Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет телекоммуникаций и информатики»

(СибГУТИ)

09.03.01 Информатика и вычислительная техника

Профиль: Программное обеспечение средств вычислительной техники и автоматизированных систем

**Отчёт по курсовому проекту**

**по дисциплине «Визуальное программирование и человеко-машинное взаимодействие»**

**Вариант 1**

**«Электронный журнал куратора»**

по направлению 09.03.01 «Информатика и вычислительная техника»,

направленность (профиль) – «Программное обеспечение средств вычислительной техники и автоматизированных систем», квалификация – бакалавр,

программа академического бакалавриата,

форма обучения – очная, год начала подготовки (по учебному плану) – 2019

Выполнил: студент гр. ИП-916 /Меньщиков Д.А./

Проверил: ассистент кафедры ПМиК / Новожилов Д.И. /

Новосибирск, 2022

Содержание

[Введение 3](#_Toc533870777)

[1.1 Постановка задачи 4](#_Toc533870778)

[1.2. Проблемно-центрированная разработка интерфейса 5](#_Toc533870779)

[1.2.1. Анализ задач и пользователей 5](#_Toc533870780)

[1.2.2. Выбор репрезентативных задач 5](#_Toc533870781)

[1.2.3. Заимствование](#_Toc533870782) 7

[1.2.4. Черновое описание дизайна 7](#_Toc533870783)

[1.2.5. Схемы предварительного интерфейса.](#_Toc533870784) 8

[1.2.6. Тестирование дизайна с пользователями](#_Toc533870785) 10

[1.2.7. CWT-анализ разработанного интерфейса](#_Toc533870786) 10

[1.2.8. Анализ GOMS 1](#_Toc533870787)1

[1.2.9. Итерирование 1](#_Toc533870788)2

[1.2.10. Изменение дизайна 1](#_Toc533870790)3

[1.3. Анализ соответствия программы правилам Нильсена-Молиха 1](#_Toc533870791)3

2. Демонстрация работы программы 15

[Заключение 1](#_Toc533870792)8

[Список литературы 1](#_Toc533870793)9

# Введение

Целью курсового проекта является закрепление навыков полученных в ходе изучения дисциплины «Человеко-машинное взаимодействие».

Для достижения поставленной цели необходимо выполнение следующих задач:

- Провести этапы проблемно-центрированного дизайна программного продукта, помогающего пользователю вести учёт посетителей бассейна;

- Провести CWT-анализ разработанного интерфейса;

- Провести анализ GOMS разработанного интерфейса;

- По результатам CWT и GOMS анализа доработать интерфейс программы и выполнить создание макета или прототипа;

- Провести анализ соответствия программы правилам Нильсена-Молиха.

# 1.1 Постановка задачи

Мною была выбрана тема «Электронный журнал куратора». Было необходимо выполнить следующие задачи в среде QtCreator:

* База данных студентов: ФИО, Дата рождения, Группа, Фотография, Почта, Телефон, с возможностью добавления, удаления и редактирования.
* Разделение на группы.
* Поиск студента по ФИО, по группе и ФИО.
* Назначение старосты, смена старосты.
* План мероприятия по группам, добавление фотографий с мероприятий, учет посещения мероприятий студентами.
* Учет успеваемости студентов по нескольким предметам за один семестр, учитывая одну сессию.
* Добавление в список “внимание” неуспевающих студентов на текущий день.
* Вывод отчета по успеваемости выбранных студентов на текущий день, вывод отчета о посещении проведенных мероприятиях любого студента.

# 1.2. Проблемно-центрированная разработка интерфейса

В данном курсовом проекте я использовали подход, сфокусированный на задачах, которые нужно решать пользователю – проблемно-центрированный подход. При таком подходе процесс разработки у меня структурировался, исходя из специфических задач, которые пользователь нашей программы будет решать с помощью разрабатываемой системы. На ранней стадии разработки я выбрал задачи, так как в дальнейшем они использовались для выявления требований к дизайну, чтобы облегчить мне выработку решений и их оценку по мере развития проекта.

## 1.2.1. Анализ задач и пользователей

Анализ задач моего проекта заключается в следующем:

* Работа с базой данных студентов, добавление, удаление, редактирование студента.
* Поиск определенного студента, или определенной группы студентов.
* Работа с базой данных мероприятий, добавление, удаление, редактирование.
* Функция вывода фотографий с мероприятий, а также их добавление и удаление.
* Функция добавление и удаление студентов в список посещенных отдельного мероприятия.
* Работа с оценками студентов, изменение оценок, вывод результирующего балла.
* Работа с базой данных старост, добавление старосты, удаление.

Следующим шагом необходимо произвести анализ пользователей. Исходя из этого, разрабатываемый программный продукт нацелен для управления бассейном, им будут пользоваться сотрудники бассейнов.

## 1.2.2. Выбор репрезентативных задач

Перед началом создания программного продукта было выделено несколько репрезентативных задач, которые будут решаться при использовании системы, перечислим их:

* Формирование базы данных студентов.
* Формирование мероприятий, добавление фотографий, учёт посещения.
* Учет успеваемости отдельного студента по нескольким предметам.
* Добавление в список “внимание” неуспевающих студентов.
* Возможность назначение или смену текущего старосты.

База данных в программном продукте состоит из 4 таблиц:

1-я таблица:

1. ФИО
2. Дата рождения
3. Группа
4. Фото
5. Почта
6. Телефон

2-я таблица:

1. Название мероприятия
2. Дата проведения

3-я таблица:

1. ФИО
2. Группа
3. Предмет успеваемости (ОС);
4. Предмет успеваемости (Математика);
5. Предмет успеваемости (Программирование);
6. Предмет успеваемости (Геометрия);
7. Результат (Сессия)

## 

## 1.2.3. Заимствование

За основу своей будущей программы, я брал программы для работы с таблицами, например Microsoft Office Excel, но с упрощенным интерфейсом, и более узконаправленным инструментарием.

1. Таблица, отображающая данные схожа с таблицей Excel

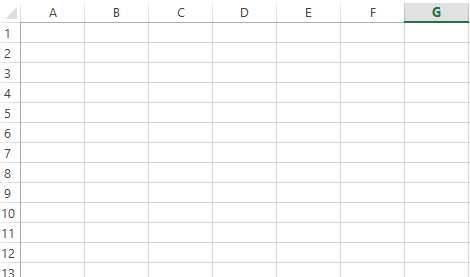


Рис 1. Таблица Excel

На основе данной таблице мне удалось построить черновой вариант моего программного продукта.

## 1.2.4. Черновое описание дизайна

В качестве черного дизайна я представляю интерфейс с несколькими вкладками, в которых будет располагаться информация о студентах, мероприятиях, результатах оценивания и таблицу со старостами.

## 1.2.5. Схемы предварительного интерфейса.

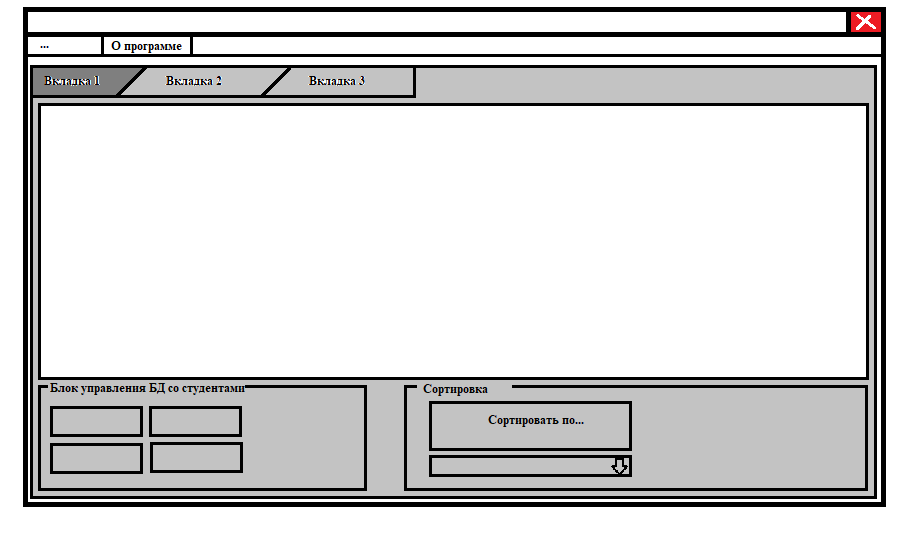


Рис 1. Вкладка со студентами

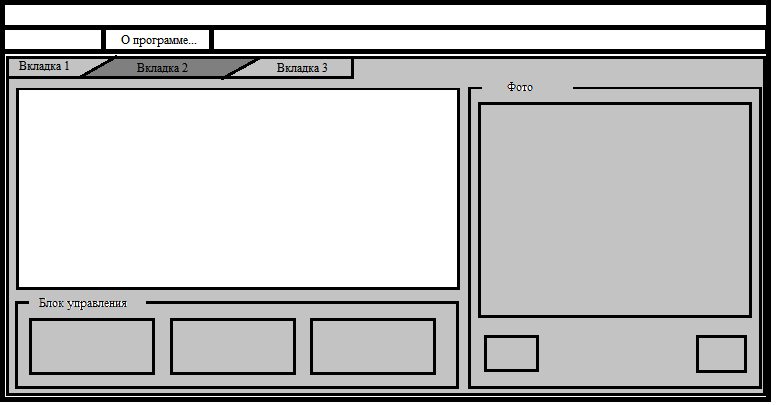


Рис 2. Вкладка с мероприятиями



Рис 3. Добавление студента

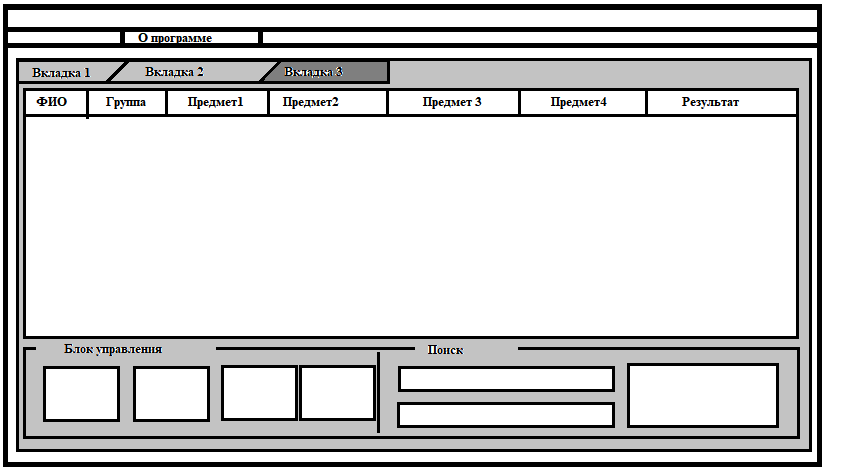


Рис. 4. База данных с результатами

## 1.2.6. Тестирование дизайна с пользователями

На тестирование моего программного продукта отозвался Исмаилов Рухид – студент 3 курса СибГУТИ, владеет высокими навыками в среде создания программных продуктов.

Также были проведены GOMS и CWT анализы чернового интерфейса:

## 1.2.7. CWT-анализ разработанного интерфейса

1. Внесение студента в базу данных.
2. Внесение оценок отдельного студента в таблицу и вывод результата.

Анализ первой задачи:

1. Запуск программы. Действие не связано с интерфейсом рассматриваемой программы и обеспечивается интерфейсом другой системы, не подвергающейся анализу.
2. Внизу окна нажимаем кнопку «Добавить студента», после чего открывается новое окно с посещениями.
3. В открывшемся окне вводим все поля, нажимаем на кнопку добавить фото.
4. Далее нажимаем кнопку добавить, окно закроется автоматически.

Проведя данный анализ, можно сказать что добавление студента выглядит достаточно интуитивно. Не опытный пользователь, или пользователь открывший программу в первый раз сразу могут разобраться в данном интерфейсе.

Само внесение студента в базу данных выглядит достаточно простым, у клиента открывается окно где наглядно просят ввести нужные данные.

Анализ второй задачи:

1. Запуск программы. Действие не связано с интерфейсом рассматриваемой программы и обеспечивается интерфейсом другой системы, не подвергающейся анализу.
2. Открыв приложение, нужно перейти в третью вкладку. Там уже выбрать определенного студента и внести оценки.
3. Внесение оценок происходит таким образом: В ячейки базы данных можно вписывать определенные оценки, по определенным предметам.
4. Что бы выставить результат можно воспользоваться кнопкой ниже которая автоматически подсчитает итоговый балл и впишет его в таблицу.

В данном анализе становится понятно, что данный черновой дизайн решает все репрезентативные задачи. Пользователь получает все то, что ему требуется от данной таблицы, а это вывод ввод/редактирование оценок отдельного студента.

Подведя итог, можно выделить следящие проблемы чернового дизайна:

1. Черновой дизайн не полностью определяет готовый продукт, в процессе разработки произойдет множество изменений в виду реализации некоторых функций.

Так же можно выделить сильные стороны чернового дизайна:

1. Однотипность окон, то есть пользователь переходя из одной вкладки в другую будет видеть один и тот же интерфейс, что позволяет легко в нем ориентироваться.
2. Интерфейс является разгруженным, то есть на нем только та информация, которая нужна пользователю, без лишних окон и кнопок.

## 1.2.8. Анализ GOMS

1. Внесение студента в базу данных.
2. Внесение оценок отдельного студента в таблицу и вывод результата.

Анализ первой задачи:

1. Переместиться к кнопке «Добавить студента» и нажать её. (МРB)
2. В открывшемся окне внести во все поля информацию о студентах. (M5(PBHK)HPB)
3. Нажать кнопку «Добавить». (PB)

MPB M5(PBHK) HPB PB

Общее время = 1.35 + 1.1 + 0.2 + 1.35 + 5 \* (1.1 + 0.2 + 0.4 + 0.2) + 0.4 + 1.1 + 0.2 + 1.1 + 0.2 = 16,5

Если сделать обработку на то, что после того как пользователь введет в нужное поле информацию и нажмет на Enter, то курсор сместиться на другое поле, то это поможет сократить время, и не нужно будет переставлять руку с клавиатуры на мышь.

Анализ второй задачи при условии того, что мы уже находимся на вкладке с оценками:

1. Найти определенного студента в списке (MDMP)
2. Нажать на поле с оценкой, и ввести её. (4(PBHKH))
3. Нажать на кнопку “Подвести результат”. (MPB)

MPBH8KH PB MPB

Общее время = 1.35 + 2 + 1.35 + 1.1 + 4 \* (1.1 + 0.2 + 0.4 + 0.2 + 0.4) + 1.35 + 1.1 + 0.2 = 17,65

Общее время можно сократить таким способом: можно выводить информацию в результирующую колонку сразу, либо по нажатию определенной клавиши, что сократит время примерно на 3 секунды, так же можно воспользоваться поиском, но время возможно не измениться, либо измениться не значительно.

Проведя данный анализ, удалось выявить явные проблемы интерфейса и предложить достойные улучшения для него, которые будут учтены при создании программного продукта.

## 1.2.9. Итерирование

Проделав анализы, можно предположить улучшения для интерфейса приложения:

1. Изменения дизайна в угоду удобству, продолжить располагать элементы так, чтобы пользователю было удобно в них ориентироваться, использовать элементы группировки инструментов.
2. Внести изменения после проведения анализа.
3. По требованию тестировщика продукта добавить отдельный элемент во вкладки, где будет отображаться информация о студенте по наведению либо нажатию на него в базе данных, для более удобного её просмотра.

## 

## 1.2.10. Изменение дизайна

Были проделаны все задачи, высказанные в итерировании.

# 

# 1.3. Анализ соответствия программы правилам Нильсена-Молиха

Проведем анализ соответствия интерфейса правилам Нильсена-Молиха:

1. Простой и естественный диалог:

Интерфейс программы ясен и прост как для опытного пользователя, так и для новичка.

2. Говорите на языке пользователя:

В программе не используются сложные, специфические слова.

3. Минимизируйте загрузку памяти пользователя:

Вся необходимая информация выводится на одном экране. При работе с программой пользователю не придётся запоминать ничего лишнего.

4. Будьте последовательны:

У пользователей имеется возможность изучить действия в одной части программы и применить их снова, чтобы получить похожие результаты в других местах.

5. Обеспечьте обратную связь:

Пользователь видит эффект, оказываемый его действиями на систему.

6. Обеспечьте хорошо обозначенные выходы:

Пользователь может выйти из любой части программы, не повредив её данные.

7. Обеспечьте быстрые клавиши и ярлыки:

В программе есть наличие быстрых переходов.

8. Хорошие сообщения об ошибках:

Программа выводит все ошибки.

9. Предотвращайте ошибки:

Программа обрабатывает наиболее распространённые ошибки.

**2. Демонстрация работы программы**

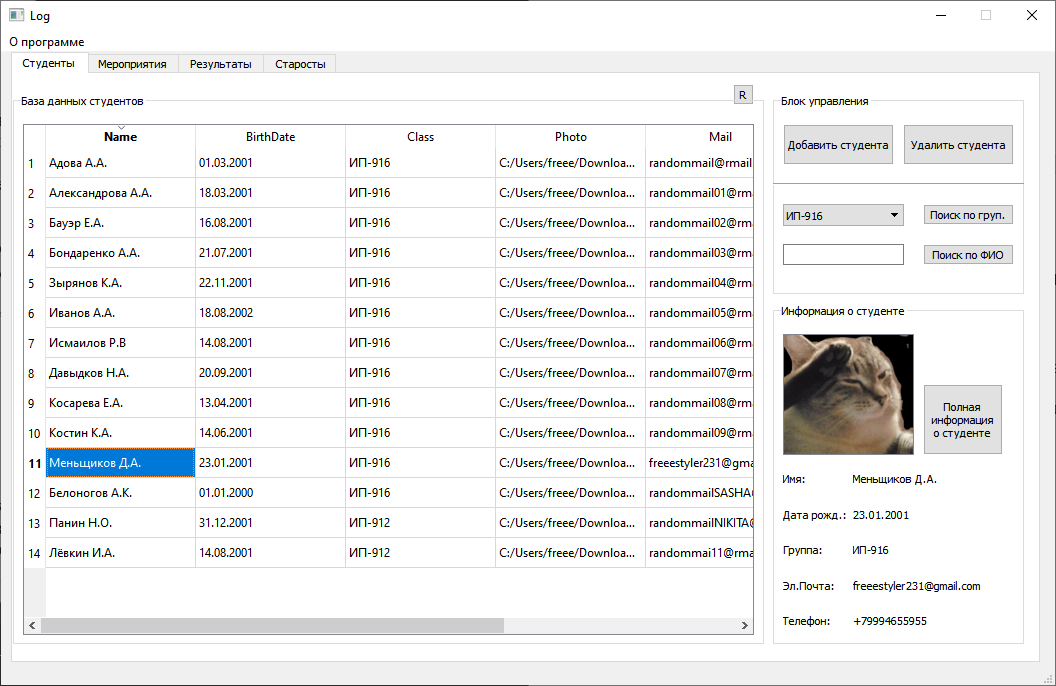


Рис 1. База данных со студентами

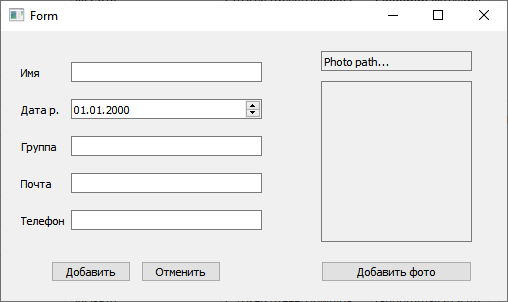
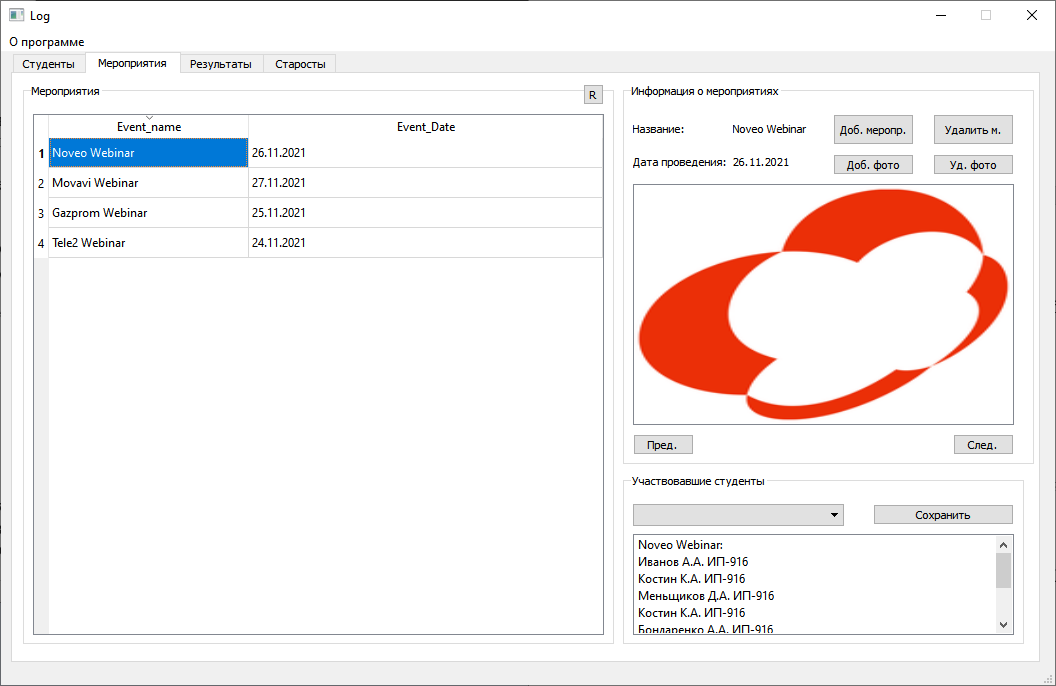


Рис 12. Форма для добавление студентов в базу данных

Рис 3. Вкладка с мероприятиями

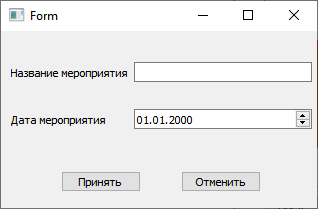


Рис 4. Форма с добавлением нового мероприятия

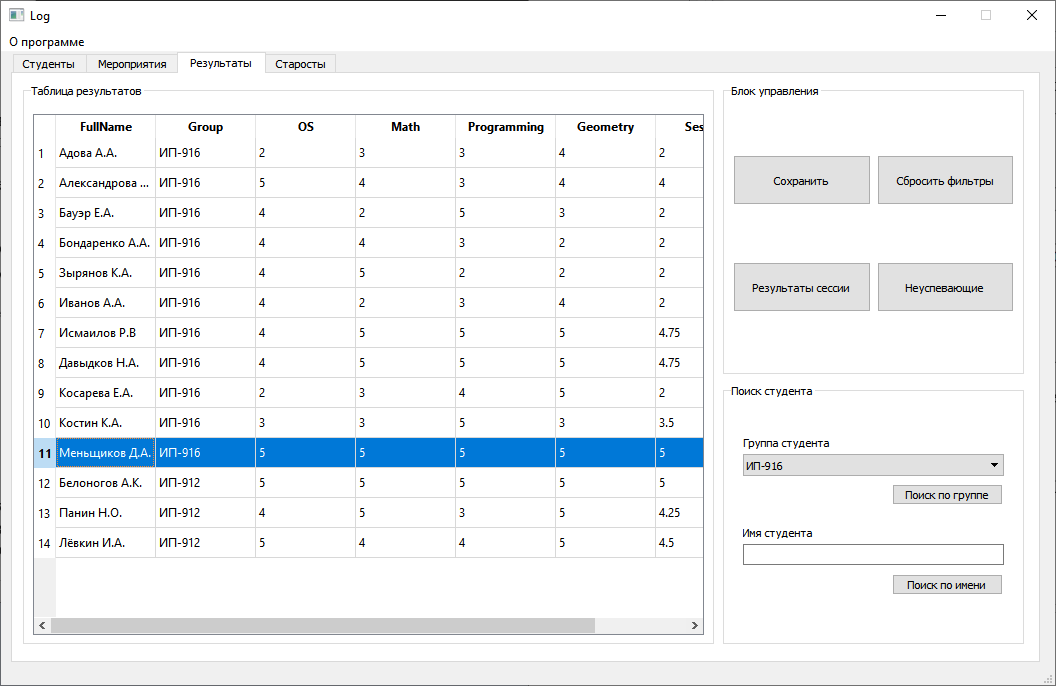


Рис 5. Таблица с текущими результатами студента

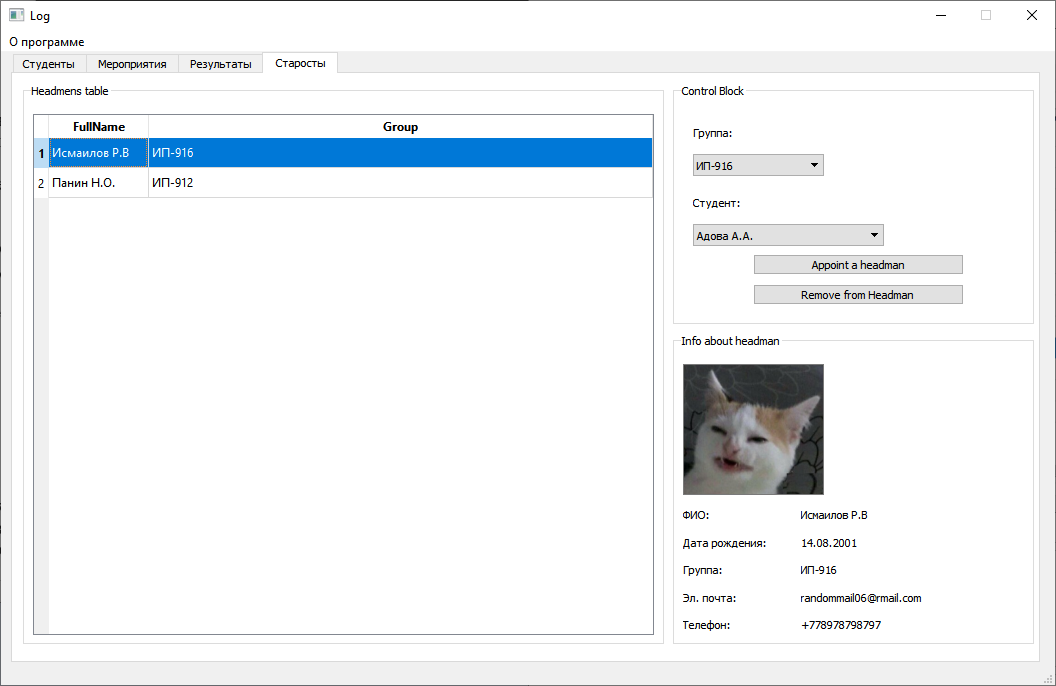


Рис 6. .

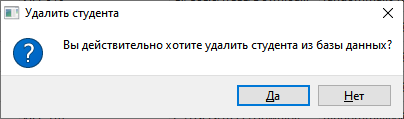


Рис 7. Одно из предупреждающих окон.

# Заключение

Целью курсового проекта является закрепление навыков, полученных в ходе изучения дисциплины «Человеко-машинное взаимодействие». В ходе выполнения курсового проекта было разработано программное решение, позволяющее редактировать клиентов, посещения и получать отчет по выручке.

Для разработки интерфейса программы были осуществлены все заявленные этапы, также были произведены CWT-анализ и GOMS-анализ. Программное обеспечение соответствует правилам Нильсена-Молиха.

# Список литературы

1. Мерзлякова Е.Ю. Человеко-машинное взаимодействие. Методические указания по выполнению РГЗ/ Сиб. гос. ун-т телекоммуникаций и информатики. – Новосибирск,2015. 24 с.

2. Акчурин Э.А. Человеко-машинное взаимодействие. Учебное пособие. Солон, 2008. 96 с. 3. Логунова О.С., Ячиков И.М., Ильина Е.А. Человеко-машинное взаимодействие: теория и практика. Учебноепособие. Феникс, 2006. 285с.