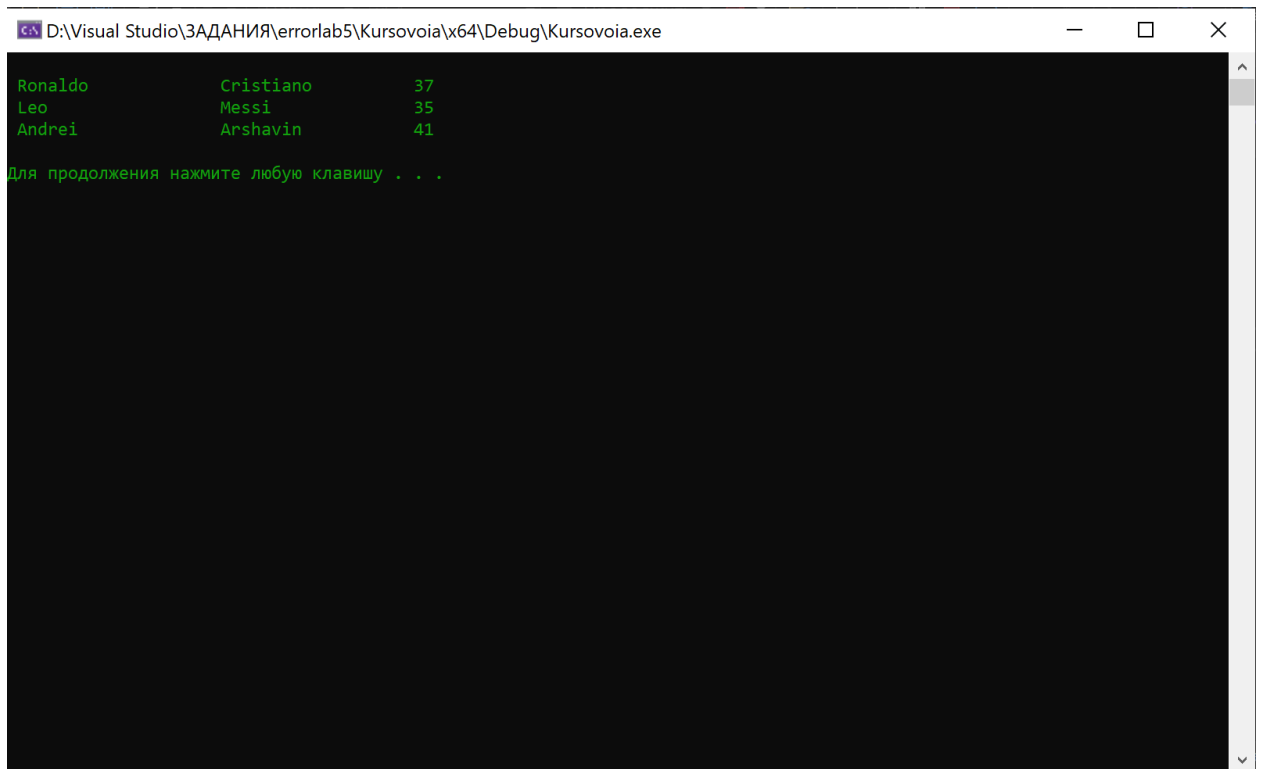
При успешной вводе информации мы можем наблюдать окно, показанный ниже на рисунке.

```
-----  
NAME          SURNAME        YEAR  
-----  
Ronaldo       Cristiano     37  
Leo           Messi        35  
-----  
Выберите тип: 1-Добавить в шаблон(в любую позицию)  
              2-Вывести шаблон  
              3-Поиск по названию  
              4-Поиск по индексу  
              5-Удаление объекта  
              6-Отмена удаления  
              7-Отмена добавления  
              8-Прочитать из текстового файла  
              9-Записать в текстовый файл  
             10-Очистить текстовый файл  
             11-показать в обратном порядке(с итератором)  
             12-показать(с итератором)  
             13-поиск по названию(с итератором)  
             14-сортировка  
             15-STL  
             16-выход
```

В данном окне мы можем выбрать один из 16 пунктов. При выборе пункта “1” мы можем добавить в любую позицию шаблона. Также пользователь может закрыть программу.

```
-----  
NAME          SURNAME        YEAR  
-----  
Ronaldo       Cristiano     37  
Leo           Messi        35  
Andrei        Arshavin     41  
-----  
Введите имя члена команды Andrei  
Введите фамилию члена команды Arshavin  
Введите возраст игрока: 41  
Введите позицию для добавления  
3  
-----  
Ronaldo       Cristiano     37  
Leo           Messi        35  
Andrei        Arshavin     41  
-----  
Для продолжения нажмите любую клавишу . . .
```

При выборе пункта “2” мы можем вывести шаблон на экран (рисунок ниже).

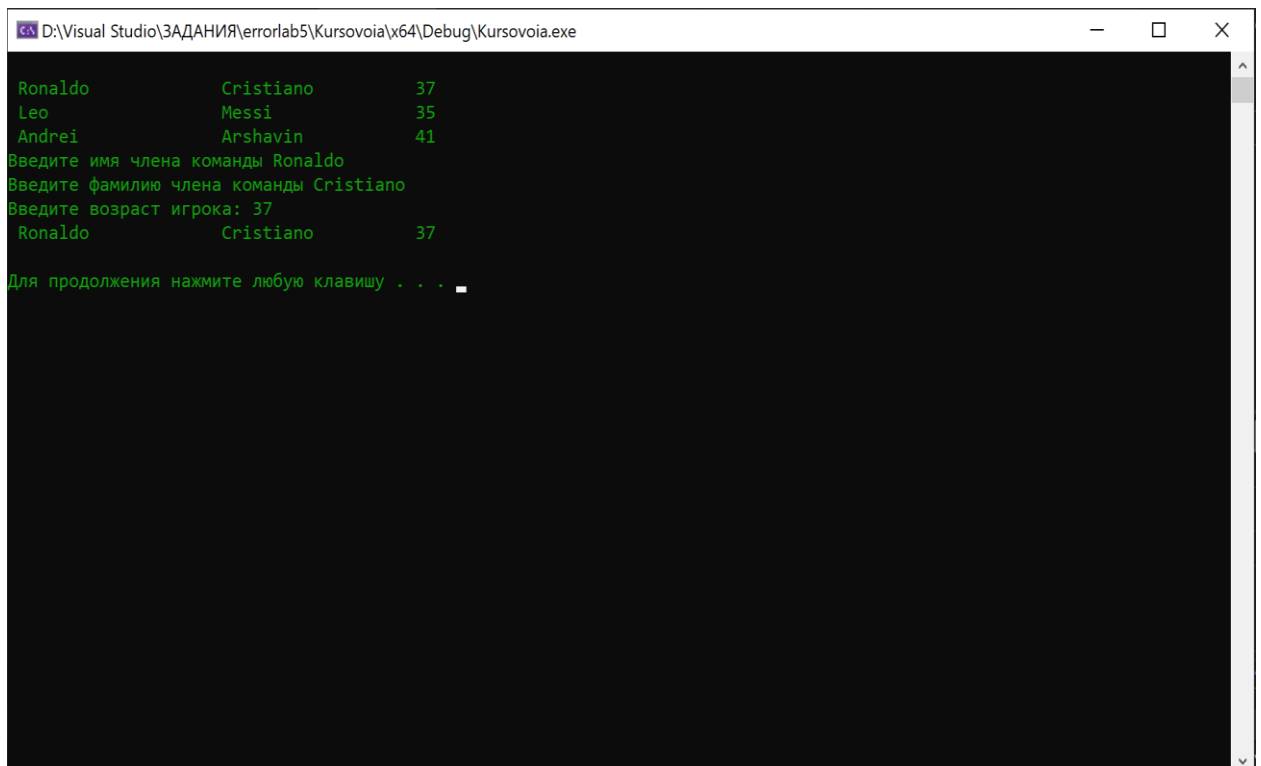


```
D:\Visual Studio\ЗАДАНИЯ\errorlab5\Kursovoia\x64\Debug\Kursovoia.exe

Ronaldo      Cristiano    37
Leo          Messi        35
Andrei       Arshavin     41

Для продолжения нажмите любую клавишу . . .
```

При выборе пункта “3” мы можем выполнить поиск по имени (рисунок ниже), если такой объект находится в шаблоне, он выводится на экран.

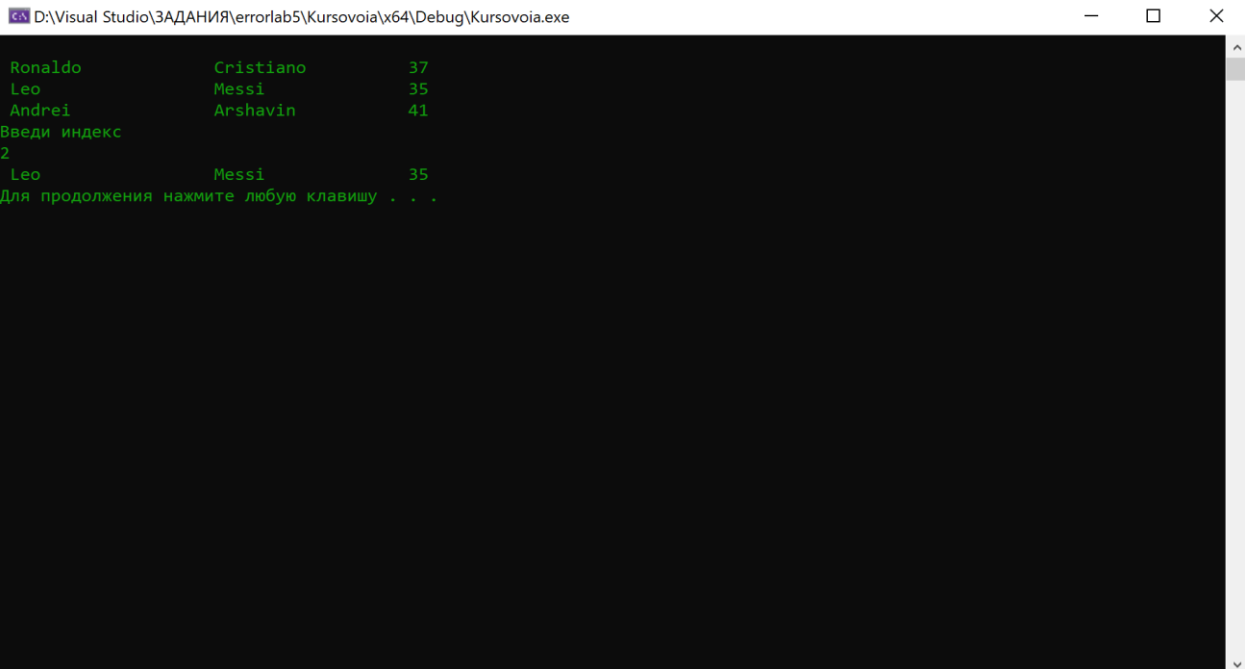


```
D:\Visual Studio\ЗАДАНИЯ\errorlab5\Kursovoia\x64\Debug\Kursovoia.exe

Ronaldo      Cristiano    37
Leo          Messi        35
Andrei       Arshavin     41
Введите имя члена команды Ronaldo
Введите фамилию члена команды Cristiano
Введите возраст игрока: 37
Ronaldo      Cristiano    37

Для продолжения нажмите любую клавишу . . .
```

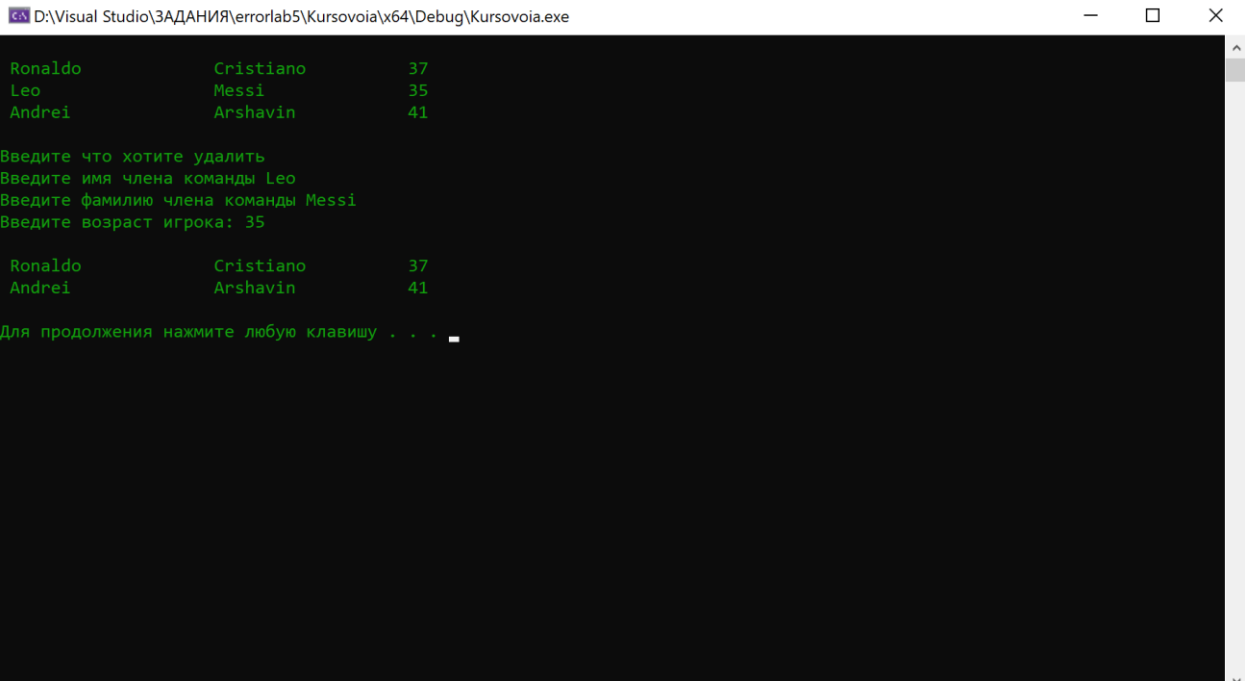
При выборе пункта “4” мы можем выполнить поиск по индексу (рисунок ниже), если такой индекс находится в шаблоне, выводится объект на экран.



```
D:\Visual Studio\ЗАДАНИЯ\errorlab5\Kursovoia\x64\Debug\Kursovoia.exe

Ronaldo      Cristiano    37
Leo          Messi        35
Andrei       Arshavin     41
Введи индекс
2
Leo          Messi        35
Для продолжения нажмите любую клавишу . . .
```

При выборе пункта “5” мы можем выполнить удаление объекта (рисунок ниже), если такой объект находится в шаблоне, он удаляется и на экран выводит измененный шаблон.

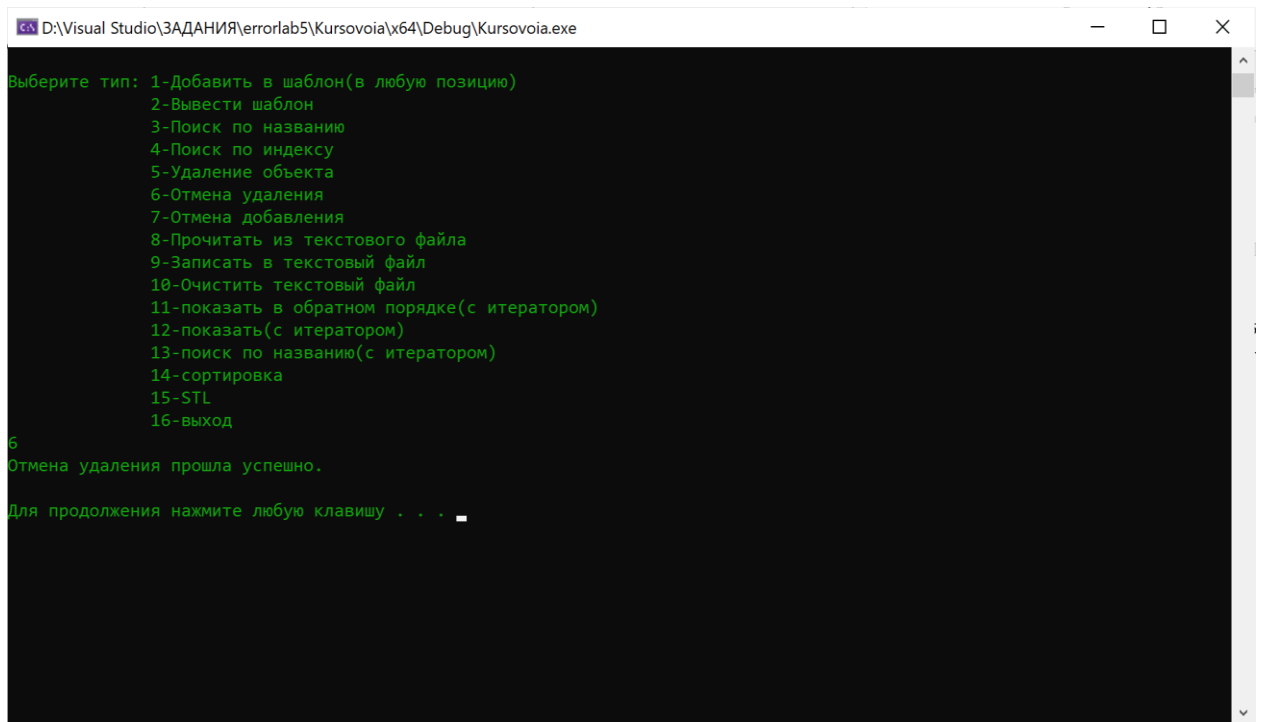


```
D:\Visual Studio\ЗАДАНИЯ\errorlab5\Kursovoia\x64\Debug\Kursovoia.exe

Ronaldo      Cristiano    37
Leo          Messi        35
Andrei       Arshavin     41
Введите что хотите удалить
Введите имя члена команды Leo
Введите фамилию члена команды Messi
Введите возраст игрока: 35

Ronaldo      Cristiano    37
Andrei       Arshavin     41
Для продолжения нажмите любую клавишу . . .
```

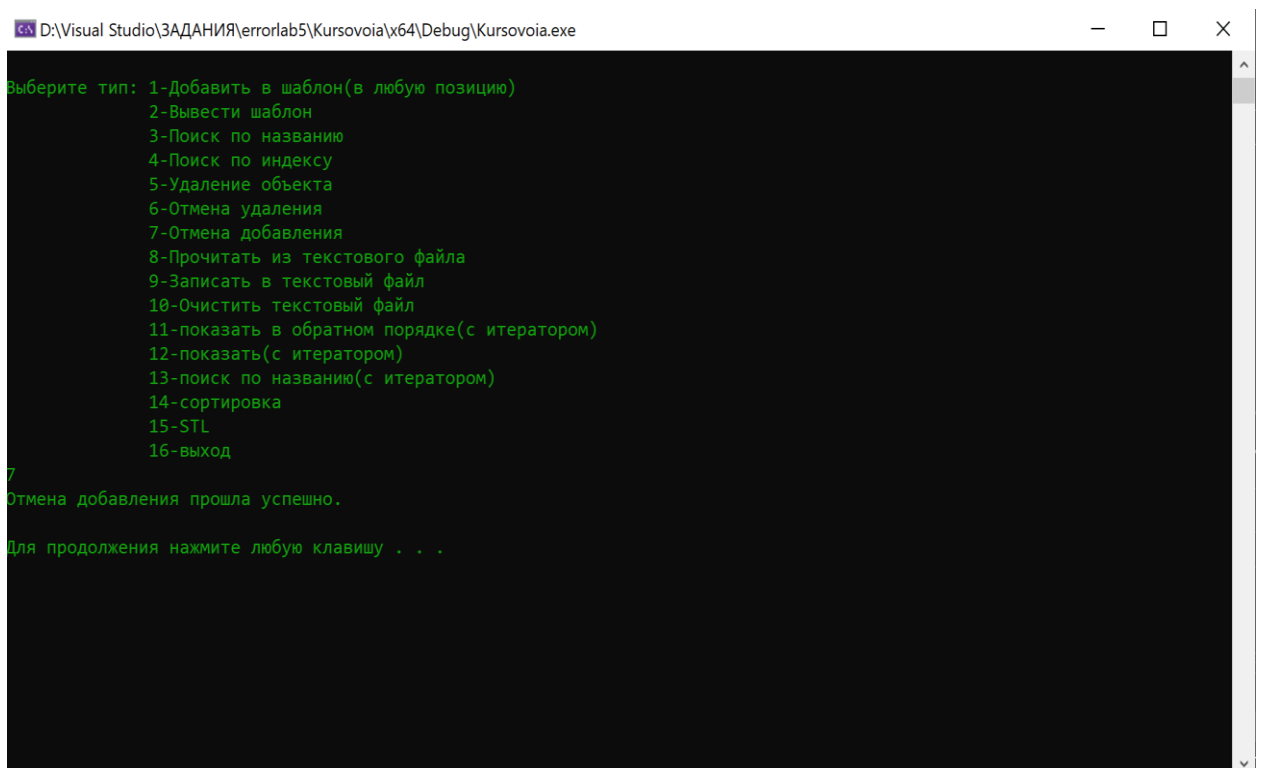
При выборе пункта “6” мы можем выполнить отмену удаления объекта (рисунок ниже).



```
D:\Visual Studio\ЗАДАНИЯ\errorlab5\Kursovoia\x64\Debug\Kursovoia.exe
Выберите тип: 1-Добавить в шаблон(в любую позицию)
              2-Вывести шаблон
              3-Поиск по названию
              4-Поиск по индексу
              5-Удаление объекта
              6-Отмена удаления
              7-Отмена добавления
              8-Прочитать из текстового файла
              9-Записать в текстовый файл
              10-Очистить текстовый файл
              11-показать в обратном порядке(с итератором)
              12-показать(с итератором)
              13-поиск по названию(с итератором)
              14-сортировка
              15-STL
              16-выход
6
Отмена удаления прошла успешно.

Для продолжения нажмите любую клавишу . . .
```

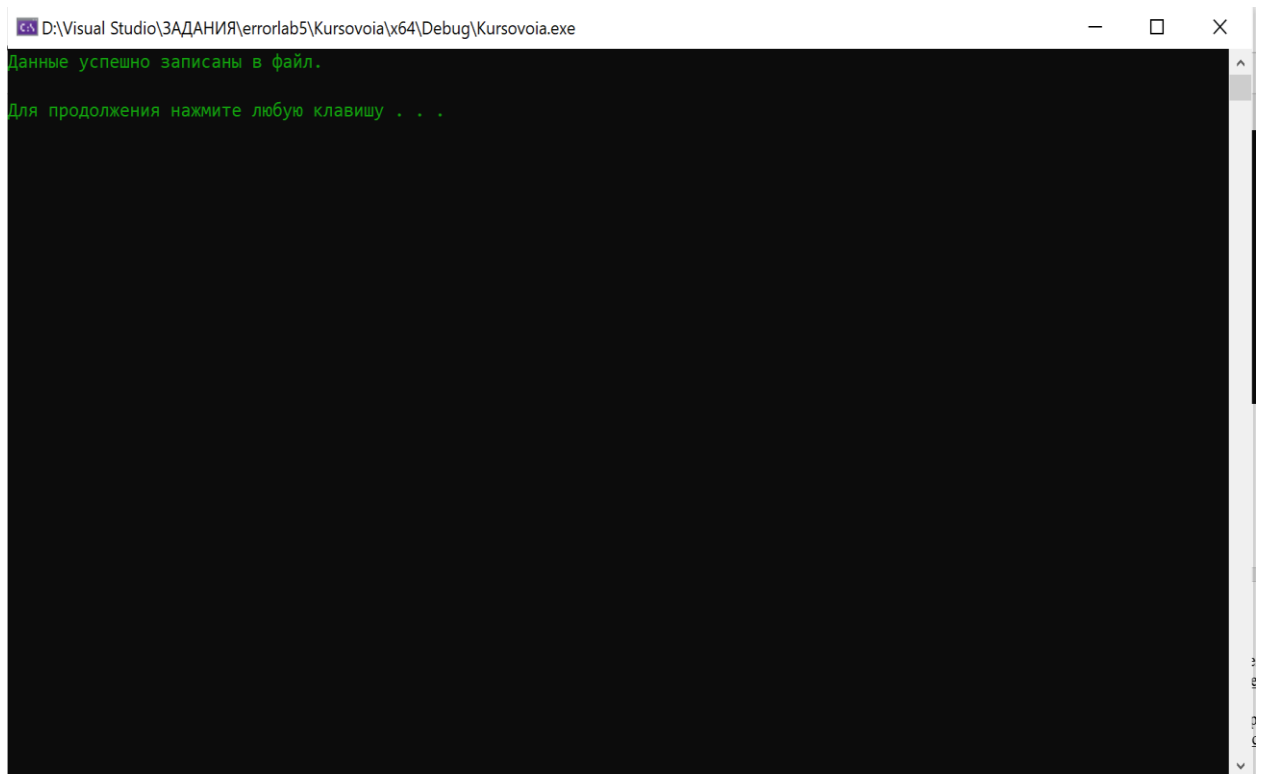
При выборе пункта “7” мы можем выполнить отмену добавления объекта (рисунок ниже).



```
D:\Visual Studio\ЗАДАНИЯ\errorlab5\Kursovoia\x64\Debug\Kursovoia.exe
Выберите тип: 1-Добавить в шаблон(в любую позицию)
              2-Вывести шаблон
              3-Поиск по названию
              4-Поиск по индексу
              5-Удаление объекта
              6-Отмена удаления
              7-Отмена добавления
              8-Прочитать из текстового файла
              9-Записать в текстовый файл
              10-Очистить текстовый файл
              11-показать в обратном порядке(с итератором)
              12-показать(с итератором)
              13-поиск по названию(с итератором)
              14-сортировка
              15-STL
              16-выход
7
Отмена добавления прошла успешно.

Для продолжения нажмите любую клавишу . . .
```

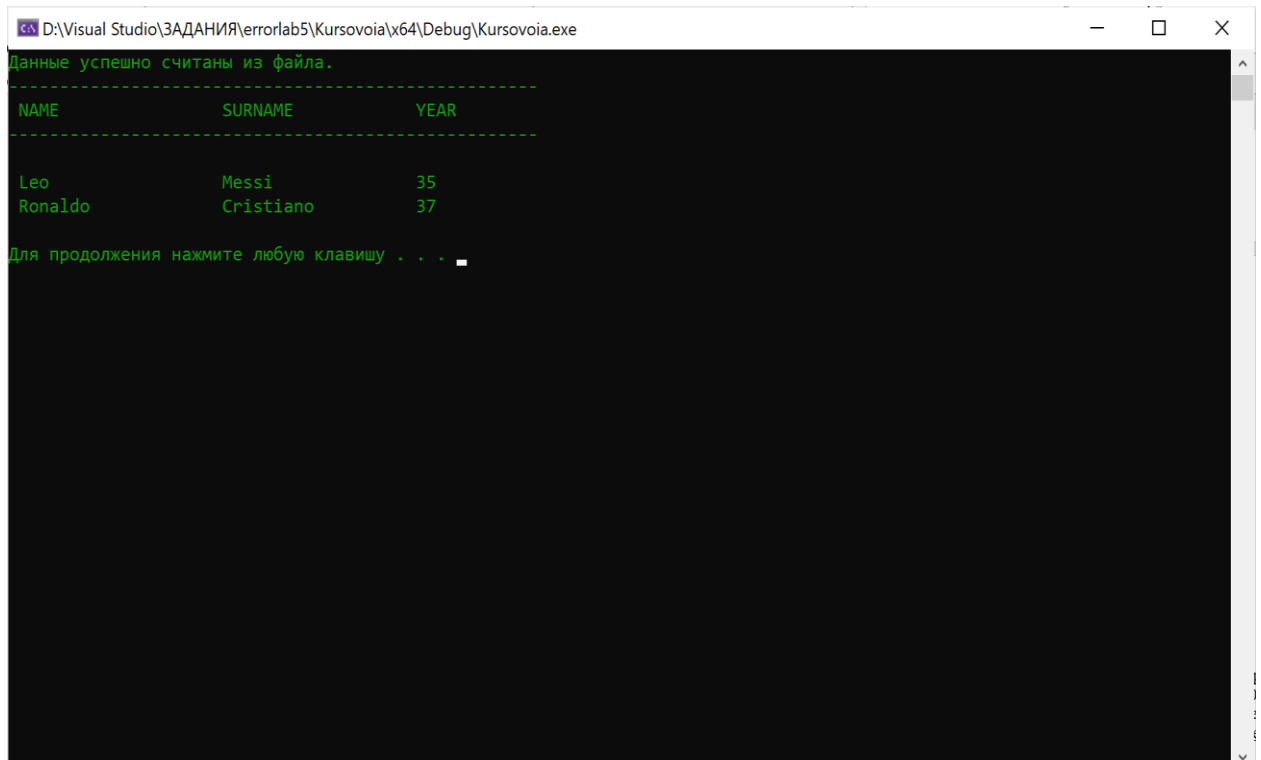
При выборе пункта “9” мы можем записать в текстовый файл(рисунок ниже).



The screenshot shows a Windows command prompt window titled "D:\Visual Studio\ЗАДАНИЯ\errorlab5\Kursovoia\x64\Debug\Kursovoia.exe". The text displayed is:

```
Данные успешно записаны в файл.  
Для продолжения нажмите любую клавишу . . .
```

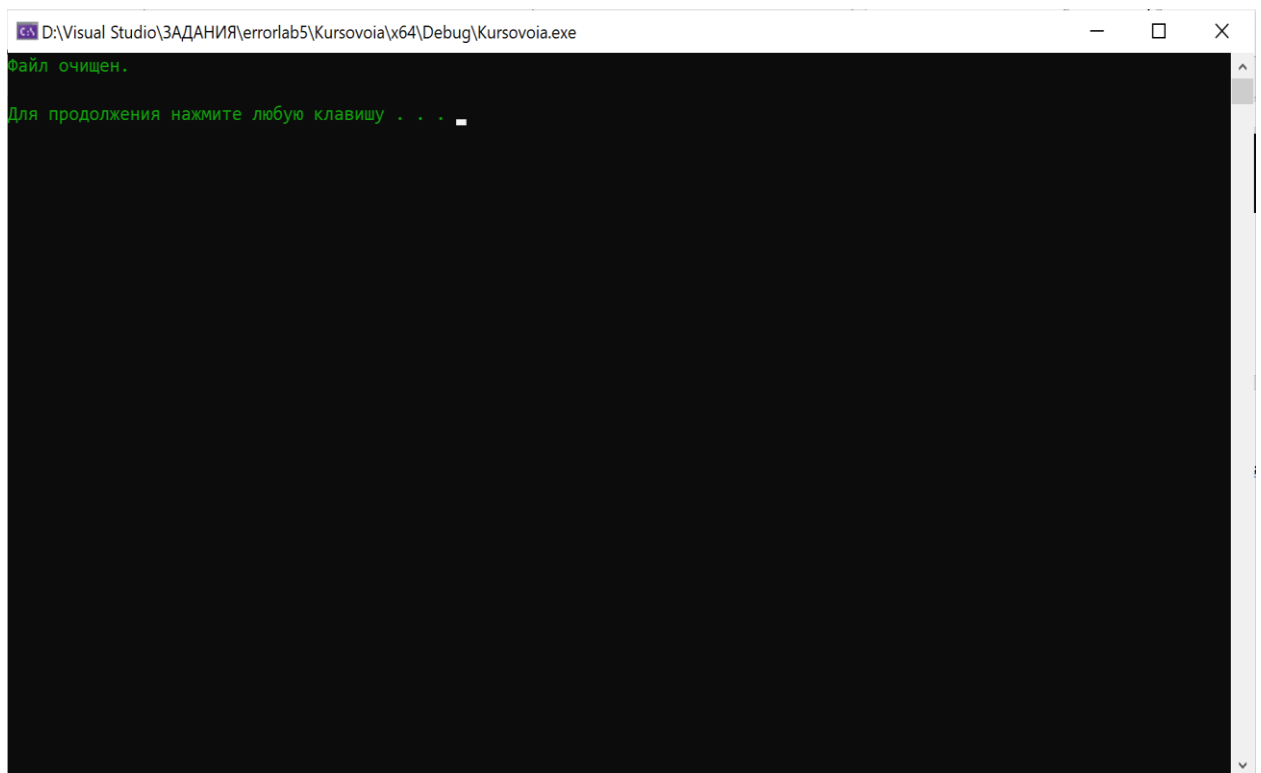
При выборе пункта “8” мы можем прочитать из текстового файл(рисунок ниже).



The screenshot shows a Windows command prompt window titled "D:\Visual Studio\ЗАДАНИЯ\errorlab5\Kursovoia\x64\Debug\Kursovoia.exe". The text displayed is:

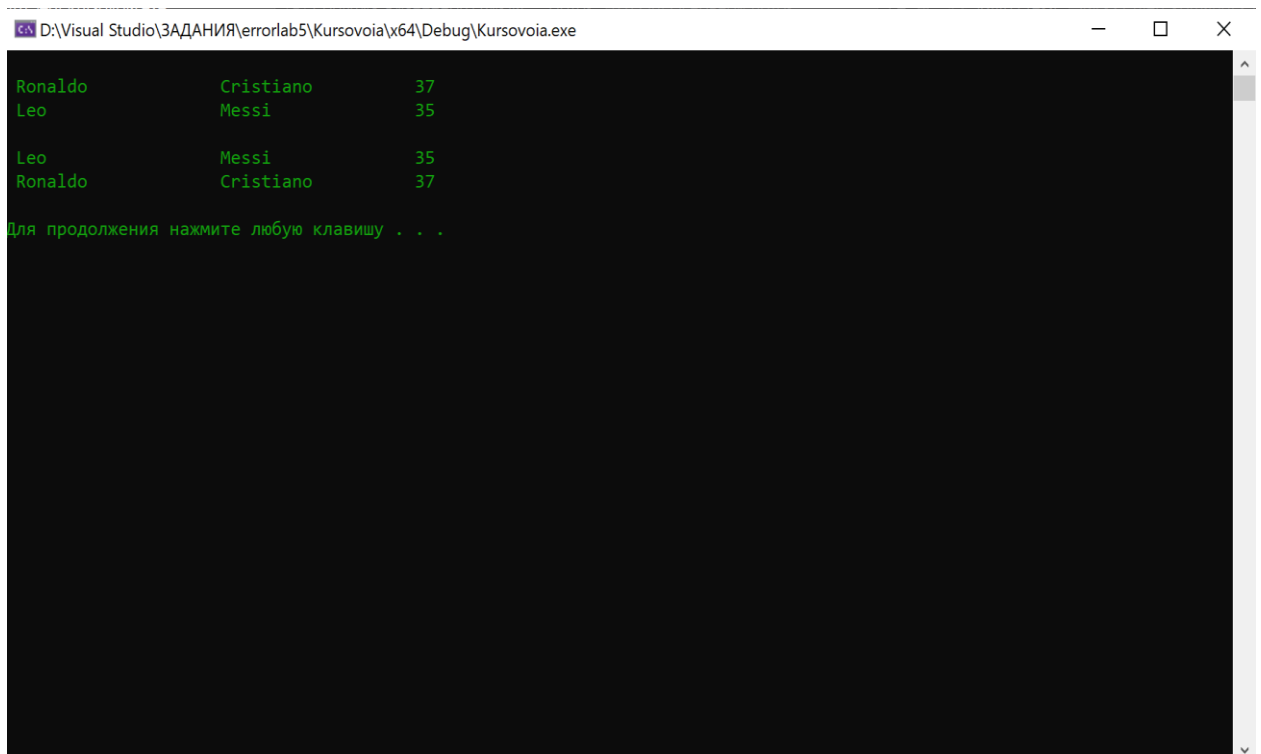
```
Данные успешно считаны из файла.  
-----  
NAME          SURNAME        YEAR  
-----  
Leo            Messi           35  
Ronaldo        Cristiano       37  
Для продолжения нажмите любую клавишу . . .
```

При выборе пункта “10” мы можем очистить текстовый файл(рисунок ниже).



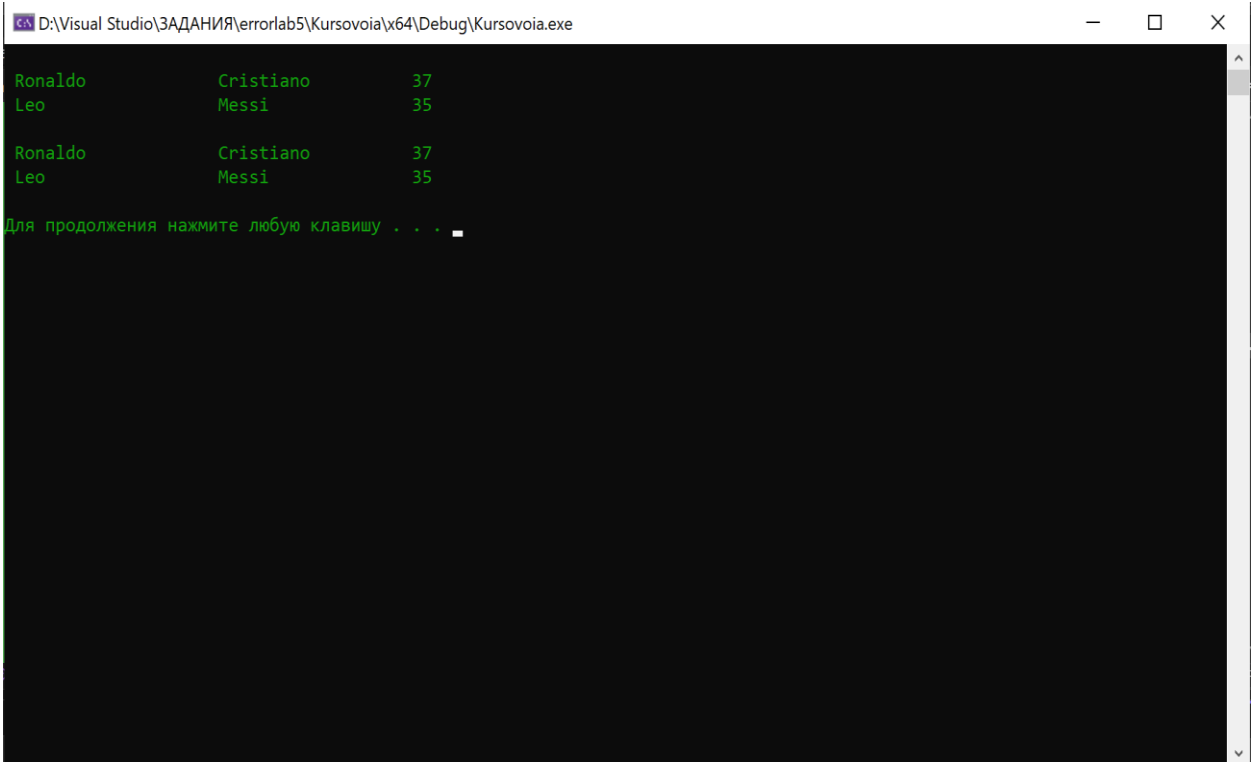
```
D:\Visual Studio\ЗАДАНИЯ\errorlab5\Kursovoia\x64\Debug\Kursovoia.exe
Файл очищен.
Для продолжения нажмите любую клавишу . . .
```

При выборе пункта “11” мы можем показать шаблон в обратном порядке с помощью итератора (рисунок ниже).



```
D:\Visual Studio\ЗАДАНИЯ\errorlab5\Kursovoia\x64\Debug\Kursovoia.exe
Ronaldo      Cristiano   37
Leo          Messi      35
Leo          Messi      35
Ronaldo      Cristiano   37
Для продолжения нажмите любую клавишу . . .
```

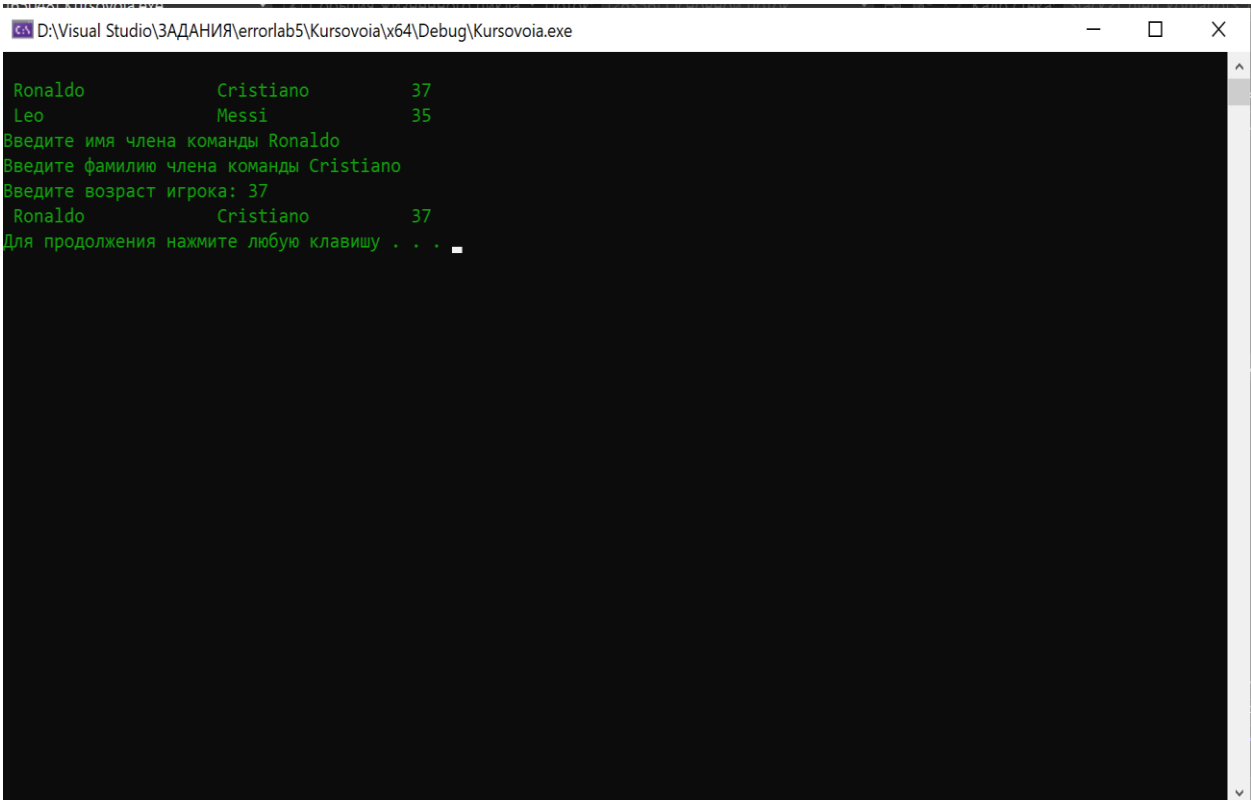
При выборе пункта “12” мы можем показать шаблон в обратном порядке с помощью итератора (рисунок ниже).



The screenshot shows a console window titled "D:\Visual Studio\ЗАДАНИЯ\errorlab5\Kursovoia\x64\Debug\Kursovoia.exe". The output displays two identical lists of players in reverse order. The first list contains Ronaldo (37), Leo (35), Cristiano (37), and Messi (35). The second list contains Ronaldo (37), Leo (35), Cristiano (37), and Messi (35). Below the lists, a prompt asks the user to press any key to continue.

```
D:\Visual Studio\ЗАДАНИЯ\errorlab5\Kursovoia\x64\Debug\Kursovoia.exe
Ronaldo      Cristiano   37
Leo          Messi       35
Ronaldo      Cristiano   37
Leo          Messi       35
Для продолжения нажмите любую клавишу . . .
```

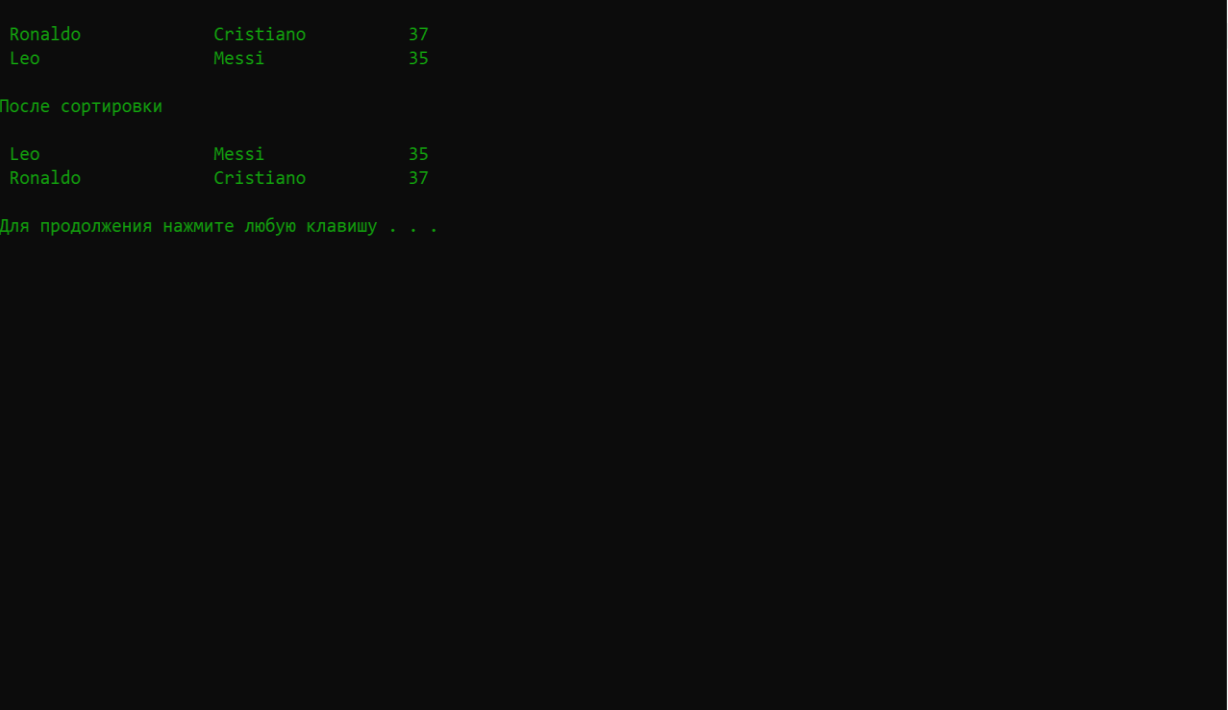
При выборе пункта “13” мы можем выполнить поиск по имени объекта в шаблоне, с помощью итератора (рисунок ниже).



The screenshot shows a console window titled "D:\Visual Studio\ЗАДАНИЯ\errorlab5\Kursovoia\x64\Debug\Kursovoia.exe". The output displays two identical lists of players in reverse order. The first list contains Ronaldo (37), Leo (35), Cristiano (37), and Messi (35). The second list contains Ronaldo (37), Leo (35), Cristiano (37), and Messi (35). Below the lists, a prompt asks the user to press any key to continue.

```
D:\Visual Studio\ЗАДАНИЯ\errorlab5\Kursovoia\x64\Debug\Kursovoia.exe
Ronaldo      Cristiano   37
Leo          Messi       35
Введите имя члена команды Ronaldo
Введите фамилию члена команды Cristiano
Введите возраст игрока: 37
Ronaldo      Cristiano   37
Leo          Messi       35
Для продолжения нажмите любую клавишу . . .
```


При выборе пункта “14” мы можем выполнить сортировку по имени объекта в шаблоне, с помощью итератора (рисунок ниже).



```
D:\Visual Studio\ЗАДАНИЯ\errorlab5\Kursovoia\x64\Debug\Kursovoia.exe

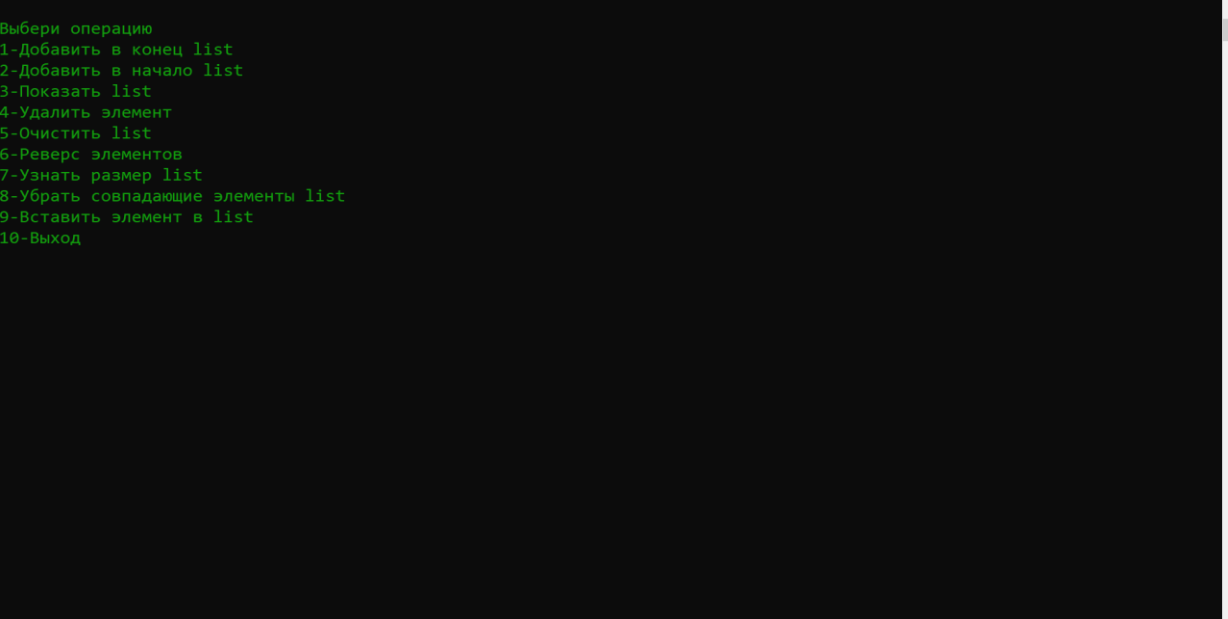
Ronaldo      Cristiano    37
Leo          Messi        35

После сортировки

Leo          Messi        35
Ronaldo      Cristiano    37

Для продолжения нажмите любую клавишу . . .
```

При выборе пункта “15” мы можем выбрать работу с STL(рисунок ниже).



```
D:\Visual Studio\ЗАДАНИЯ\errorlab5\Kursovoia\x64\Debug\Kursovoia.exe

Выбери операцию
1-Добавить в конец list
2-Добавить в начало list
3-Показать list
4-Удалить элемент
5-Очистить list
6-Реверс элементов
7-Узнать размер list
8-Убрать совпадающие элементы list
9-Вставить элемент в list
10-Выход
```

6. СПИСОК ЛИТЕРАТУРЫ

[1].Язык программирования C++ [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/C++>. – Дата доступа 19.9.2019.

[2].Языки программирования [Электронный ресурс]. – Режим доступа: http://sd-company.su/article/help_computers/programming_language – Дата доступа 5.10.2019.

[3].Технологии программирования [Электронный ресурс]. – Режим доступа: <http://info-master.su/programming/profi/programming-technologies.php> – Дата доступа 7.10.2019.

[4].Структура программного обеспечения ПК [Электронный ресурс]. – Режим доступа: https://studopedia.ru/4_4312_tema--struktura-programmnogo-obespecheniya-pk.html – Дата доступа 20.10.2019.

[5].Руководство по разработке на компонентах [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/company/ruvds/blog/461661/> – Дата доступа 11.10.2019.

[6].Теоретические основы алгоритмизации и программирования [Электронный ресурс]. – Режим доступа: <https://studfile.net/preview/4693947/> – Дата доступа 19.10.2019.

[7].Луцик, Ю. А. Объектно-ориентированное программирование на языке C++: учеб. пособие по курсу «Объектно-ориентированное программирование» для студ. спец. «Вычислительные машины, системы и сети» всех форм обуч. / Ю. А. Луцик, А. М. Ковальчук, И. В. Лукьянова. – Минск: БГУИР, 2003. – 203 с.:ил.

ЗАКЛЮЧЕНИЕ

В результате выполнения курсового проекта была разработано программное средство по управлению сведениями об игроках футбольной команды. Во время реализации данного проекта были использован язык программирования C++. При разработке программы стало понятно, что классы и объекты в C++ являются основными концепциями объектно-ориентированного программирования — ООП. А объектно-ориентированное программирование — расширение структурного программирования, в котором основными концепциями являются понятия классов и объектов. Стало ясно что основное отличие языка программирования C++ от C состоит в том, что в C нет классов, а следовательно язык C не поддерживает ООП, в отличие от C++.

В программе удалось реализовать удобный пользовательский интерфейс, хороший дизайн. Программа проста в использовании и имеет большой функционал, который помогает при работе с сведениями об игроках футбольной команды.

ПРИЛОЖЕНИЕ А

(обязательное)

Диаграмма классов

ПРИЛОЖЕНИЕ Б

(обязательное)

Схема алгоритма метода

```
template<typename TYPE>  
TYPE Algorithm<TYPE>::search1(Node<TYPE>* beg, int c)
```

ПРИЛОЖЕНИЕ В

(обязательное)

Схема алгоритма метода

```
template<class TYPE>
void Algorithm<TYPE>::sort(linklist<TYPE>& _a)
```

ПРИЛОЖЕНИЕ Г
(обязательное)

Код программы

ПРИЛОЖЕНИЕ Д
(обязательное)

Ведомость документов