**ОПИСАНИЕ ОСОБЕННОСТЕЙ РЕАЛИЗАЦИИ КЛИЕНТСКОГО МОДУЛЯ СИСТЕМЫ АВТОМАТИЗИРОВАННОГО СБОРА ДАННЫХ**

**Версия 1.0**

**СОДЕРЖАНИЕ**

1. **СТРУКТУРА ПРОЕКТА КЛИНЕТСКОГО МОДУЛЯ…………………………………..1**
2. **ОСОБЕННОСТИ РЕАЛИЗАЦИИ И ИСПОЛЬЗОВАНИЯ НЕКОТОРЫХ КЛАССОВ……………………………………………………………………………………………..2**
3. **ФОРМАТЫ БИНАРНЫХ ФАЙЛОВ…………………………………………………………4**
4. **ИНТЕРФЕЙС ПРИЛОЖЕНИЯ…………………………………………………………………6**
5. **Some changes for Lab3**

**Структура проекта клиентского модуля**

**-логика приложения (классы и соответствующие файлы .h и .cpp):**

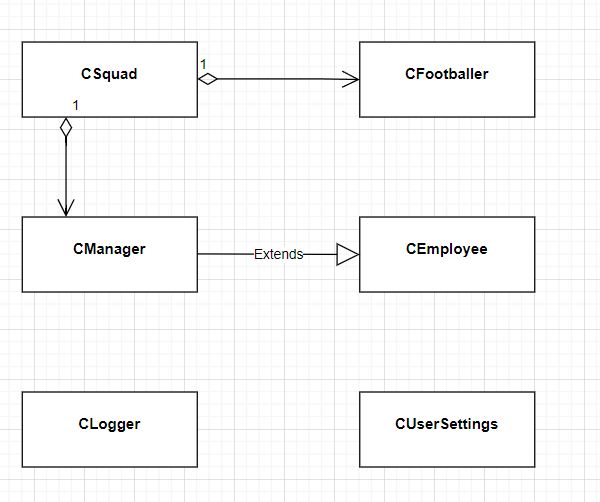
|  |  |
| --- | --- |
| CFootballer | Класс футболиста |
| CSquad | Класс состава команды |
| CEmployee | Класс сотрудника персонала клуба |
| CManager | Класс тренера состава |
| CUserSettings | Класс настроек текущего пользователя |
| CLogger | Логгер (класс, работающий с логом событий и действий пользователя) |

**-окна и GUI-элементы (классы и соответствующие файлы .h и .cpp):**

|  |  |
| --- | --- |
| CAuthorizeDialog | Окно авторизации пользователя; агрегирует класс списка пользователей CUserList; обращается к свойствам CUserSettings |
| CRequestTasksDialog | Окно получения заданий |
| CLogDialog | Окно с логом событий |
| CImageTextButton | Класс кнопки с картинкой и текстом |
| CCloseButton | Класс кнопки для закрытия окна |
| CAddFootballerDialog | Окно добавления футболиста |
| CAddEmployeeDialog | Окно добавления сотрудника |
| CSquadDataDialog | Окно информации о составе команды |
| CEmployeeDataDialog | Окно информации о сотрудниках клуба |

**-служебные файлы приложения:**

|  |
| --- |
| resource.h |
| stdafx.h и stdafx.cpp |

**(UML диаграмма основных классов приложения)**

**ОСОБЕННОСТИ РЕАЛИЗАЦИИ И ИСПОЛЬЗОВАНИЯ НЕКОТОРЫХ КЛАССОВ**

**-CFootballer**

Данный класс содержит информацию о футболисте. В нём будут иметься следующие поля:

-имя

-фамилия

-возраст (int)  
-зарплата (float)

-дата рождения (string)

-позиция на поле (string)

-дата окончания контракта (string)

-ид.номер (int)

**-CSquad**

Класс содержит в себе информацию о составе игроков. Он агрегирует класс CFootballer и делит их на разные составы (основной состав, запас, резерв, молодежная команда), а также содержит поле id.

**-CEmployee**

Класс содержит информацию о сотруднике клуба. Поля класса:

-имя

-фамилия

-дата рождения

-должность

-зарплата

**-CManager**

Наследует класс CEmployee и расширяет его следующими полями: специализация и состав, с которым тренер работает

Для каждого из классов CFootballer, CEmployee и CSquad будет реализован метод ReadDataFromFile(string filename), который будет считывать информацию об объекте класса из соответствующего бинарного файла (см. форматы бинарных файлов)

**-CLogger**

Класс для логирования всех событий и действий пользователя. Главная функция логгера – пишущая, т.е запись действий пользователя в соответствующий .dat файл с определённой структурой.

У каждого действия есть свой уникальный номер:

Добавить – 00  
Изменить – 01  
Удалить – 02  
При этом соответствующие действия мы можем производить с футболистом/составом/сотрудником

Футболист – 00  
Состав – 01  
Сотрудник – 02

Таким образом мы можем закодировать соответствующее действие в виде двух двузначных чисел.

**ФОРМАТЫ БИНАРНЫХ ФАЙЛОВ**

**Файл footballer.data**

**-4 байта** : ID футболиста (unsigned int)

-**4 байта** : количество символов в имени футболиста M (unsigned int)

-**M\*2 байт** : M символов в имени футболиста (M\*sizeof(TCHAR))

-**4 байта** : количество символов в фамилии футболиста M (unsigned int)

-**M\*2 байт** : M символов в фамилии футболиста (M\*sizeof(TCHAR))

-**4 байта** : возраст (int)

-**4 байта** : зарплата (float)

-**2 байта\*10** : дата рождения футболиста в формате DD.MM.YYYY (string)

-**2 байта\*2** : позиция футболиста на поле в формате из двух символов (CB,RB,ST и т.д) (string)

-**2 байта\*10** : дата окончания контракта футболиста в формате DD.MM.YYYY (string)

**Файл employee.data**

**-4 байта** : ID футболиста (unsigned int)

-**4 байта** : количество символов в имени сотрудника M (unsigned int)

-**M\*2 байт** : M символов в имени сотрудника (M\*sizeof(TCHAR))

-**4 байта** : количество символов в фамилии сотрудника M (unsigned int)

-**M\*2 байт** : M символов в фамилии сотрудника (M\*sizeof(TCHAR))

-**2 байта\*10** : дата рождения сотрдуника в формате DD.MM.YYYY (string)

-**4 байта** : количество символов в должности сотрудника M (unsigned int)

-**M\*2 байт** : M символов в должности сотрудника (M\*sizeof(TCHAR))

-**4 байта** : зарплата (float)

**Файл squad.dat**

**-4 байта** : ID состава (unsigned int)

-**4 байта** : количество игроков в составе M (unsigned int)

-**4 байта \* M** : M подряд идущих id игроков в составе, каждый имеет тип (int)

**Файл log.dat**

**-8 байт :** Дата / время события(FILETIME)

**-4 байта** : тип операции (unsigned int)

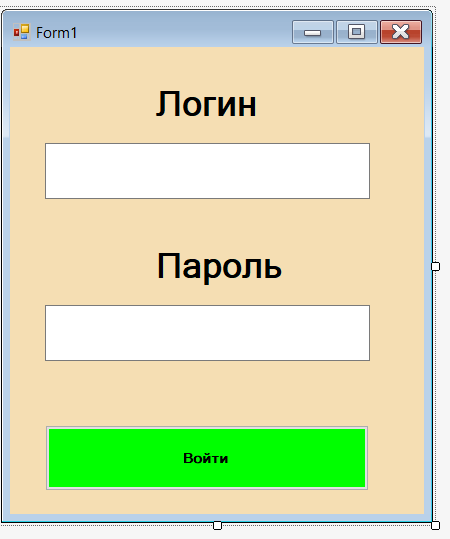
-**4 байта** : к какому объекту системы она относится (игрок, состав, сотрудник) (unsigned int)

-**4 байта** : id объекта, к которому применена операция

**ИНТЕРФЕЙС ПРИЛОЖЕНИЯ**

На данный момент дизайн приложения находится на начальной стадии разработки. Приложенные скриншоты являются лишь ***наработками*** интерфейса программы, и вполне могут видоизменяться до её конца разработки.

Окно входа пользователя (первым появляется при запуске приложения):



Окно выбора таблицы:

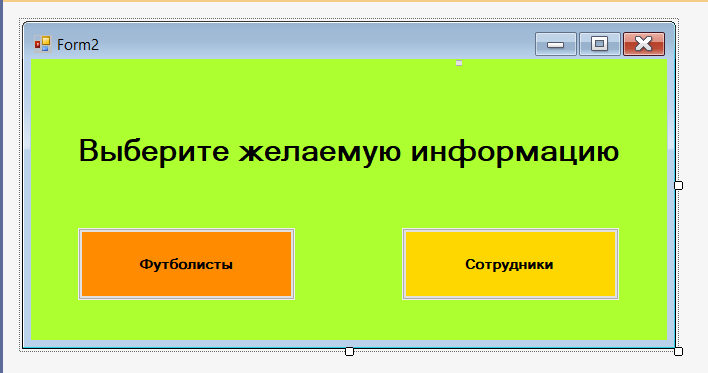
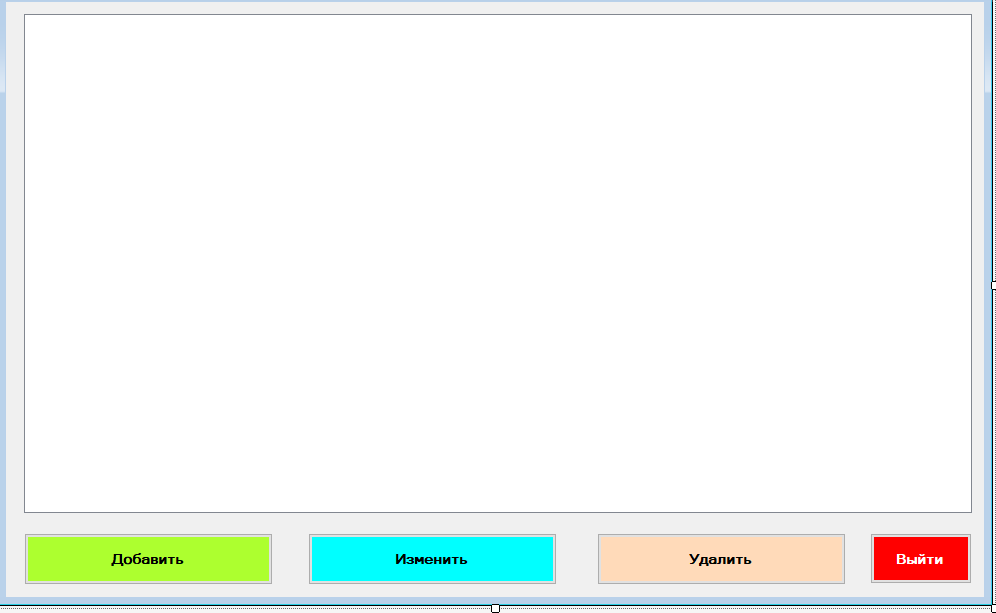


Таблица с данными:



Сама таблица находится в компоненте ListView. Ожидается, что она будет иметь стандартный вид (по типу таблиц в Access)