МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНО-ЭКОНОМИЧЕСКИЙ КОЛЛЕДЖ»

Специальность 2-40 01 01 «Программное обеспечение

информационных технологий»

**КУРСОВОЙ ПРОЕКТ**

по предмету «Конструирование программ и языки программирования»

**на тему Информационная система «»**

Пояснительная записка

Разработал обучающийся

группы ПО-32 \_\_\_\_\_\_\_\_\_\_\_\_\_ \_ Игнатенко А.С.\_\_\_\_\_\_\_\_\_\_\_

(подпись) (ФИО)

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_Пилецкая С.А.\_\_\_\_\_\_\_\_\_

(подпись) (ФИО)

Дата защиты \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Отметка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Гомель, 2024

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc167434950)

[1 ВЫБОР МЕТОДОВ И СРЕДСТВ РАЗРАБОТКИ ВЕБ- ПРИЛОЖЕНИЯ 4](#_Toc167434951)

[**1.1** **Описание предметной области и круга задач для создания веб-приложения** 4](#_Toc167434952)

[**1.2 Описание используемых технологий и языков программирования** 6](#_Toc167434953)

[2 ПРОЕКТИРОАНИЕ ПРОГРАММНОГОПРОДУКТА 9](#_Toc167434954)

[**2.1** **Концептуальное проектирование** 9](#_Toc167434955)

[3 РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ «ТЕМА» 12](#_Toc167434956)

[**3.1 Графический интерфейс программы** 12](#_Toc167434957)

[**3.2 Логическая модель данных** 24](#_Toc167434958)

[**3.3 Реализация функций программы** 27](#_Toc167434959)

[4 ДЕМОНСТРАЦИЯ РАБОТЫ ПРОГРАММЫ 34](#_Toc167434960)

[ЗАКЛЮЧЕНИЕ 50](#_Toc167434961)

[Список использованных источников 51](#_Toc167434962)

[ПРИЛОЖЕНИЕ А 52](#_Toc167434963)

## ВВЕДЕНИЕ

Автошкола — это учебное заведение, специализирующееся на обучении вождению и подготовке будущих водителей к получению водительских прав. Курсы в автошколе предоставляют обучающимся необходимые навыки, знания и практический опыт, чтобы стать уверенными и ответственными водителями на дороге. Автошколы играют важную роль в обучении новых водителей и повышении безопасности на дорогах. Они обучают не только навыкам вождения, но и важным аспектам безопасного и эффективного использования автомобиля. Традиционно обучение происходит непосредственно в автошколе, но не у всех людей есть возможно посещать все занятия в автошколе по тем или иным причинам, проверка многочисленных тестов от учащихся в автошколе в письменном виде создает большую нагрузку на инструкторов. Для решения этих проблем возникает необходимость создания веб-приложения «Автошкола», которая предоставит доступ к теоретическим материалам и тестам в электронном виде.

Цель разработка + тема

Цель разработки является обеспечении эффективного и удобного инструмента для проведения тестов и проверки знаний учащихся автошколы, подготовка студентов к сдаче экзаменов на получение водительского удостоверения, предоставляя им практические и теоретические тесты.

Задачи исследования:

Радел 1.1 и тд

* Описание предметной области и круга задач;
* Описание используемых технологий и языков программирования;
* Концептуальное проектирование;
* Описание логической модели;

Актуальность выбранной темы обусловлена необходимостью цифровизации, которая способствует получению доступа к материалам автошколы из любой точки, упрощение тестирования, снижение нагрузки на инструкторов, упрощение процесса анализа информации.

Таким образом, предметная область связана с проектированием, разработкой, внедрением и эксплуатацией информационной системы, которая позволяет автоматизировать тестирование и получение материалов необходимых для подготовки к тестам в автошколе.

## 1 ВЫБОР МЕТОДОВ И СРЕДСТВ РАЗРАБОТКИ ВЕБ- ПРИЛОЖЕНИЯ

### **Описание предметной области и круга задач для создания веб-приложения**

Тестирование в автошколе является важным этапом в процессе обучения будущих водителей. Оно помогает оценить знания и навыки учеников, а также убедиться в их готовности к безопасному управлению автомобилем на дороге. В автошколе обычно применяются письменные экзамены и практические тесты, которые позволяют проверить различные аспекты вождения.

Письменные экзамены в автошколе проводятся для проверки теоретических знаний учеников о правилах дорожного движения и других важных аспектах. Вопросы в экзаменах могут касаться различных ситуаций на дороге и требуют от учеников правильных ответов и демонстрации их понимания правил и процедур.

Категории вопросов включают следующие темы:

* правила дорожного движения: Вопросы в этой категории охватывают основные правила, которые регулируют движение на дороге. Это может включать правила обгона, дистанции между автомобилями, повороты, остановки и другие;
* знаки и сигналы: Вопросы, связанные с дорожными знаками, светофорами и другими сигнальными устройствами. Ученик должен знать значение различных знаков и правила, связанные с сигналами светофора, чтобы правильно реагировать на них;
* процедуры безопасности: Вопросы в этой категории относятся к мерам безопасности, которые необходимо соблюдать при вождении автомобиля. Это может включать правила использования ремней безопасности, детских автокресел, антиблокировочной системы тормозов (ABS) и других;
* приоритеты и проезд перекрестков: Вопросы в этой категории охватывают правила, касающиеся приоритетов на дороге, особенно при проезде перекрестков. Ученик должен знать, кто должен уступать дорогу и как правильно действовать в различных ситуациях на перекрестках;
* скоростной режим: Вопросы, связанные с ограничениями скорости и правилами ее соблюдения на разных участках дороги. Ученик должен знать, как определить и соблюдать допустимую скорость в различных ситуациях;
* повороты и перестроения: Вопросы, связанные с правилами поворотов, разворотов и перестроений на дороге. Ученик должен знать, как использовать указатели поворота и безопасно выполнять маневры;
* парковка: Вопросы, касающиеся правил парковки автомобиля. Ученик должен знать правила параллельной парковки, парковки на горке, парковки на ограниченном пространстве и другие аспекты парковки;

Вопросы в письменных экзаменах могут быть представлены в форме одиночного выбора, где ученику предлагается несколько вариантов ответа, и он должен выбрать правильный.

Процесс письменного тестирования состоит из:

* подготовка вопросов: Инструкторы или автошкола составляют набор вопросов, основываясь на официальных правилах дорожного движения и учебных материалах. Вопросы могут быть разного типа, включая одиночный выбор, множественный выбор или открытый ответ.
* раздача тестовых листов: Ученикам выдаются тестовые листы, на которых они должны записать свои ответы на каждый вопрос. Листы могут содержать номера вопросов и соответствующие поля для ответов.
* ответы на вопросы: Ученики читают каждый вопрос и выбирают наиболее подходящий ответ или записывают свой ответ в отведенное поле. Они должны использовать свои знания и понимание правил дорожного движения для выбора правильных ответов.
* сбор и проверка ответов: После того, как ученики завершают тестирование, их тестовые листы собираются. Инструкторы или автошкола затем проверяют ответы каждого ученика, чтобы определить, правильно ли они ответили на каждый вопрос.
* оценка результатов: После проверки ответов учеников, результаты оцениваются. Обычно используется система баллов, где каждый правильный ответ дает определенное количество баллов. В конце тестирования ученикам сообщается их общий балл или процент правильных ответов.
* обратная связь и исправление ошибок: Ученикам предоставляется обратная связь о результатах их тестирования. Они могут узнать, какие вопросы были отвечены правильно, а какие ошибочно. Это помогает им определить свои слабые места и сосредоточиться на изучении соответствующих тем.
* повторное тестирование: В случае неудачи или низкого результата, ученикам может быть предоставлена возможность повторного тестирования после дополнительной подготовки и изучения материала.

Кроме письменных экзаменов, в автошколе проводятся практические тесты. При этом ученик садится за руль автомобиля под наблюдением инструктора или экзаменатора. На практическом тесте проверяются навыки управления автомобилем, выполнение поворотов, перестроений, парковки и других маневров. Экзаменатор может также оценивать умение ученика соблюдать правила дорожного движения, реагировать на дорожные ситуации и общаться с другими участниками движения.

Повторно экзамены назначаются не ранее чем через шесть дней со дня, следующего за днём последнего экзамена. Для лиц, не сдавших теоретический экзамен либо один из этапов практического экзамена с третьей и последующих попыток, повторный экзамен назначается не ранее чем через 30 дней.

В случае непредставления экзаменационной карточки по теоретическому и практическому экзаменам в РЭП для получения водительского удостоверения в течение трёх лет теория и практика назначаются вновь.

Положительная оценка по теоретическому экзамену «сдан» считается действительной в течение трех месяцев, по истечении которых теоретический экзамен назначается вновь.

Важно отметить, что цель тестирования в автошколе не только в проверке знаний, но и в формировании навыков безопасного вождения. Тестирование помогает ученикам осознать и исправить свои ошибки, повысить уровень безопасности на дороге и стать ответственными водителями.

### **1.2 Описание используемых технологий и языков программирования**

Существует множество языков программирования, которые можно использовать для создания веб-приложений.

* «JavaScript»;
* «Python»;
* «PHP»;
* «Java»;
* «C#»;
* «TypeScript».

Для разработки веб-приложений, была выбрана мощная и гибкая платформу «ASP.NET Core». Эта открытая платформа основана на языке программирования «C#», который обеспечивает надежность, эффективность и безопасность разработки. Вместе с «C#» используется «JavaScript», «HTML» и «CSS», чтобы создавать привлекательный пользовательский интерфейс и обеспечивать интерактивность веб-приложения. В качестве среды разработки выбрана «Visual Studio».

«ASP.NET Core» — это открытая платформа для разработки веб-приложений и сервисов, основанная на языке программирования «C#» и фреймворке «.NET». «ASP.NET Core» позволяет разработчикам использовать языки программирования и технологии, включая «C#», «JavaScript», «HTML» и «CSS», для создания веб-приложений. «ASP.NET Core» также предоставляет поддержку архитектурного шаблона «MVC» («Model-View-Controller») для разработки веб-приложений. «MVC» позволяет разделить приложение на три основных компонента: модель («Model»), представление («View») и контроллер («Controller»).

«C#» — это современный объектно-ориентированный язык программирования, разработанный «Microsoft», который является частью платформы «.NET» и предназначен для создания различных приложений, включая веб-сайты, десктопные приложения, мобильные приложения и игры.

«JavaScript» — это интерпретируемый язык программирования, который применяется для создания динамических веб-страниц и веб-приложений. JavaScript является основным языком программирования для разработки клиентской части веб-приложений. Он позволяет взаимодействовать с элементами «HTML» и «CSS», изменять содержимое страницы, обрабатывать события (например, клики мыши или отправку форм) и создавать анимацию.

«HTML» — это язык разметки, используемый для создания структуры и представления содержимого веб-страниц. «HTML» состоит из различных элементов, которые определяют, как содержимое будет отображаться в браузере.

«CSS» — это язык стилей, используемый для оформления и визуального представления веб-страниц, созданных с помощью «HTML». «CSS» позволяет управлять внешним видом элементов «HTML», таких как цвета, шрифты, размеры, расположение и другие аспекты дизайна.

«MVC» («Model-View-Controller») - это архитектурный шаблон, в котором приложение разделено на три основных компонента: «Model» (Модель), «View» (Представление) и «Controller» (Контроллер). Каждый компонент выполняет свою роль в обработке запросов и представлении данных пользователю.

Модель представляет данные и бизнес-логику приложения. Она отвечает за доступ к данным, их валидацию, обработку бизнес-правил и взаимодействие с базой данных или другими источниками данных.

Представление отвечает за отображение данных пользователю. Это может быть «HTML»-страница, шаблон или другой элемент пользовательского интерфейса. Представление обычно содержит «HTML»-разметку, а также механизмы для вставки динамических данных из модели.

Контроллер управляет взаимодействием между моделью и представлением. Он получает запросы от пользователя, обрабатывает их, взаимодействует с моделью для получения необходимых данных и передает эти данные в представление для отображения пользователю. Контроллер также отвечает за обработку пользовательских действий и управление переходами между различными представлениями.

«Entity Framework» («EF») — это набор инструментов и библиотек, разработанных «Microsoft», который облегчает доступ и работу с базами данных в приложениях «.NET». Он предоставляет возможность работать с данными в виде объектов и автоматически выполняет маппинг (отображение) этих объектов на соответствующие таблицы в базе данных. Это позволяет использовать объектно-ориентированный подход при работе с данными, вместо прямой работы с запросами. В «Entity Framework» каждая сущность представляет собой класс или объект, который отображается на таблицу в базе данных. Каждое свойство сущности соответствует столбцу в таблице, а отношения между сущностями отображаются с помощью свойств и навигационных свойств.

Для сохранения данных в базе данных существует широкий выбор различных систем управления базами данных (СУБД), каждая из которых имеет свои особенности и применение. К самым популярным относят:

* «MySQL»;
* «PostgreSQL»;
* «MongoDB»;
* «SQLite»;
* «Microsoft SQL Server»;
* «Oracle Database».

Для обеспечения надежного и безопасного хранения пользовательских данных было выбрано «Microsoft SQL Server» («MSSQL») — коммерческая реляционная система управления базами данных (СУБД), разработанная компанией «Microsoft». «MSSQL» известна своей высокой производительностью, надежностью и масштабируемостью, что делает ее превосходным выбором для хранения, организации и управления структурированными данными.

«Visual Studio» — это интегрированная среда разработки, разработанная компанией «Microsoft». Она предоставляет разработчикам широкий спектр инструментов и функциональности для создания приложений различных типов, включая веб-приложения, мобильные приложения, настольные приложения и многое другое.

Таким образом, предметная область связана с проектированием, разработкой, внедрением и эксплуатацией информационной системы, которая позволяет автоматизировать тестирование в автошколе.

## 2 ПРОЕКТИРОАНИЕ ПРОГРАММНОГОПРОДУКТА

### **2.1** **Концептуальное проектирование**

Разработка любого программного продукта подразумевает его моделирование и проектирование.

В процессе проектирования программного средства после изучения предметной области по теме индивидуального задания были разработаны:

* диаграмма «IDEF0»;
* декомпозиция первого уровня диаграммы «IDEF0»;
* диаграмма потоков данных («DFD»).

Данные диаграммы предназначены для определения структуры будущей базы данных, последовательности выполнения процесса, а также последовательности работы программного средства.

Диаграмма «IDEF0» представляет собой автоматизируемый бизнес-процесс в общем виде. Она состоит из:

* входных данных;
* выходных данных;
* блока бизнес-процесса;
* аппарата управления (документы, контролирующие выполнение бизнес-процесса);
* механизма решения бизнес-процесса (сотрудник или сотрудники, выполняющие бизнес-процесс).

Диаграмма «IDEF0» представлена на рисунке 2.1.

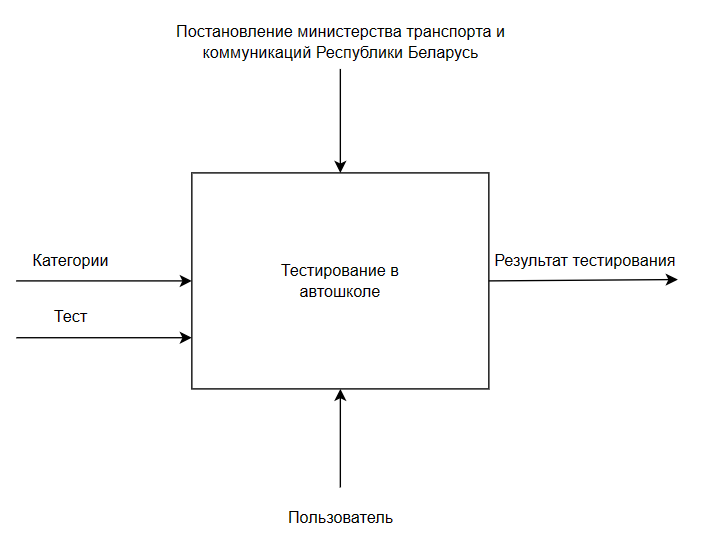


Рисунок 2.1 — Диаграмма «IDEF0»

Исходя из темы «ТЕМА» в качестве входных данных выступают категории и тест, в качестве выходных данных выступают результаты теста. Механизмом служит — администратор и пользователь. Управляет данным процессом — «Постановление министерства транспорта и коммуникаций Республики Беларусь».

Декомпозиция первого уровня диаграммы «IDEF0» представлена на рисунке 2.2.

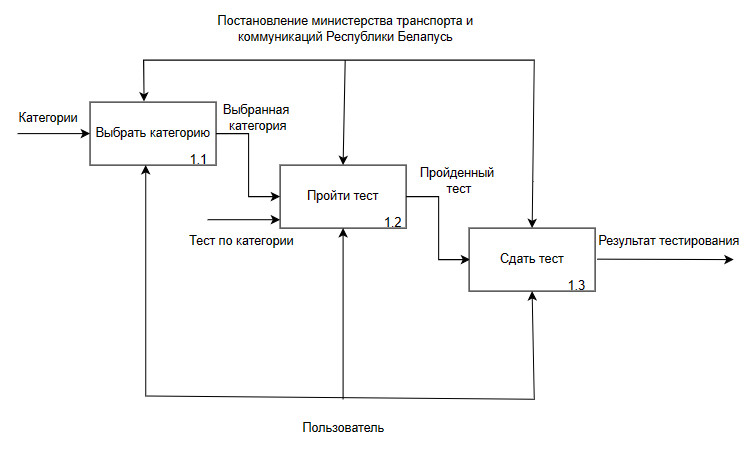


Рисунок 2.2 — Декомпозиция первого уровня диаграммы «IDEF0»

На данной диаграмме более подробно отражен процесс тестирования в автошколе. На диаграмме отражены четыре основных функциональных блока и связи между ними. Они отражают задачи и функции моделируемой системы, которые происходят в течении определенного времени и имеют распознаваемые результаты, которые являются исходными данными для последующих блоков во всей цепочке программного приложения.

1. Функциональный блок «Выбрать категорию»: входные параметры — категории, выходные параметры — выбранная категория.

2. Функциональный блок «Пройти тест»: входные параметры — тест, выходные параметры — пройденный тест.

1. Функциональный блок «Сдать тест»: входные параметры —пройденный тест, вы4ходные параметры — результат.

«DFD» — это нотация, которая используется при моделировании информационных систем с точки зрения хранения, обработки и передачи данных.

Основными компонентами диаграмм потоков данных являются:

* внешние сущности;
* системы и подсистемы;
* процессы;
* накопители данных;
* потоки данных.

К главным преимуществам можно отнести:

* способность нотации точно определить внешние сущности, при этом используя анализ потоков информации внутри и за пределами системы;
* способность проектирование сверху вниз;
* описание процессов нижнего уровня.

Диаграмма потоков данных «DFD» представлена на рисунке 2.3.

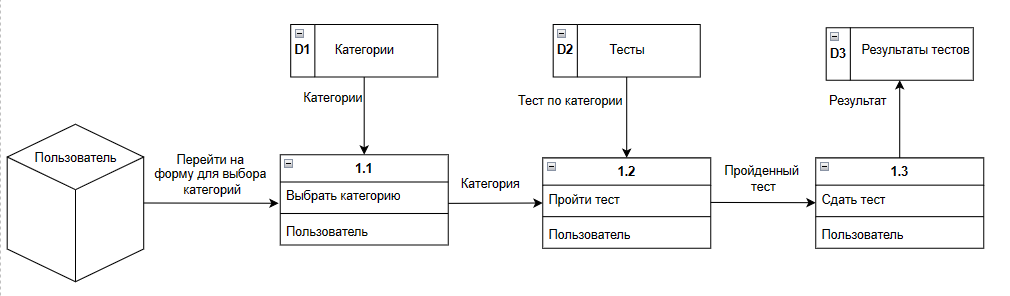


Рисунок 2.3 — Диаграмма потоков данных «DFD»

Данная диаграмма отображает процесс тестирования в автошколе. В качестве функциональных блоков используются блоки «Выбрать категорию», «Пройти тест», «Сдать тест». Для описания потоков данных использовались хранилища «Категории», «Тесты», «Результаты тестов». Тестирование в автошколе осуществляет с помощью реализованных функций.

В ходе построения диаграмм был подробно описан ход работы для облегчения создания программы.

## 3 РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ «ТЕМА»

### **3.1 Графический интерфейс программы**

Графический пользовательский интерфейс — это способ представления информации на компьютере, который позволяет пользователям взаимодействовать с программным обеспечением через графические элементы, такие как окна, кнопки, поля ввода, флажки, списки вместо того, чтобы вводить текстовые команды в командной строке.

На рисунке 3.1 представлен макет страниц разделов.

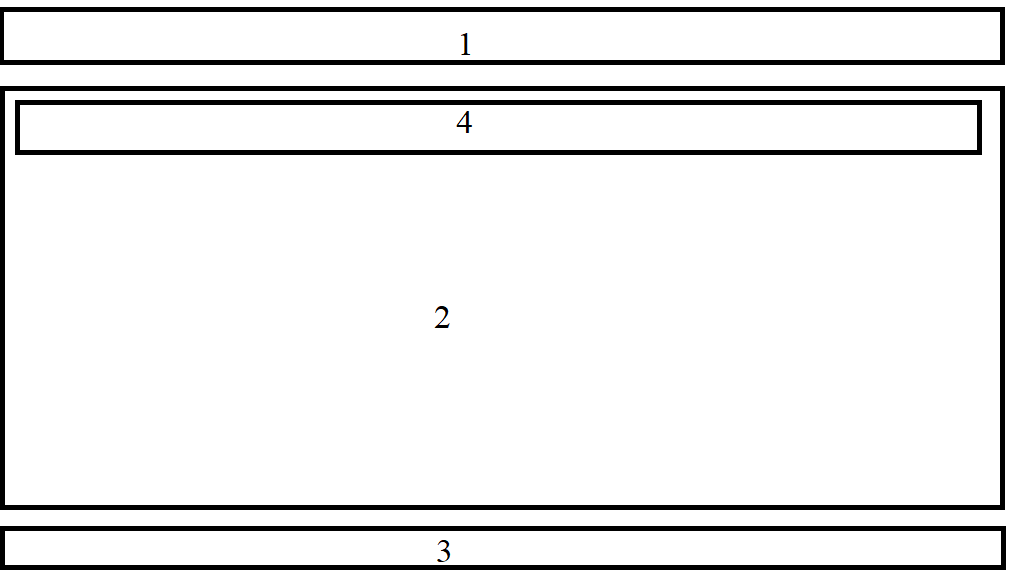


Рисунок 3.1 — Макет страниц разделов; 1— шапка; 2 — блок контента; 3— блок дополнительной информации; 4— блок навигации и конопок;

На рисунке 3.2 представлен макет страниц тестов.

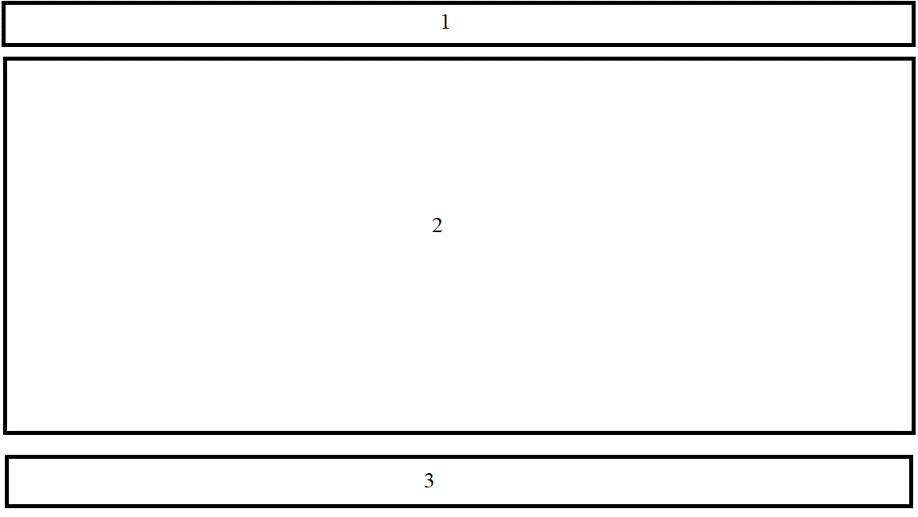


Рисунок 3.2 — Макет страниц тестов; 1— шапка; 2 — блок контента; 3— блок дополнительной информации;

На рисунке 3.3 представлен макет страниц регистрации, входа, личного кабинеты, добавления и редактирования тестов и вопросов.

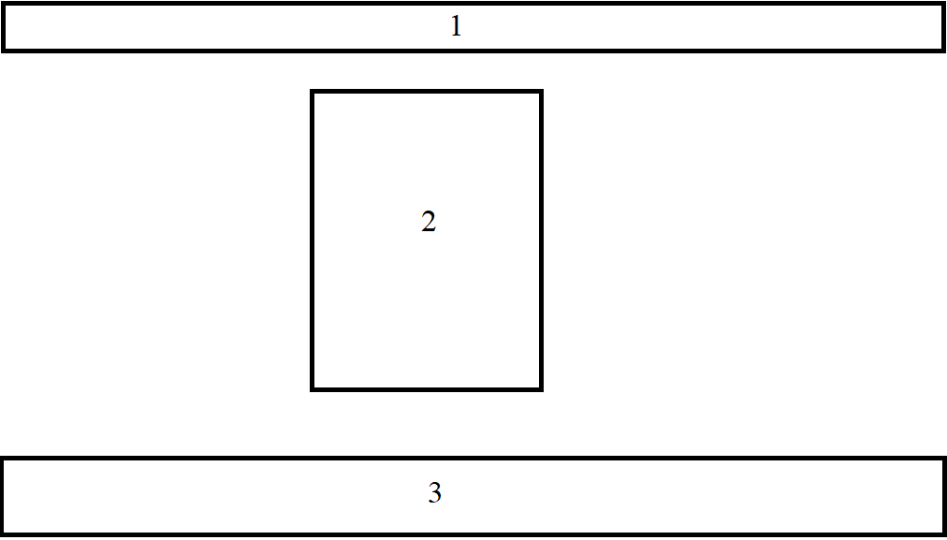


Рисунок 3.3 — Макет страниц регистрации, входа, личного кабинеты, добавления и редактирования тестов и вопросов и тестов; 1— шапка; 2 — блок контента; 3— блок дополнительной информации;

На рисунке 3.4 представлен макет страниц просмотра всех тестов, вопросов и пользователей.

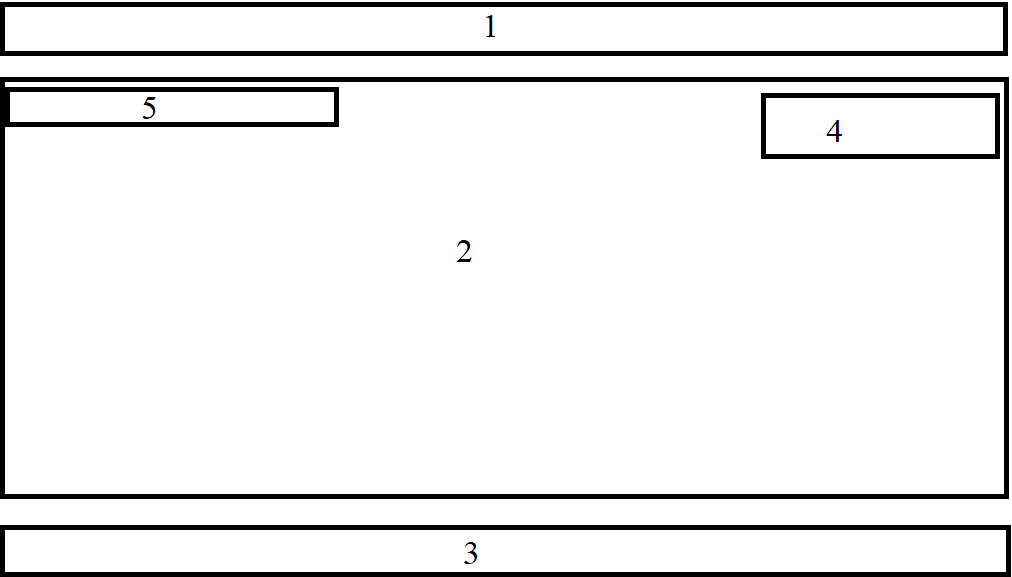


Рисунок 3.4 — Макет страниц просмотра всех тестов, вопросов и пользователей; 1— шапка; 2 — блок контента; 3— блок дополнительной информации; 4— блок дополнительной обработки; 5— блок кнопок;

На рисунке 3.5 представлен графический пользовательский интерфейс первой страницы веб-приложения. На странице представлены разделы с темами для изучения и тестирования. На странице есть возможно перехода на страницу регистрации или личный кабинет и на страницы разделов. На первой странице для реализации интерфейса были использованы следующие компоненты:

* ссылки: Ссылки (или гиперссылки) позволяют пользователям переходить по различным страницам или ресурсам в Интернете. Они могут быть отформатированы как текст или изображения, и обычно имеют подчеркивание или изменение цвета для обозначения того, что они являются кликабельными. При нажатии на ссылку пользователь перенаправляется на целевую страницу или ресурс.
* изображения: Изображения используются для визуального представления информации или элементов интерфейса. Они могут быть использованы для создания логотипов, кнопок, иллюстраций и других элементов дизайна. Изображения также могут быть кликабельными и использоваться в качестве ссылок, например, для открытия полноразмерной версии изображения или перехода на связанный ресурс.

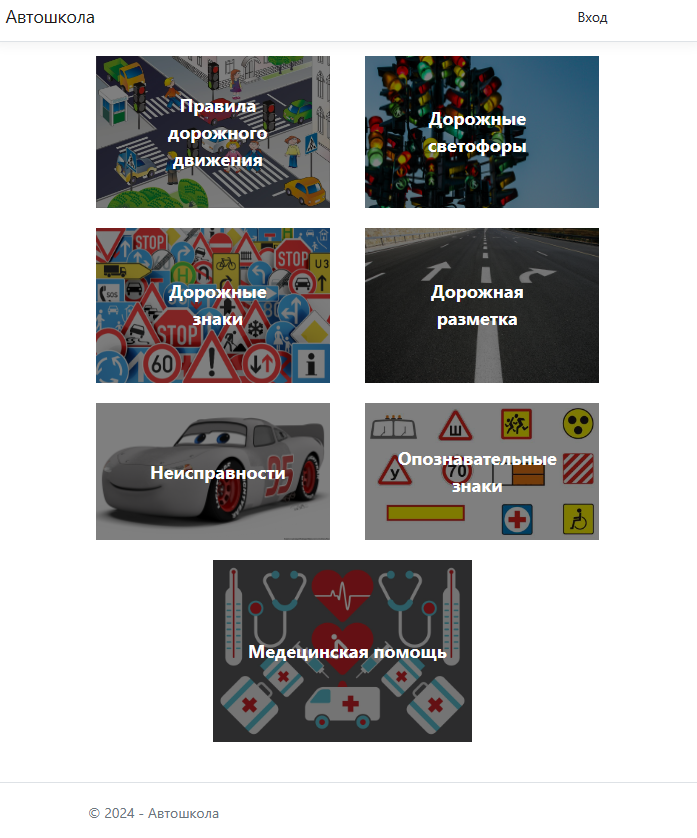


Рисунок 3.5 — Графический пользовательский интерфейс первой страницы

На рисунке 3.6 представлена страница входа в аккаунт. На странице есть возможность: войти в аккаунт, перейти на страницу регистрации и вернется на первую страницу. На странице входа в аккаунт для реализации интерфейса были использованы следующие компоненты:

* ссылки;
* форма — это структурированный набор компонентов интерфейса, который позволяет пользователям вводить и отправлять данные. Формы широко применяются на веб-страницах и в приложениях для сбора информации от пользователей;
* метки: Метки (или надписи) используются для отображения текстовой информации, которая является описательной или идентифицирующей для других элементов на странице. Они могут содержать заголовки, подписи к полям ввода, инструкции или любой другой текст, который помогает пользователю понять, что ожидается от них на странице;
* поля ввода: Поля ввода позволяют пользователям вводить текстовую или числовую информацию. Они могут быть использованы для получения данных от пользователя, таких как имя, адрес, электронная почта и т.д. Поля ввода могут быть однострочными или многострочными, в зависимости от требований для ввода данных;
* кнопки: Кнопки предоставляют пользователю возможность выполнить определенное действие при нажатии на них. Они могут использоваться для отправки форм, перехода на другие страницы, вызова функций или выполнения любых других операций, определенных в интерфейсе. Кнопки обычно содержат текст или иконку, которая ясно указывает на предполагаемое действие;
* заголовок первого уровня — это элемент «HTML», который используется для создания наиболее выразительного и визуально привлекательного заголовка на веб-странице.

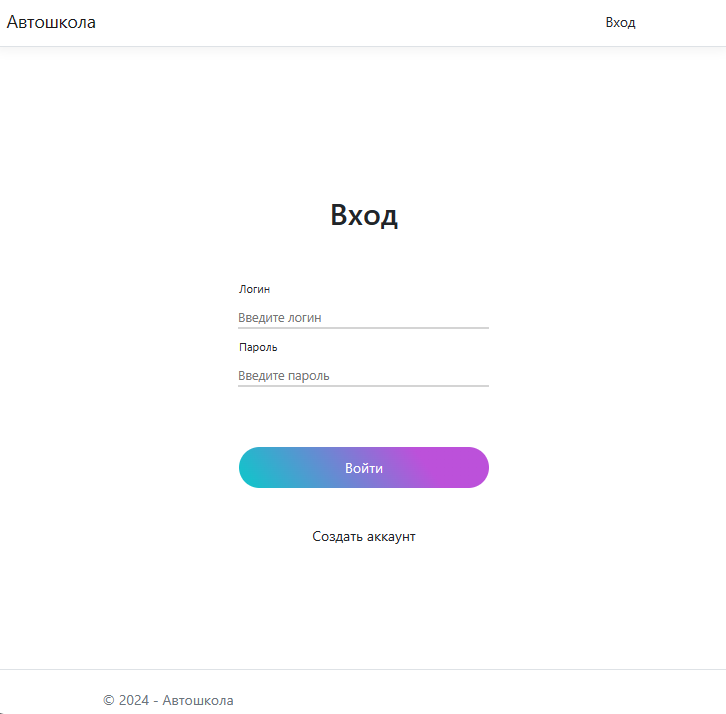


Рисунок 3.6 — Графический пользовательский интерфейс страницы входа в аккаунт

На рисунке 3.7 представлена страница регистрации. На странице есть возможность: зарегистрироваться, вернутся на страницу входа в аккаунт и на первую страницу. На странице регистрации для реализации интерфейса были использованы следующие компоненты:

* ссылки;
* форма;
* метки;
* поля ввода;
* кнопки;
* заголовок первого уровня.

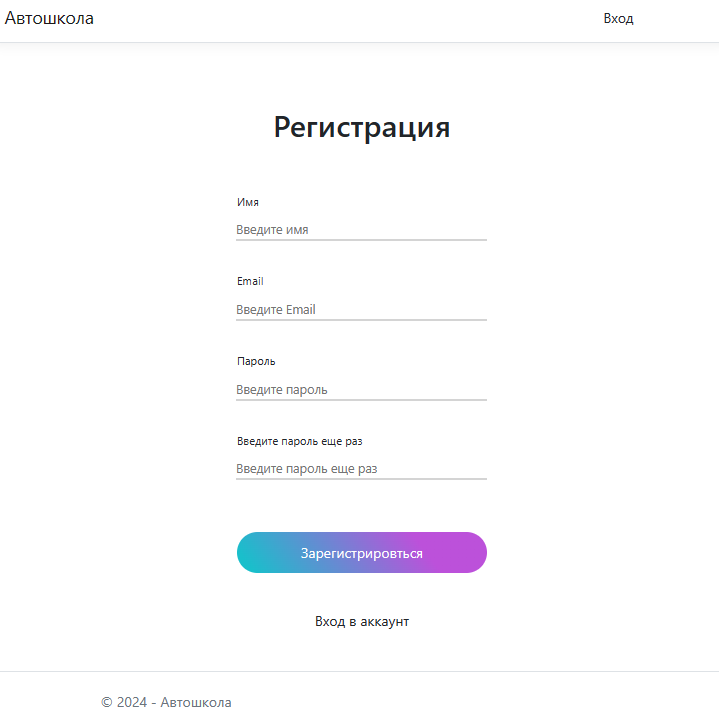


Рисунок 3.7 — Графический пользовательский интерфейс страницы регистрации

На рисунке 3.8 представлена страница «Личный кабинет». На странице есть возможность: просмотра информации о сданных экзаменах, изменения имени и пароля, выхода из аккаунта, перехода на первую страницу.

На странице «Личный кабинет» для реализации интерфейса были использованы следующие компоненты:

* ссылки;
* форма;
* метки;
* поля ввода;
* кнопки;
* заголовок первого уровня.

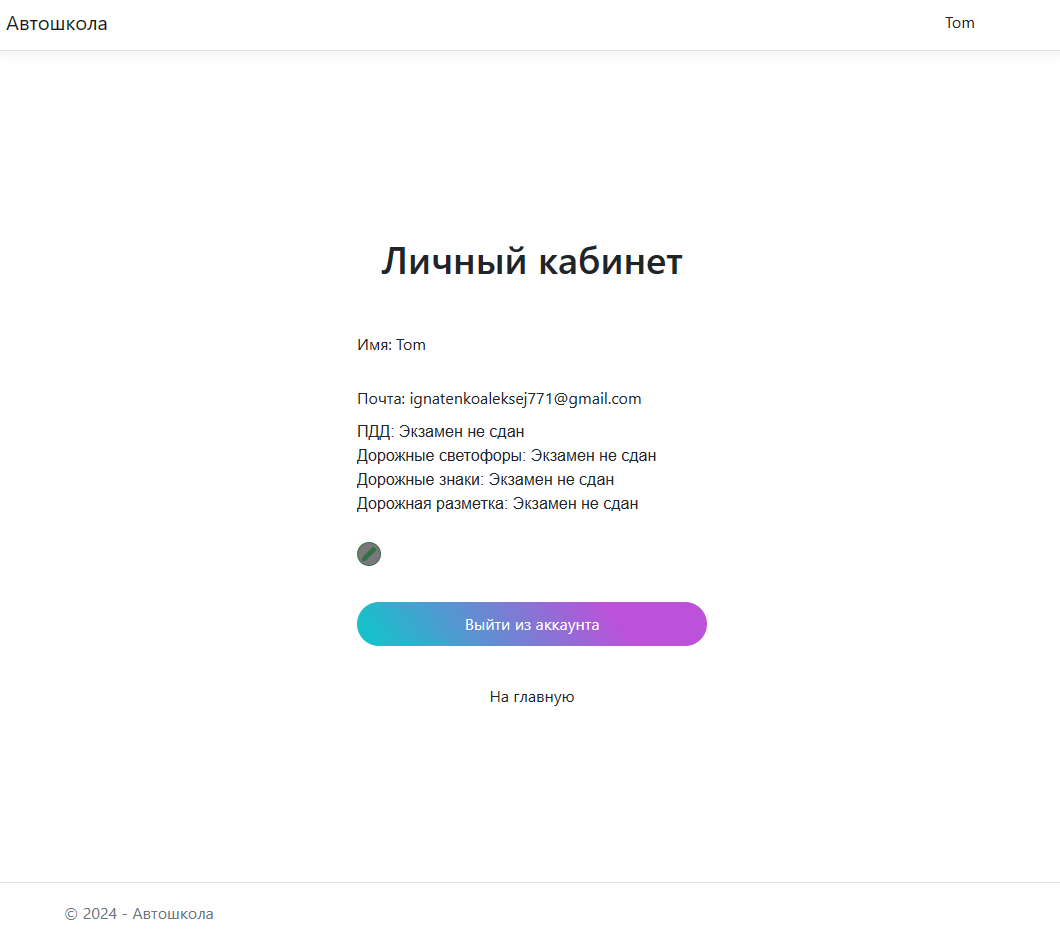


Рисунок 3.8 — Графический пользовательский интерфейс страницы «Личный кабинет»

На рисунке 3.9 представлена страница раздела одной из категорий. На странице есть возможность: перейти к необходимому разделу, перейти на страницу входа в аккаунт или личный кабинет, перейти на первую страницу и страницу с тестом. На странице категорий для реализации интерфейса были использованы следующие компоненты:

* ссылки;
* поля ввода;
* заголовок первого уровня;
* заголовок второго уровня: заголовок второго уровня является элементом «HTML», предназначенным для создания второстепенного заголовка на веб-странице. Он обычно используется для подзаголовков или разделов, которые следуют после заголовка первого уровня. Заголовок второго уровня имеет меньший размер шрифта, чем заголовок первого уровня, и помогает организовать информацию на странице.
* заголовок пятого уровня: заголовок пятого уровня — это элемент «HTML», который используется для создания заголовка пятого уровня на веб-странице. заголовок пятого уровня обычно используется для заголовков низшего уровня или для создания дополнительной иерархии заголовков внутри разделов, описанных заголовком второго уровня или третьего пятого. Заголовок пятого уровня имеет еще меньший размер шрифта, чем заголовок второго уровня, и может быть использован для создания более специфических подзаголовков.
* параграф: представляет собой элемент «HTML», который используется для создания абзацев или текстовых блоков на веб-странице. Параграф позволяет организовывать текстовую информацию в виде параграфов, разделяя их на логические блоки. Это основной элемент для отображения обычного текста на веб-странице.

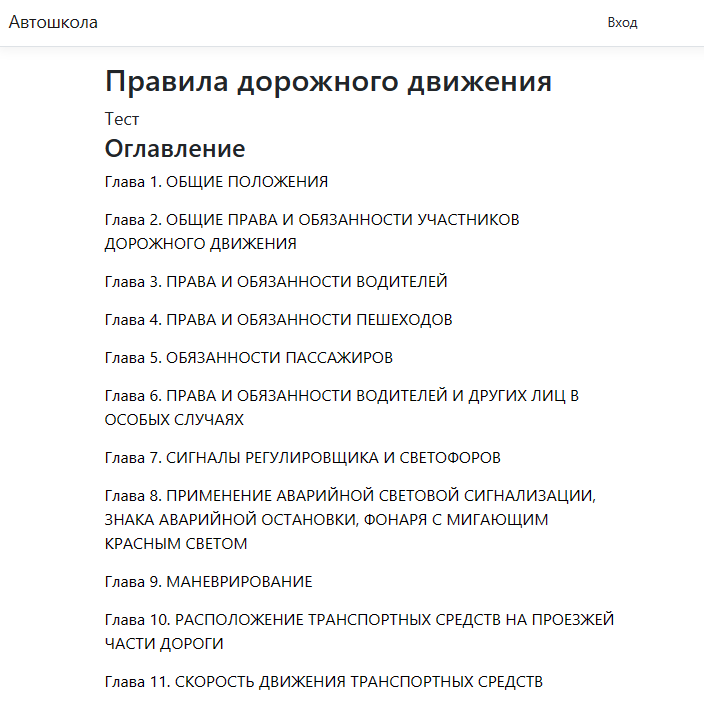


Рисунок 3.9 — Графический пользовательский интерфейс страницы раздела «Правила дорожного движения»

На рисунке 3.10 представлена страница теста. На странице есть возможность: пройти тест, перейти на страницу «Личный кабинет» или перейти на первую страницу. На странице теста для реализации интерфейса были использованы следующие компоненты:

* поля ввода;
* форма;
* метки;
* кнопки.
* заголовок 3 уровня — обычно используется для подразделов или дополнительных заголовков внутри разделов, указанных с помощью заголовков второго или первого уровня. Заголовок 3 уровня имеет меньший размер шрифта, чем заголовок второго уровня, и помогает создать более узкую иерархию заголовков.

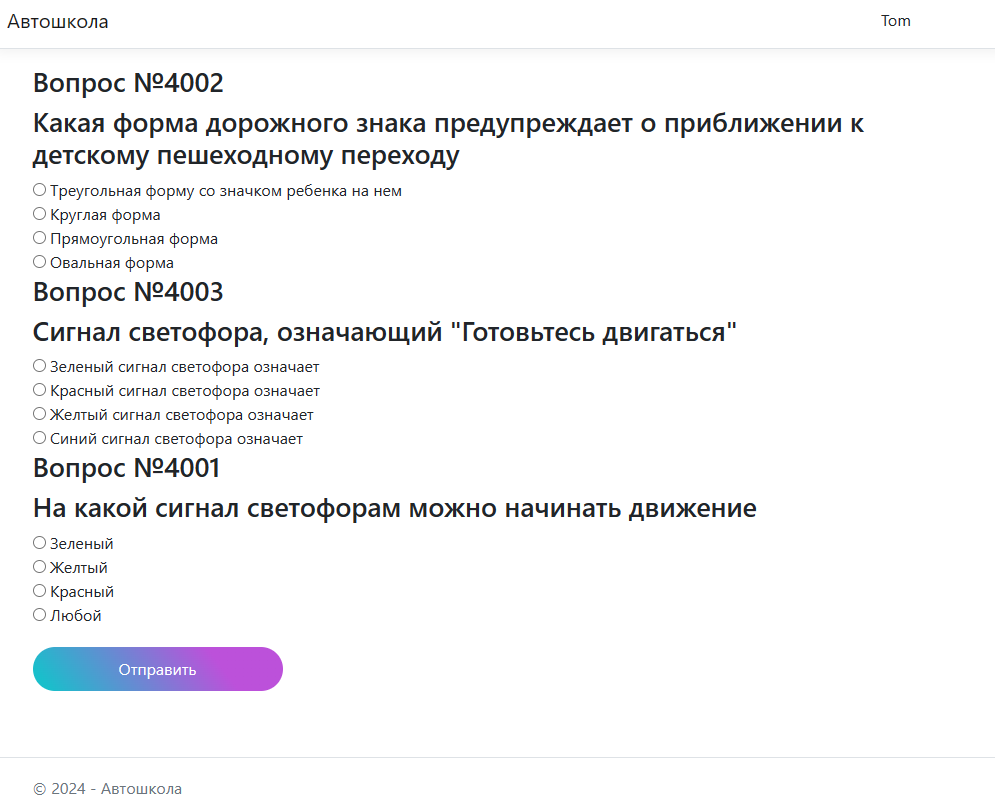


Рисунок 3.10 — Графический пользовательский интерфейс страницы раздела теста

У пользователей с правами «Admin» есть возможность создавать, обновлять и удалять тесты и вопросы в тестах. На рисунке 3.11 представлена страница «Тесты». На странице есть возможность: перейти на страницу для добавления теста, перейти на страницу для добавления вопроса в тест, найти тест, отсортировать тесты в алфавитном порядке или по номеру, отменить сортировку, удалить тест, отредактировать тест, посмотреть вопросы теста и перейти в личный кабинет или на первую страницу, найти информацию на странице и отсортировать по номеру. На странице «Тесты» для реализации интерфейса были использованы следующие компоненты:

* ссылки;
* кнопки;
* поля ввода;
* заголовок первого уровня;
* форма;
* метки.

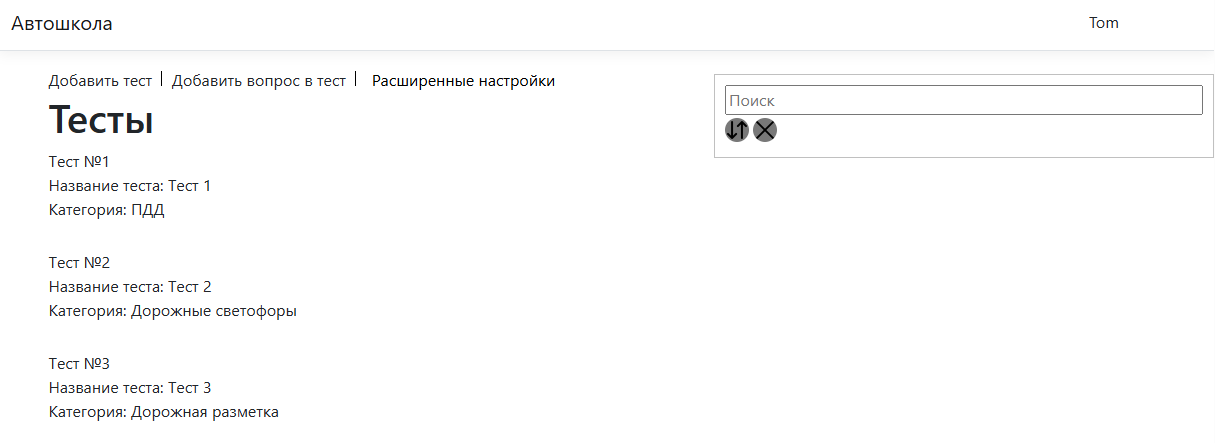


Рисунок 3.11 — Графический пользовательский интерфейс страницы «Тесты»

На рисунке 3.12 представлена страница «Добавление теста». На странице есть возможность: добавить тест, перейти на страницу «Тесты», перейти в личный кабинет или на первую страницу. На странице «Добавление теста» для реализации интерфейса были использованы следующие компоненты:

* ссылки;
* кнопки;
* поля ввода;
* заголовок первого уровня;
* форма;
* метки.

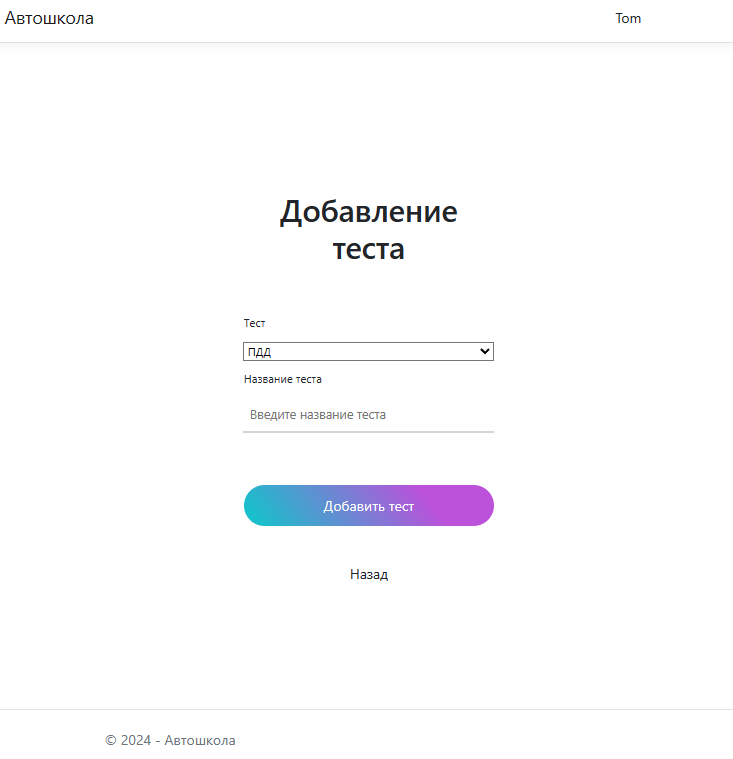


Рисунок 3.12 — Графический пользовательский интерфейс страницы

«Добавление теста»

На рисунке 3.13 представлена страница «Добавить вопрос». На странице есть возможность: добавить вопрос, перейти на страницу «Тесты», перейти в личный кабинет или на первую страницу. На странице «Добавление вопроса» для реализации интерфейса были использованы следующие компоненты:

* ссылки;
* кнопки;
* поля ввода;
* заголовок первого уровня;
* форма;
* метки.

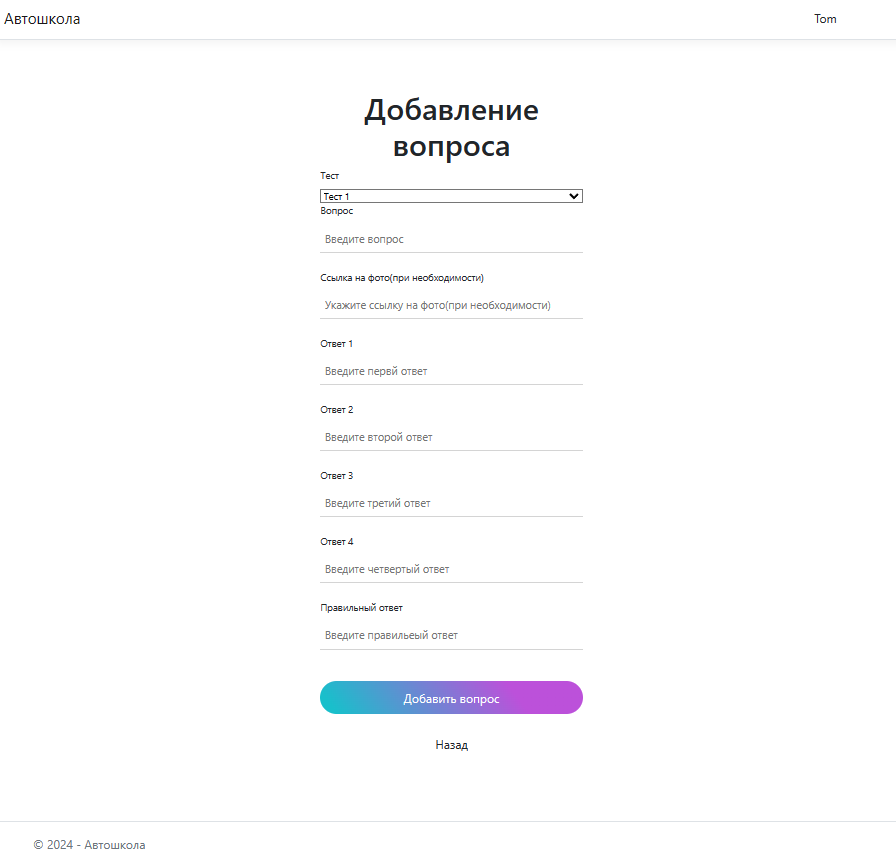


Рисунок 3.13 — Графический пользовательский интерфейс страницы

«Добавить вопрос»

На рисунке 3.14 представлена страница «Редактирования теста». На странице есть возможность: внести изменения и сохранить их, перейти на страницу «Тесты», перейти в личный кабинет или на первую страницу. На странице «Редактирование теста» для реализации интерфейса были использованы следующие компоненты:

* ссылки;
* кнопки;
* поля ввода;
* заголовок первого уровня;
* форма;
* метки.

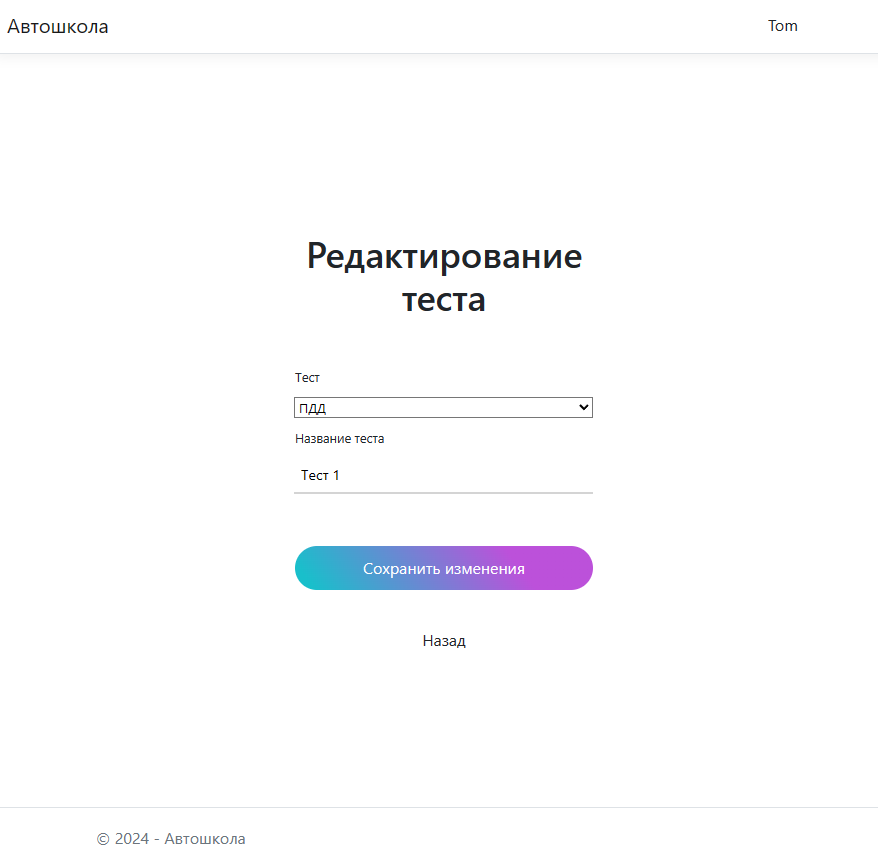


Рисунок 3.14 — Графический пользовательский интерфейс страницы

«Редактирования теста»

На рисунке 3.15 представлена страница с вопросами выбранного теста. На странице есть возможность: удалить вопрос, отредактировать вопрос, найти вопрос, отсортировать вопросы в алфавитном порядке или по номеру, отменить сортировку, перейти в личный кабинет или на первую страницу, найти информацию на странице и отсортировать по номеру. На странице «Добавление теста» для реализации интерфейса были использованы следующие компоненты:

* ссылки;
* кнопки;
* поля ввода;
* заголовок первого уровня;
* форма;
* метки.



Рисунок 3.15 — Графический пользовательский интерфейс страницы

с вопросами выбранного теста

На рисунке 3.16 представлена страница с «Редактирование вопроса». На странице есть возможность: внести и сохранить все изменения, перейти в личный кабинет или на первую страницу. На странице «Редактирование вопроса» для реализации интерфейса были использованы следующие компоненты:

* ссылки;
* кнопки;
* поля ввода;
* заголовок первого уровня;
* форма;
* метки.

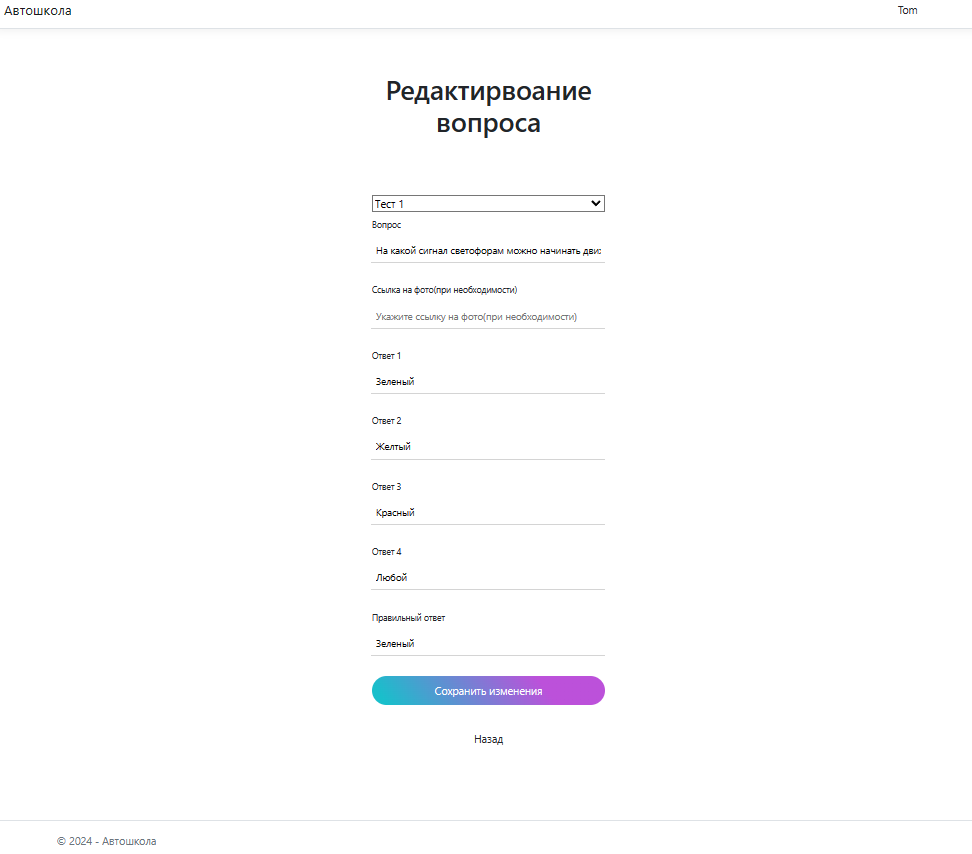


Рисунок 3.16 — Графический пользовательский интерфейс страницы

«Редактирование вопроса»

На рисунке 3.17 представлена страница с «Менеджер». На странице есть возможность: изменить имя пользователю, получить информацию о результатах теста, выдать и забрать права администратора, перейти в личный кабинет или на первую страницу, найти информацию на странице и отсортировать по номеру. На странице «Менеджер» для реализации интерфейса были использованы следующие компоненты:

* ссылки;
* кнопки;
* поля ввода;
* заголовок первого уровня;
* форма;
* метки;
* список— используется для создания неупорядоченного списка. Неупорядоченный список представляет собой список элементов, которые не имеют определенного порядка или последовательности. Элементы списка отображаются с маркерами (обычно точками, кружками или квадратами) для обозначения каждого элемента.

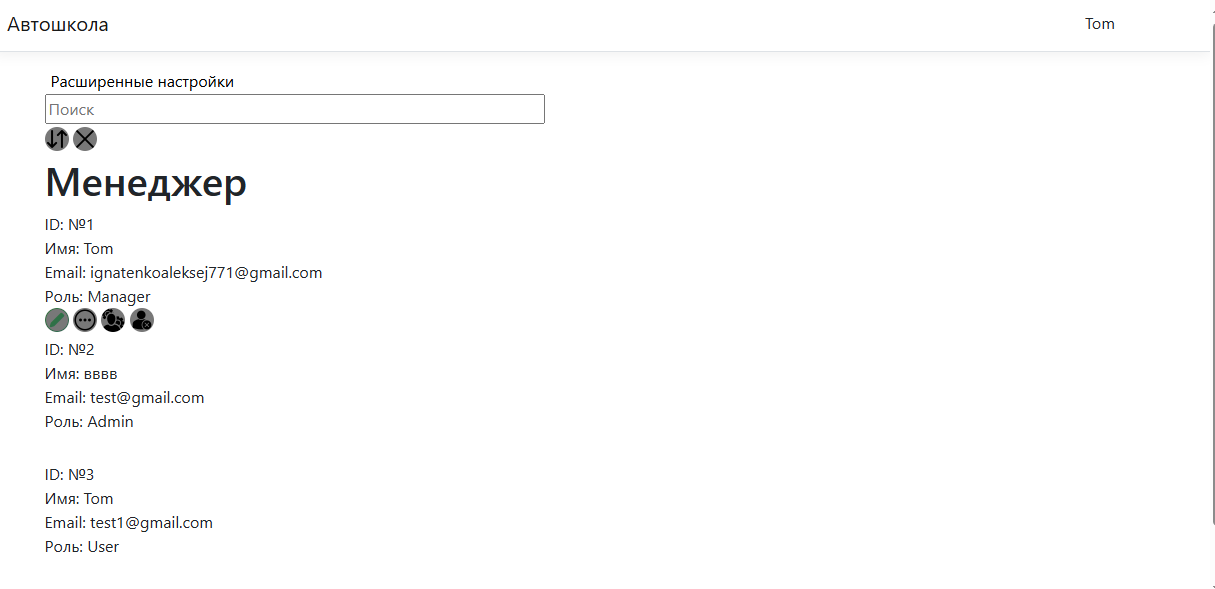


Рисунок 3.17 — Графический пользовательский интерфейс страницы

«Редактирование вопроса»

На рисунке 3.18 представлена страница с «Менеджер». На странице есть возможность: изменить имя пользователю, получить информацию о результатах теста, выдать и забрать права администратора, перейти в личный кабинет или на первую страницу, найти информацию на странице и отсортировать по номеру. На странице «Менеджер» для реализации интерфейса были использованы следующие компоненты:

* ссылки;
* кнопки;
* поля ввода;
* заголовок первого уровня;
* форма;
* метки;
* список.

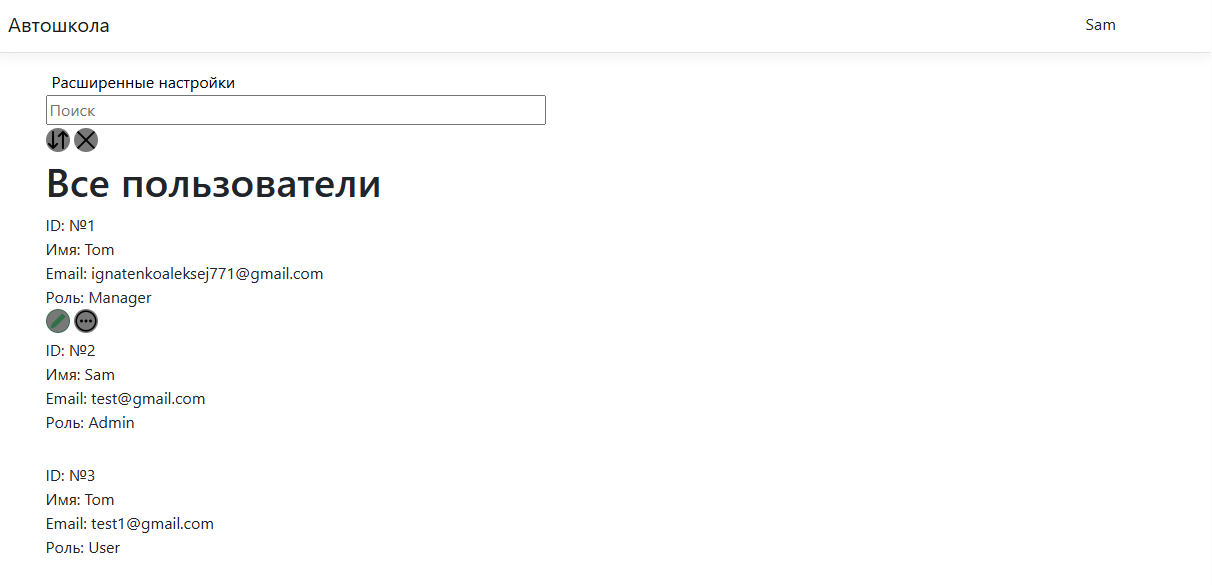


Рисунок 3.18 — Графический пользовательский интерфейс страницы

«Редактирование вопроса»

### **3.2 Логическая модель данных**

База данных — это совокупность данных, организованных по определённым правилам. Она представляет собой набор информации, которая хранится упорядоченно в электронном виде.

Обычно база данных представляется в виде совокупности взаимосвязанных файлов или таблиц, предназначенных для решения конкретной задачи.

Схема системы базы данных — это план или архитектура того, как будут выглядеть наши данные, он не содержит самих данных, а вместо этого описывает форму данных и то, как они могут быть связаны с другими таблицами или моделями.

Таблица «AnswerUserTest» используется для хранения ответов пользователей.

Поля, которые находятся в этой таблице, показаны в таблице 3.1.

Таблица 3.1 — Структура таблицы «AnswerUserTest»

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных поля | Описание |
| IdAnswerUser | Числовой | Первичный ключ, счетчик |
| UserId | Числовой | Вторичный ключ, связь с таблицей «User». |
| TestId | Числовой | Вторичный ключ, связь с таблицей «Test». |
| ResultTest | Числовой | Результат теста |

Таблица «Category» используется для хранения категорий.

Поля, которые находятся в этой таблице, представлены в таблице 3.2.

Таблица 3.2 — Структура таблицы «Category»

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных поля | Описание |
| IdCategory | Числовой | Первичный ключ, счетчик |
| NameCategory | Текстовый | Название категории |

Таблица «Question» используется для хранения вопросов тестов.

Поля, которые находятся в этой таблице, показаны в таблице 3.3.

Таблица 3.3 — Структура таблицы «Question»

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных поля | Описание |
| IdQuestion | Числовой | Первичный ключ, счетчик |
| IdTest | Числовой | Вторичный ключ, связь с таблицей «Test». |
| QuestionText | Текстовый | Текст вопроса |
| LinkPhoto | Текстовый | Ссылка на фото |
| Answer1 | Текстовый | Ответ 1 |
| Answer2 | Текстовый | Ответ 2 |
| Answer3 | Текстовый | Ответ 3 |
| Answer4 | Текстовый | Ответ 4 |
| CorrectAnswer | Текстовый | Правильный ответ |

Таблица «Tests» используется для хранения тестов.

Поля, которые находятся в этой таблице, показаны в таблице 3.4.

Таблица 3.4 — Структура таблицы «Tests»

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных поля | Описание |
| IdTest | Числовой | Первичный ключ, счетчик |
| IdCategory | Числовой | Вторичный ключ, связь с таблицей «Category». |
| NameTest | Текстовый | Название теста |

Таблица «Users» используется для пользователей.

Поля, которые находятся в этой таблице, показаны в таблице 3.5.

Таблица 3.5 — Структура таблицы «Users»

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных поля | Описание |
| IdUser | Числовой | Первичный ключ, счетчик |
| UserName | Текстовый | Имя пользователя |
| Email | Текстовый | Почта |
| Password | Текстовый | Пароль |
| Role | Текстовый | Роль |

### **3.3 Реализация функций программы**

Для создания нового аккаунта необходимо заполнить все поля и нажать на кнопку «Зарегистрироваться». Фрагмент кода представлен на рисунке 3.19.

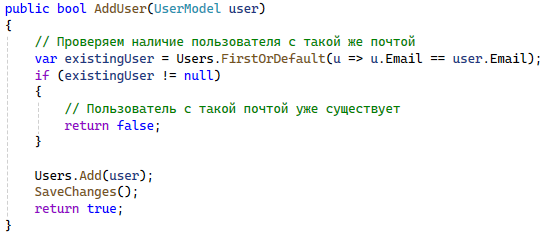


Рисунок 3.19 — Фрагмент код регистрации

Для входа в аккаунт необходимо заполнить все поля и нажать на кнопку «Войти». Фрагмент кода представлен на рисунке 3.20.

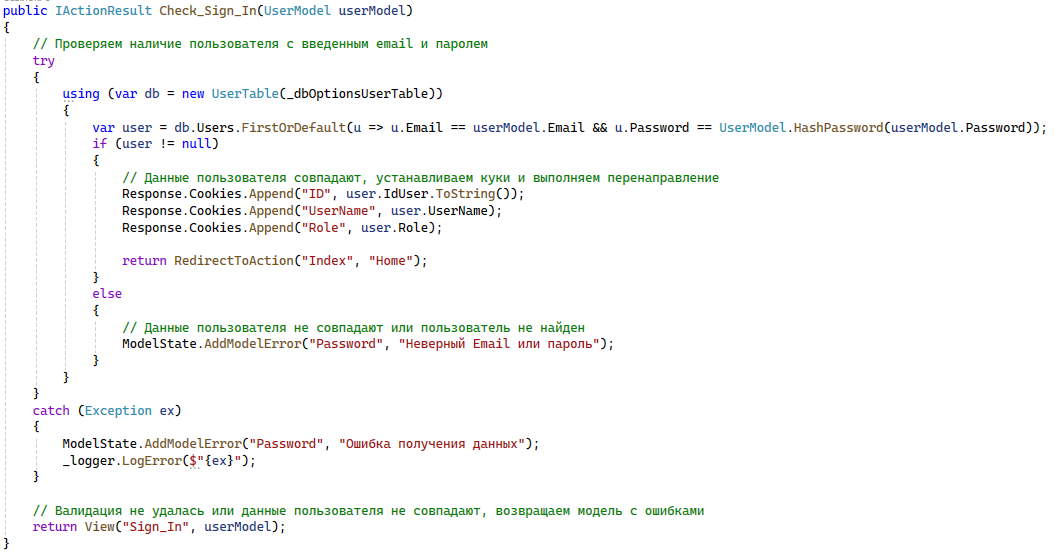


Рисунок 3.20 — Фрагмент код входа в аккаунт

Для выхода из аккаунт необходимо нажать на кнопку «Выйти из аккаунта». Фрагмент кода представлен на рисунке 3.21.

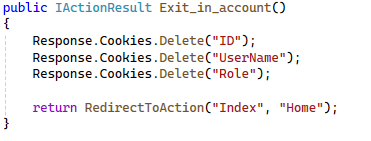


Рисунок 3.21 — Фрагмент код выхода из аккаунта

Для изменения имени пользователя необходимо перейти в личный кабинет нажать на кнопку «Изменить имя», ввести новое имя и нажать на кнопку «Изменить имя». Фрагмент кода изменения имени на рисунке 3.22.

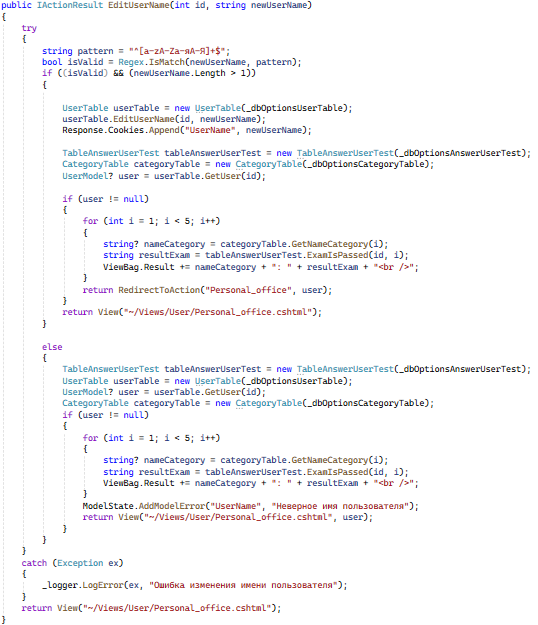


Рисунок 3.22 — Фрагмент код изменения имени

Для прохождения теста необходимо выбрать категорию теста, ответить на вопросы и нажать на кнопку «Сдать». Фрагмент кода тестирования представлен на рисунке 3.23.

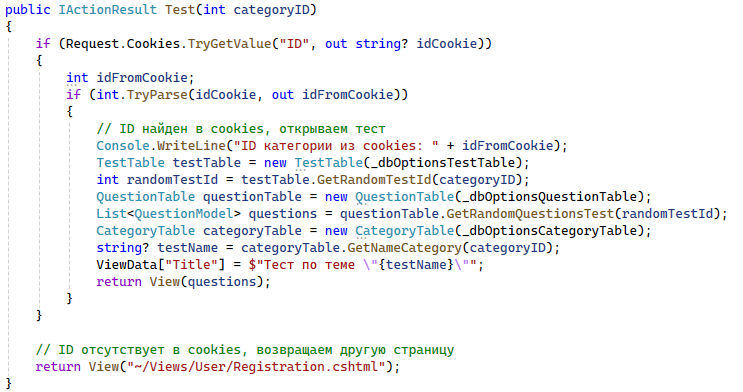


Рисунок 3.23 — Фрагмент кода тестирования

Для добавления теста необходимо перейти в раздел «Тесты», нажать на кнопку «Добавить тест», заполнить поля и нажать на кнопку «Добавить тест». Фрагмент кода добавления тест представлен на рисунке 3.24.



Рисунок 3.24 — Фрагмент кода добавления теста

Для удаления теста необходимо перейти в раздел «Тесты», навестись на тест и нажать на кнопку «Удалит». Фрагмент кода удаления теста представлен на рисунке 3.25.



Рисунок 3.25 — Фрагмент кода удаления теста

Для редактирования теста необходимо перейти в раздел «Тесты», навестись на тест и нажать на кнопку «Редактировать». Фрагмент кода редактирования теста представлен на рисунке 3.26.



Рисунок 3.26 — Фрагмент кода редактирования теста

Для просмотра вопросов теста необходимо перейти в раздел «Тесты», нажать на кнопку «Вопросы теста». Фрагмент кода просмотра вопросов теста представлен на рисунке 3.27.

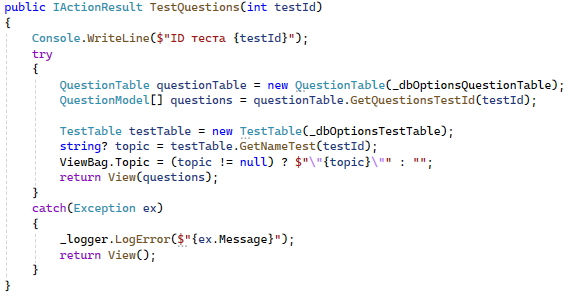


Рисунок 3.27 — Фрагмент кода просмотра вопросов теста

Для добавления вопроса в тест необходимо перейти в раздел «Тесты», нажать на кнопку «Добавить вопрос в тест». Фрагмент кода добавления вопроса в тест представлен на рисунке 3.28.



Рисунок 3.28 — Фрагмент кода добавления вопроса в тест

Для удаления вопроса необходимо перейти в раздел «Тесты», выбрать тест, в котором необходимо удалить вопрос, нажать на кнопку «Вопросы теста» выбрать вопрос и нажать на кнопку «Удалить». Фрагмент кода удаления вопроса из теста представлен на рисунке 3.29.

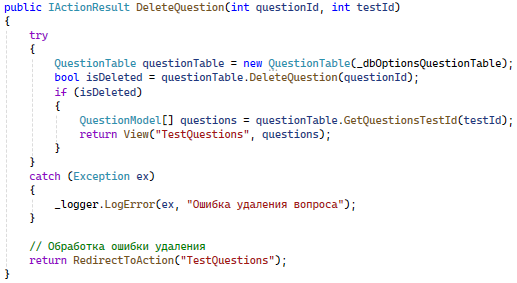


Рисунок 3.29 — Фрагмент кода удаления вопроса из теста

Для редактирования вопроса необходимо перейти в раздел «Тесты», выбрать тест, в котором необходимо отредактировать вопрос, нажать на кнопку «Вопросы теста» выбрать вопрос и нажать на кнопку «Редактировать», внести изменения и нажать на кнопку «Сохранить изменения». Фрагмент кода редактирования вопроса теста представлен на рисунке 3.30.



Рисунок 3.30 — Фрагмент кода редактирования вопроса теста

Для редактирования имени пользователя необходимо перейти в раздел «Пользователи», выбрать пользователя, нажать на кнопку «Изменить имя», ввести новое имя пользователя и нажать на кнопку «Сохранить». Фрагмент кода редактирования имени пользователя представлен на рисунке 3.31.



Рисунок 3.31 — Фрагмент кода изменения имени пользователя

Для просмотра информации о результатах тестирования пользователя необходимо перейти в раздел «Пользователи», выбрать пользователя, нажать на кнопку «Результаты тестирования». Фрагмент кода получения результатов тестирования пользователя представлен на рисунке 3.32.

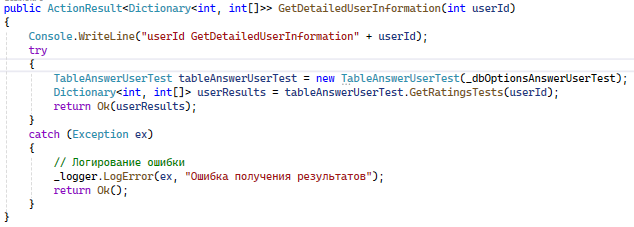


Рисунок 3.32 — Фрагмент кода получения результатов тестирования пользователя

Для выдачи пользователю роли «Admin» необходимо перейти в раздел «Пользователи», выбрать пользователя, нажать на кнопку «Выдать права Администратора» и нажать на кнопку «Выдать права Администратора». Фрагмент кода выдачи роли «Admin» представлен на рисунке 3.33.



Рисунок 3.33 — Фрагмент кода выдачи роли «Admin»

Для удаления у пользователя роли «Admin» необходимо перейти в раздел «Пользователи», выбрать пользователя, нажать на кнопку «Забрать права Администратора» и нажать на кнопку «Забрать права Администратора». Фрагмент кода удаления роли «Admin» представлен на рисунке 3.34.

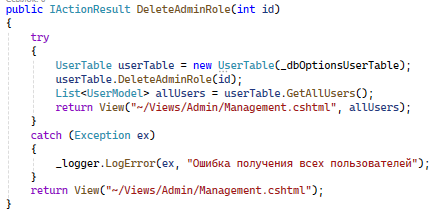


Рисунок 3.34 — Фрагмент кода удаления роли «Admin»

Полный код программы представлен в приложении А.

## 4 ДЕМОНСТРАЦИЯ РАБОТЫ ПРОГРАММЫ

Для начала работы программы необходимо запустить загрузочный файл «Project1.exe», после чего откроется первая форма приложения. На форме находится информация о разработчике, название приложения и кнопки для перехода на основную форму приложения (рисунок 4.1).

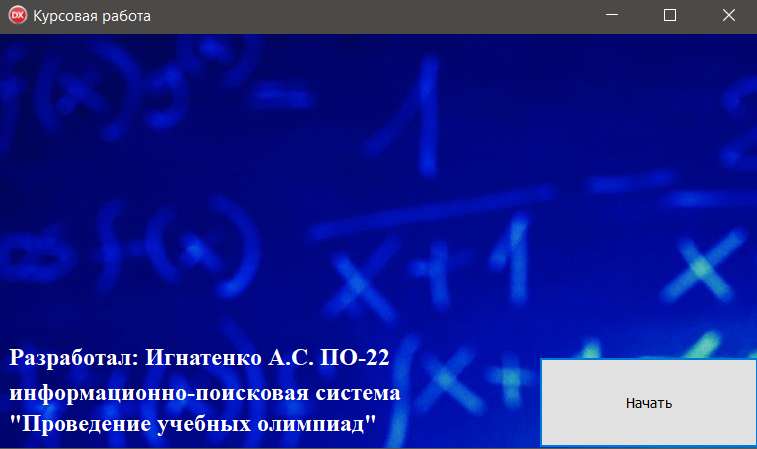


Рисунок 4.1 — Первая форма приложения

Для перехода на основную форму необходимо нажать кнопку «Начать». На рисунке 4.2 показана основная форма приложения.

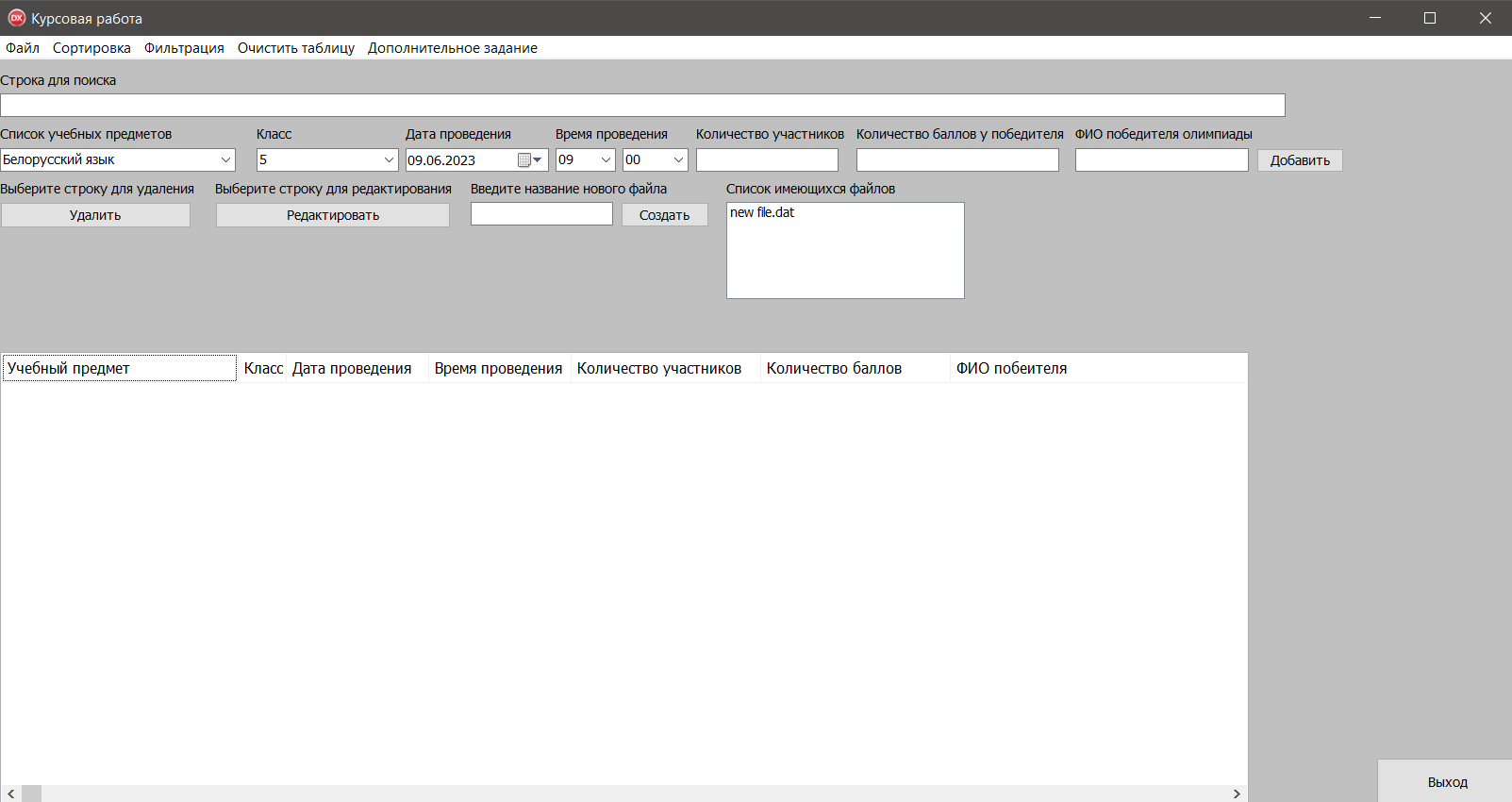


Рисунок 4.2 — Основная форма приложения

Для добавления новой записи в таблицу необходимо выбрать предмет, класс, дату проведения и время проведения, заполнить поля «Количество участников», «Количество баллов у победителя», «ФИО победителя» и нажать на кнопку «Добавить» после чего запись добавится в таблицу. На рисунке 4.3 показана таблица до добавления новой записи.

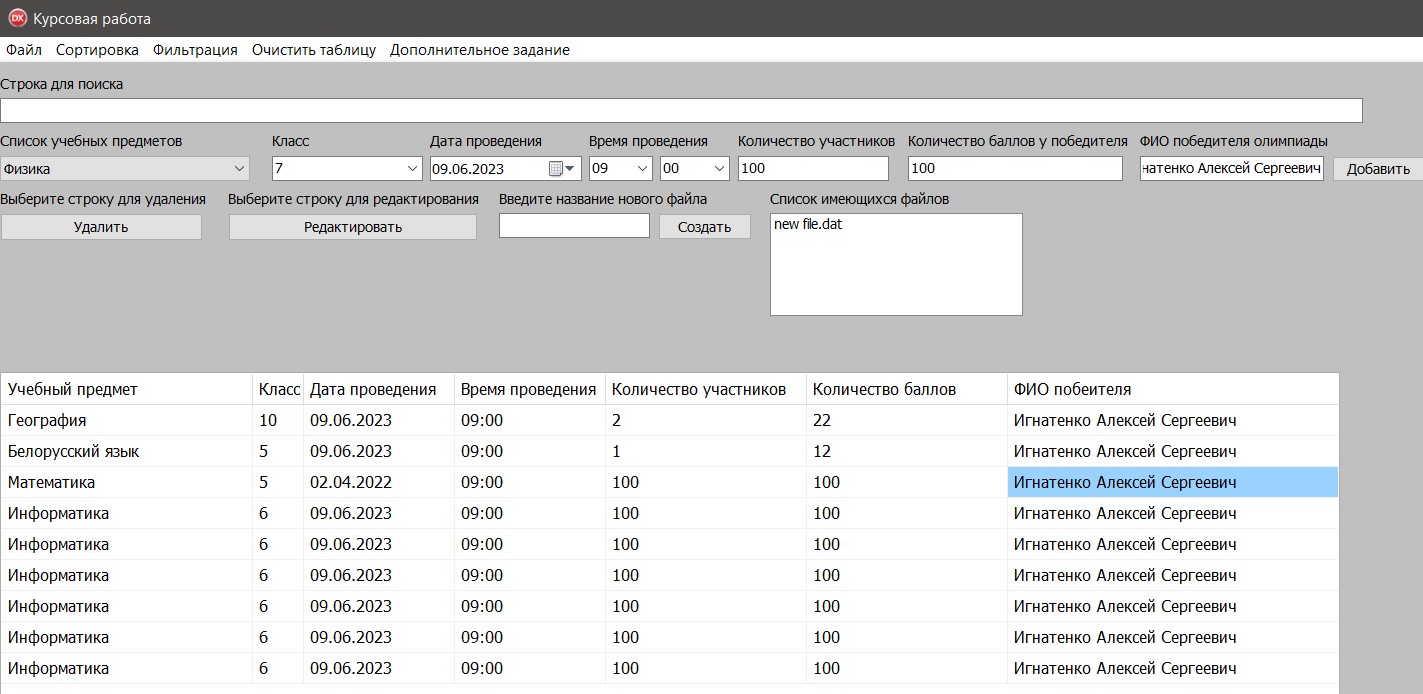


Рисунок 4.3 — Таблица до добавления записи

На рисунке 4.4 показана таблица после добавления новой записи.

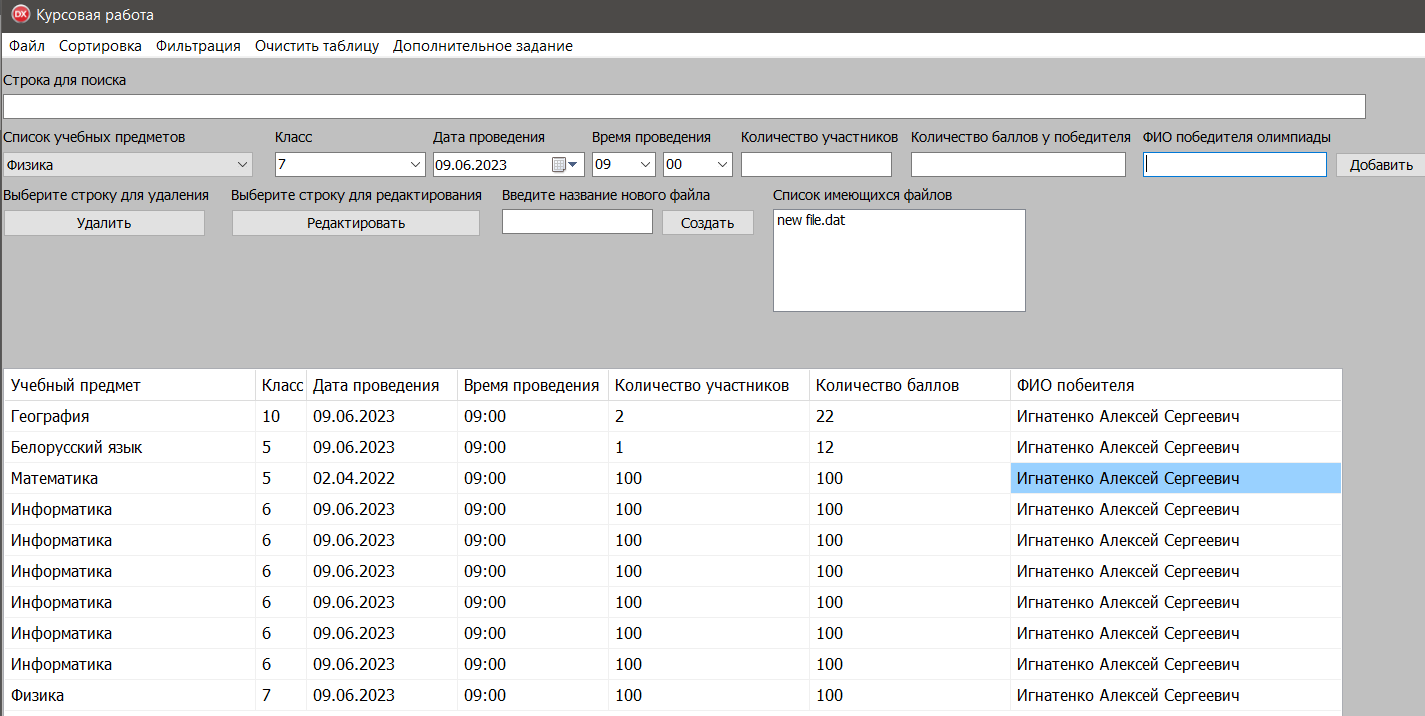


Рисунок 4.4 — Таблица после добавления новой записи

Для удаления записи необходимо выбрать ячейку из строки для удаления и нажать на кнопку «Удалить» после чего появится сообщение, показанное на рисунке 4.5.

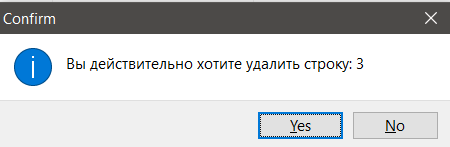


Рисунок 4.5 — Сообщение с предупреждением об удалении сроки и номером удаляемой строки

При нажатии на кнопку «Yes» строка удалится, а при нажатии кнопки «No» строка останется. На рисунке 4.6 показан фрагмент таблицы до удаления строки.

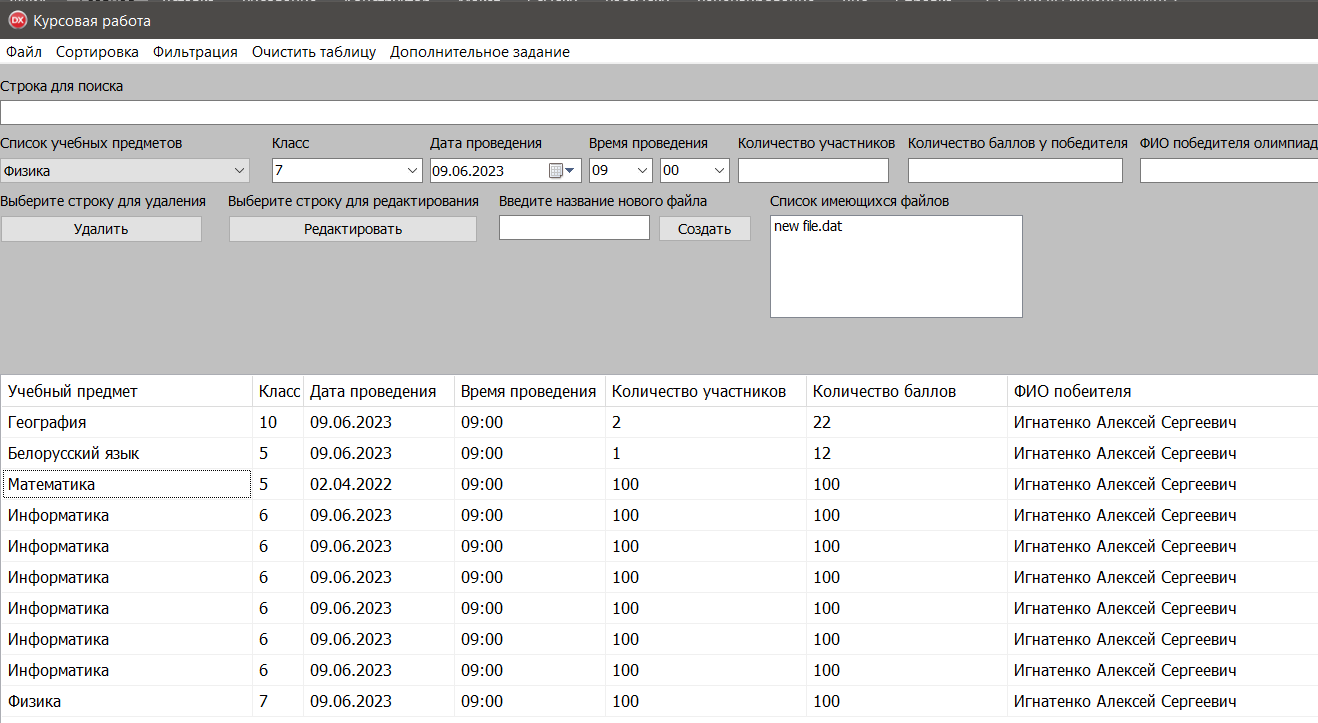


Рисунок 4.6 — Фрагмент таблицы до удаления строки

На рисунке 4.7 показан фрагмент таблицы после удаления строки.

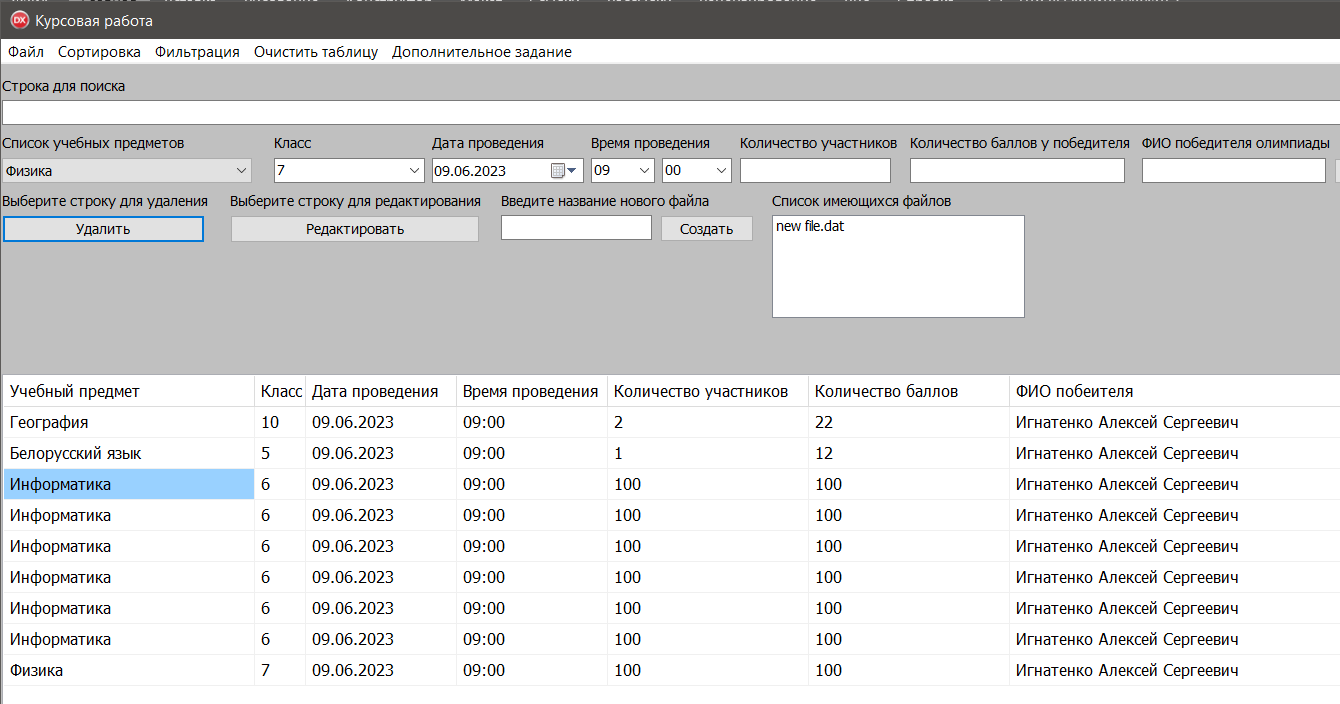


Рисунок 4.7 — Фрагмент таблицы после удаления строки

Для редактирования необходимо выбрать ячейку из строки для редактирования и нажать кнопку «Редактировать», после чего содержимое строки отобразится в полях для ввода информации. Для подтверждения редактирования необходимо нажать на кнопку «Сохранить», для отмены редактирования необходимо нажать на кнопку «Отменить редактирование», после отмены редактирования содержимое выбранной строки не изменится. На рисунке 4.8 показан фрагмент таблицы до редактирования.

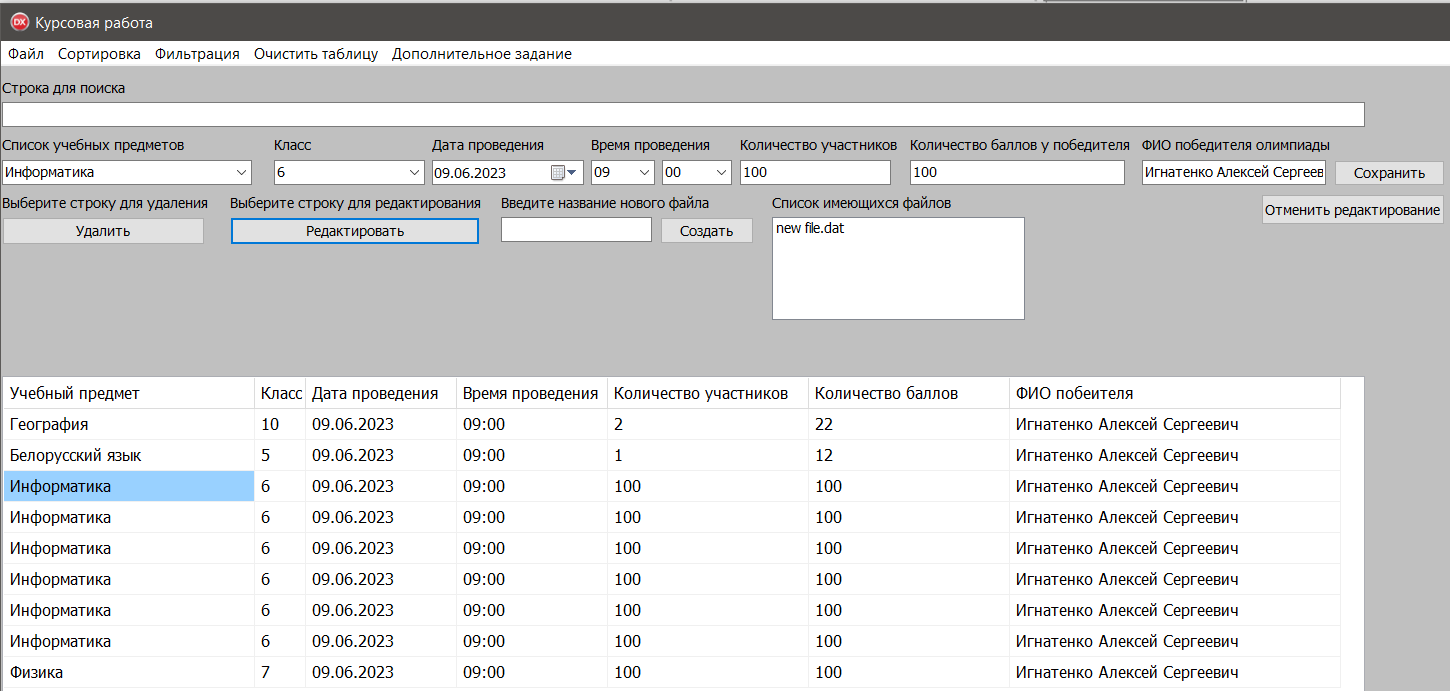


Рисунок 4.8 — Фрагмент таблицы до редактирования

На рисунке 4.9 показан фрагмент таблицы после редактирования.

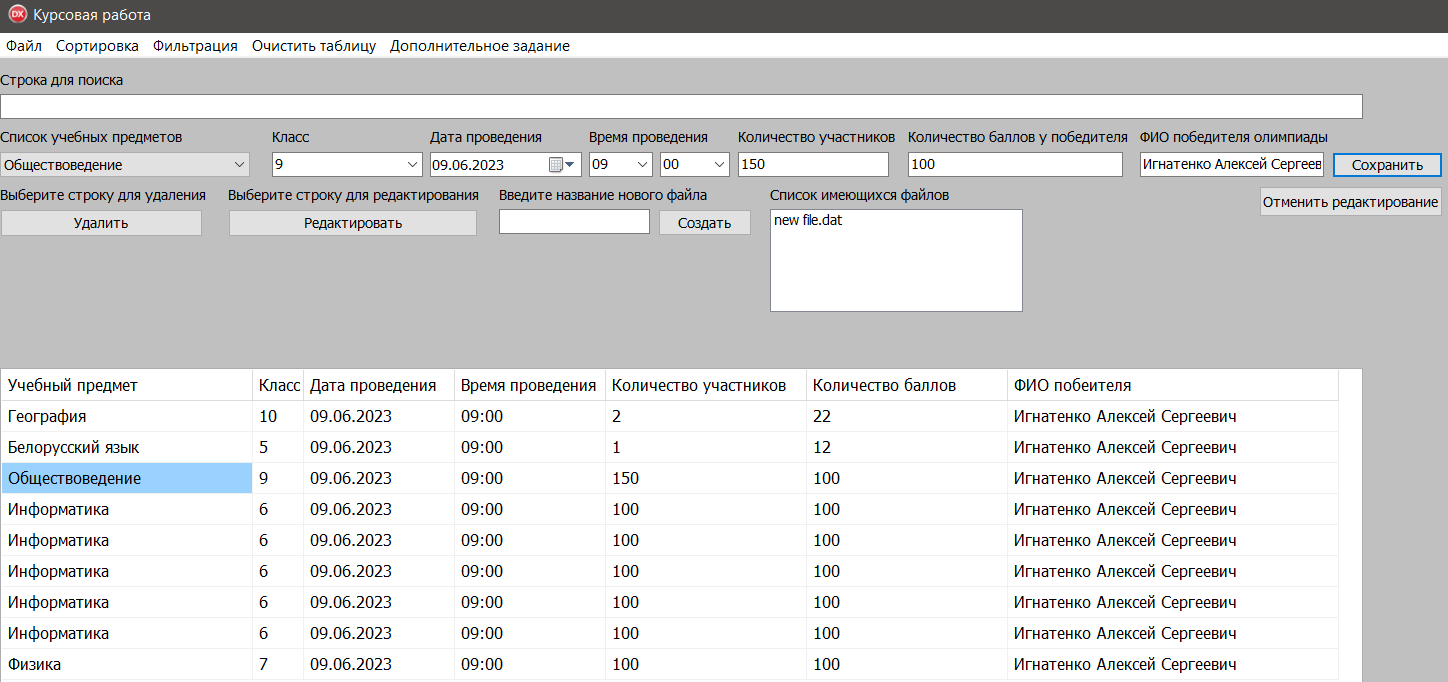


Рисунок 4.9 — Фрагмент таблицы после редактировании

Для создания нового типизированного файла необходимо в поле «Введите название нового файла» ввести название файла и нажать кнопку «Создать». На риске 4.10 показано создание нового типизированного файла.

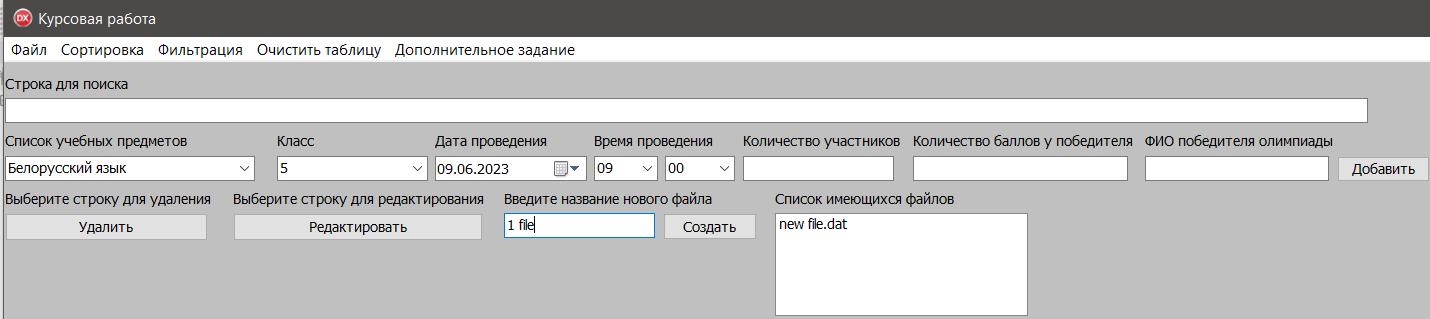


Рисунок 4.10 — Создание типизированного файла

После чего в «Список существующих файлов» отобразится созданный файл. На рисунке 4.11 показано отображение созданного файла в «Список имеющихся файлов».

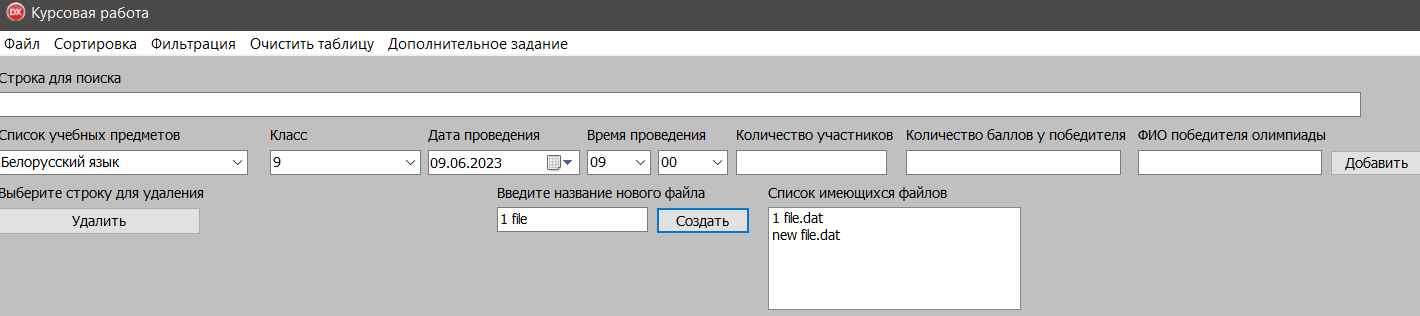


Рисунок 4.11 — Отображение созданного файла

При создании нового файла если дать ему название файла, который уже существует, то отобразится предупреждение о том, файл с таким названием уже существует. При нажатии на кнопку «OK» создастся файл с таким названием, и вся информация из старого файла потеряется, при нажатии на кнопку «Cancel» создание отменится. На рисунке 4.12 показано предупреждение.

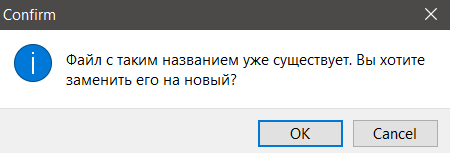


Рисунок 4.12 — Предупреждение о создании файла с названием файла который уже существует

Для открытия файлов созданных через приложение необходимо дважды нажать на имя файла в «Список существующих файлов» и в таблице отобразится информация из файла. На рисунке 4.13 показано содержимое файла «new file.dat».

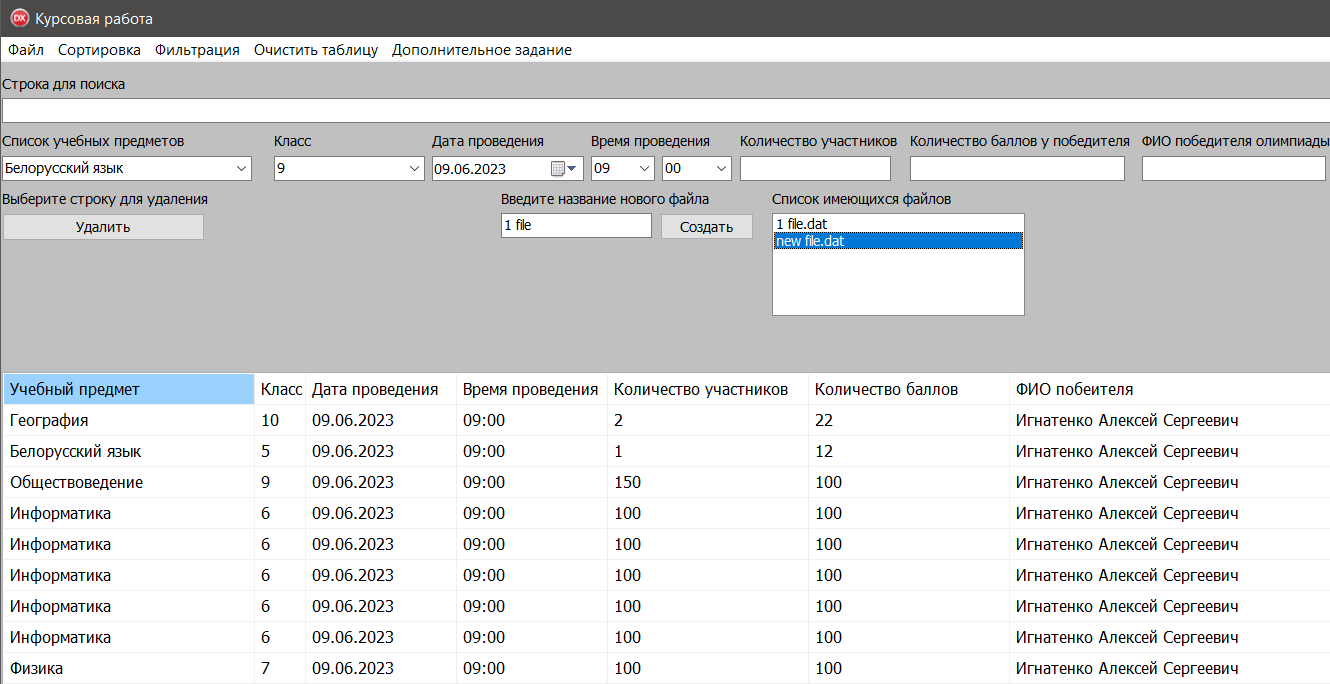


Рисунок 4.13 — Содержимое файла «new file.dat»

На рисунке 4.14 показано содержимое файла «1 file.dat».

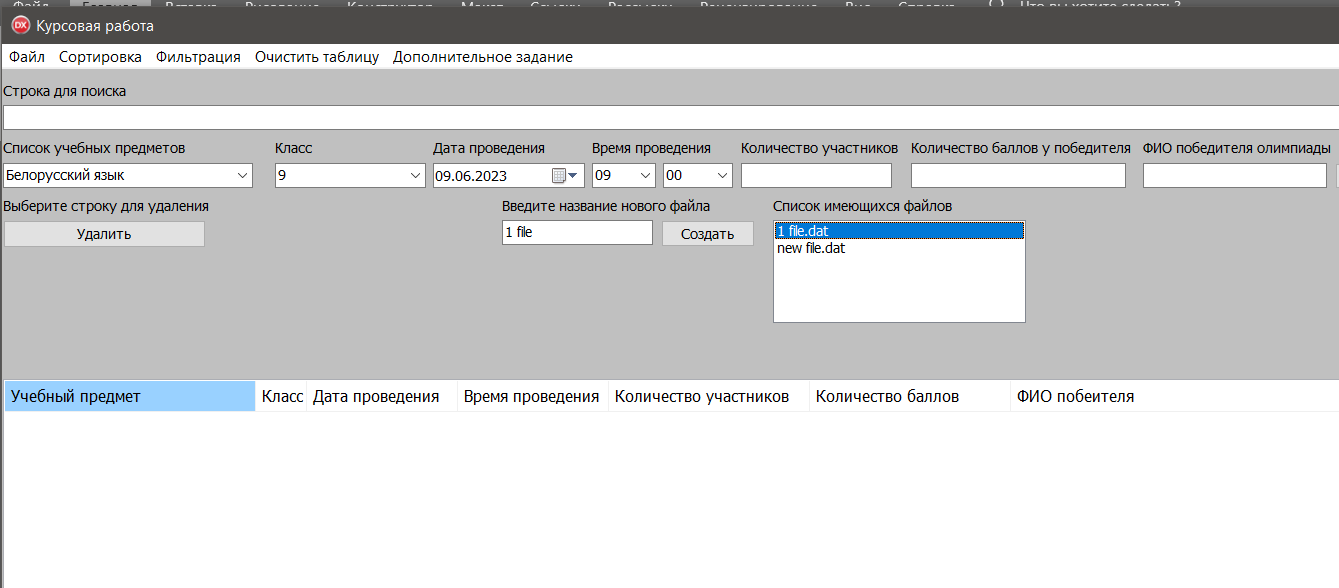


Рисунок 4.14 — Содержимое файла «1 file.dat»

Для открытия файла созданного не через программу необходимо в главном меню в разделе «Файл» нажать на кнопку «Открыть», после чего откроется проводник. В проводнике найти необходимый файл, который нужно открыть и нажать на кнопку «Открыть». На рисунке 4.15 показано содержимое файла, созданного не через программу.

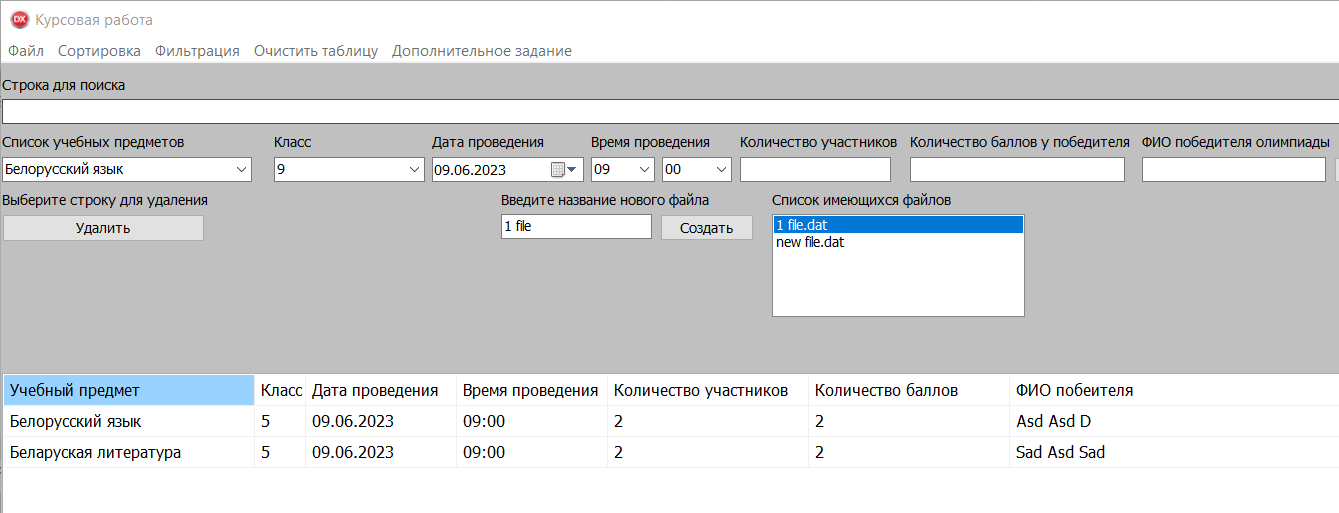


Рисунок 4.15 — Содержимое файла созданного не через программу

Для сохранения изменений в файл необходимо в главном меню перейти в раздел «Файл» и нажать на кнопку «Сохранить».

Для сортировки информации необходимо в главном меню в разделе «Сортировка» выбрать подходящую сортировку. На рисунке 4.16 показаны все виды сортировки.

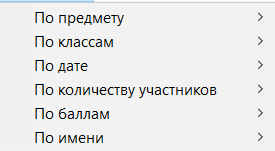


Рисунок 4.16 — Все виды сортировки

На рисунке 4.17 показана фрагмент информация до сортировки по классам по убыванию.

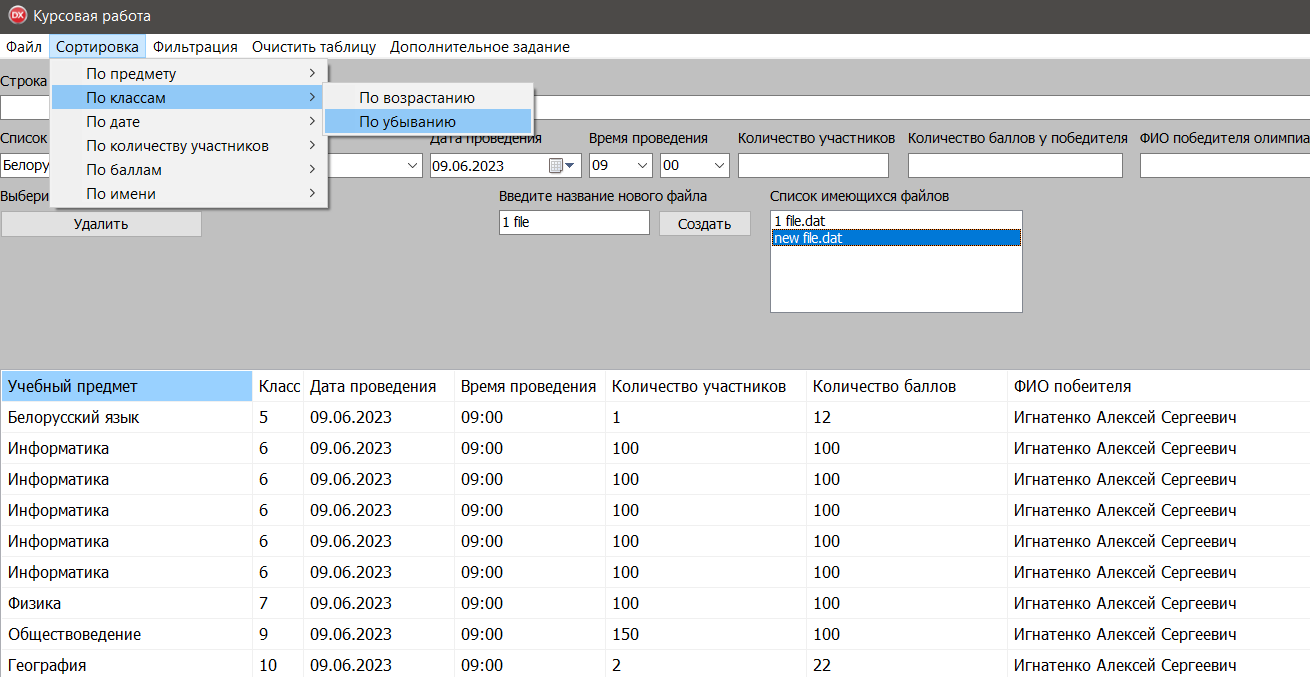


Рисунок 4.17 — Фрагмент информация до сортировки по классам по убыванию

На рисунке 4.18 показан фрагмент информация после сортировки по классам по убыванию.

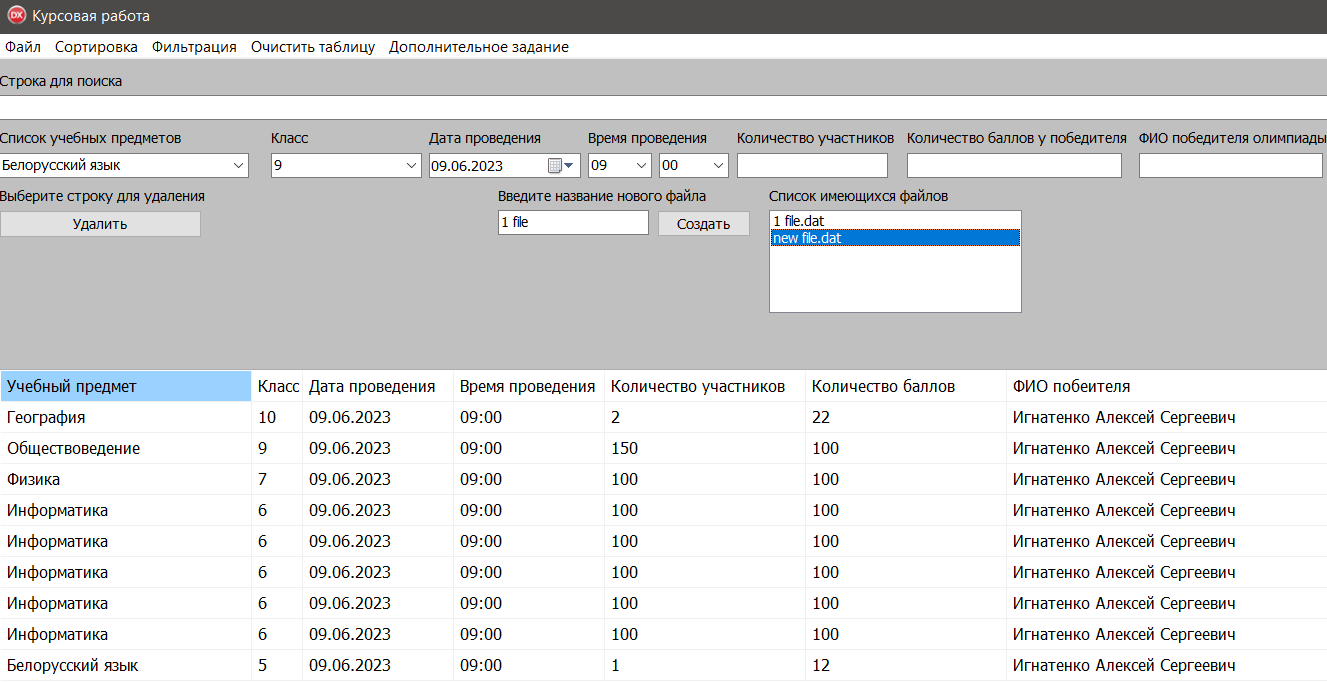


Рисунок 4.18 — Фрагмент информация после сортировки по классам по убыванию

Для отмены сортировки необходимо в главном меню в разделе «Файл» нажать на кнопку «Отменить последние действия». На рисунке 4.19 показан фрагмент информация до отмены сортировки.

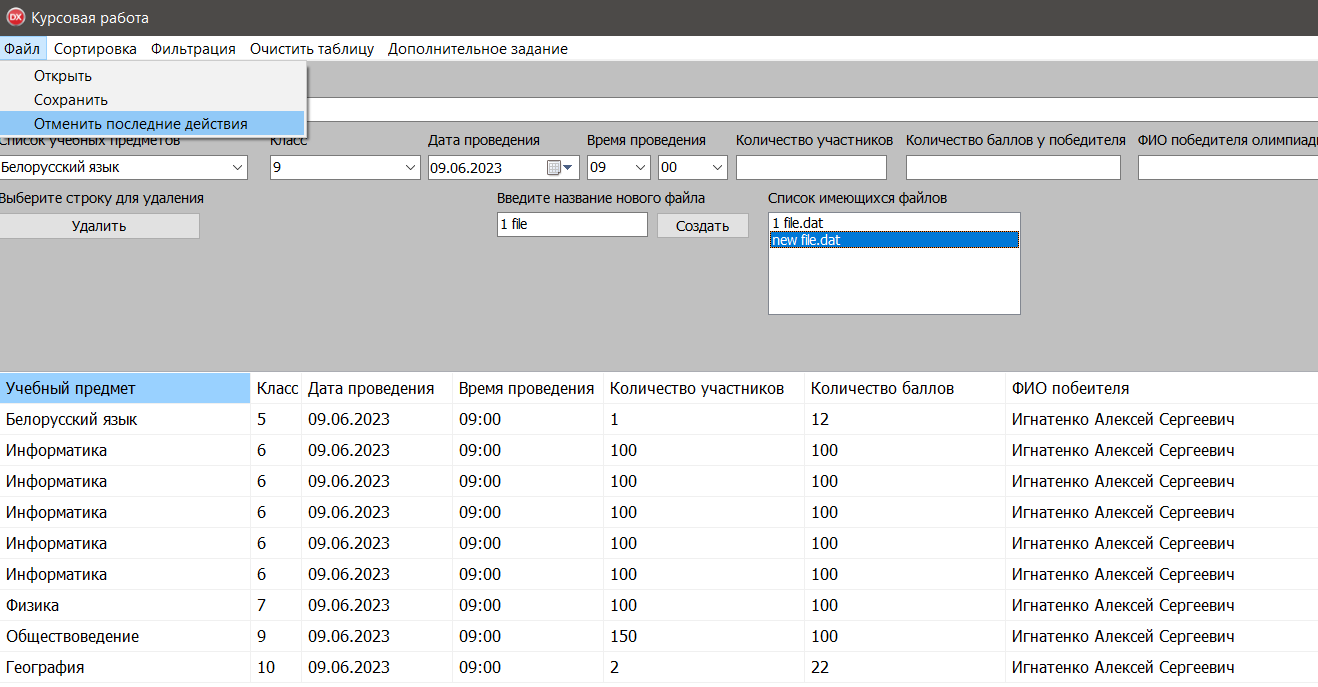


Рисунок 4.19 —Фрагмент информация до отмены сортировки

На рисунке 4.20 показан фрагмент информация после отмены сортировки.

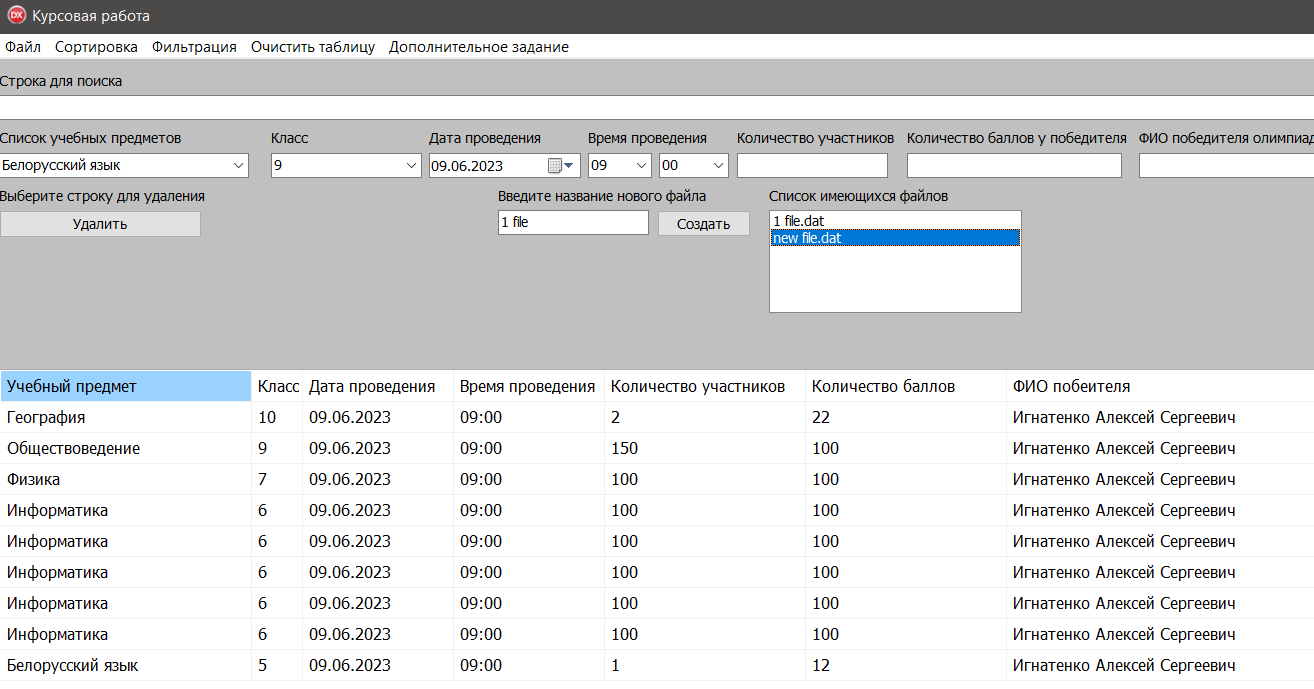


Рисунок 4.20 — Фрагмент информация после отмены сортировки

Для фильтрации информации пользователю необходимо в главном меню в разделе «Фильтрация» выбрать подходящую фильтрацию и выбрать критерий фильтрации. На рисунке 4.21 показан фрагмент информация до фильтрации по классам.

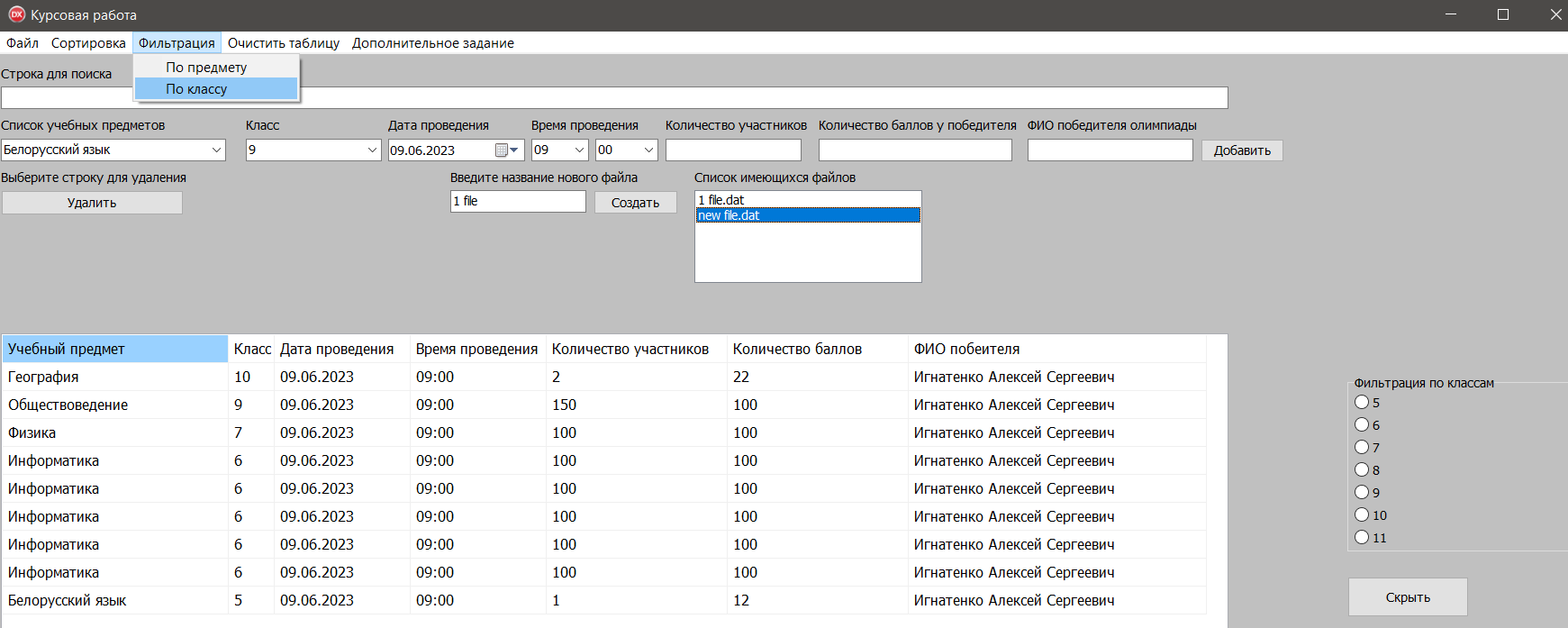


Рисунок 4.21 — Фрагмент информация до фильтрации по классам

На рисунке 4.22 показана фрагмент информация после фильтрации по классам.

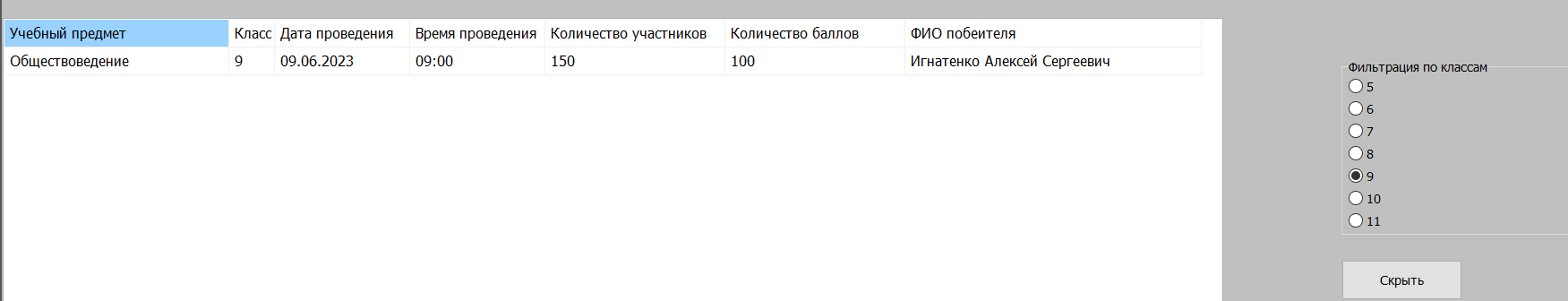


Рисунок 4.22 — Фрагмент информация после фильтрации по классам

Для отмены фильтрации необходимо нажать на кнопку «Скрыть» после чего отобразится информация до фильтрации. На рисунке 4.23 показана фрагмент информация после отмены фильтрации.

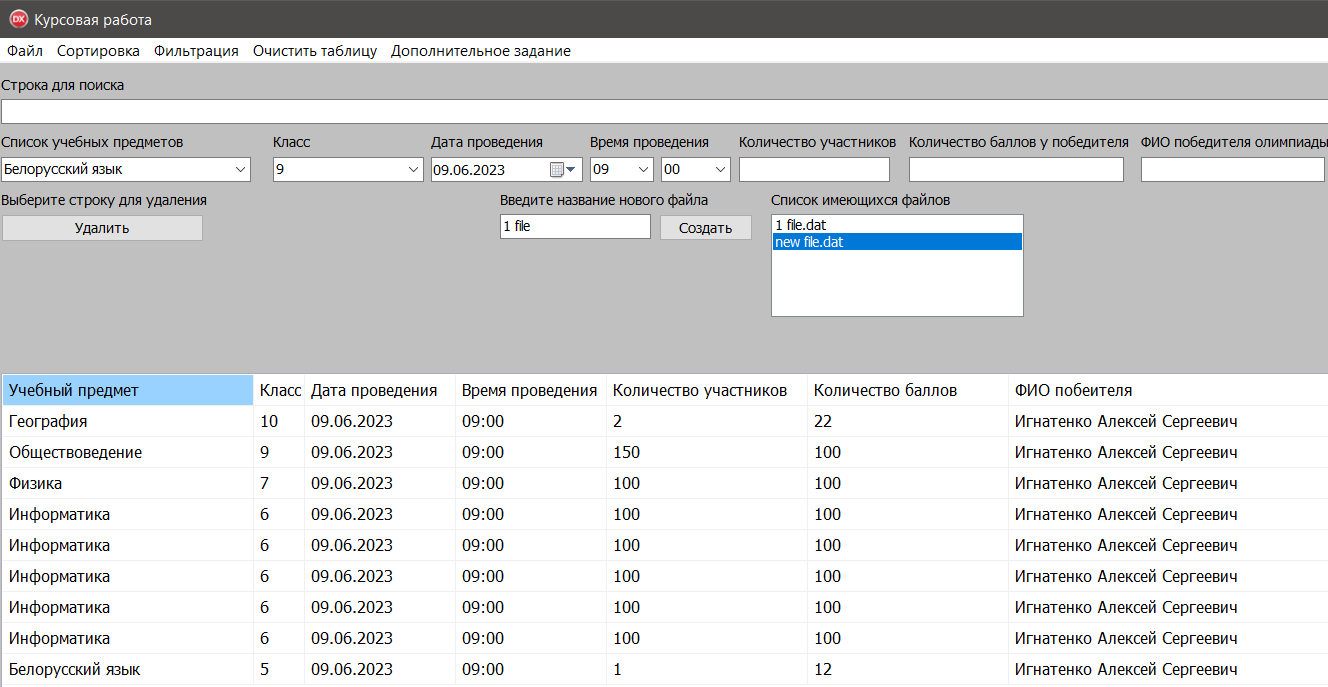


Рисунок 4.23 — Информация после отмены фильтрации

Для очистки таблицы от всей информации необходимо в главном меню выбрать «Очистить таблицу» после чего таблица будет очищена. На рисунке 4.24 показана фрагмент таблицы до очистки.

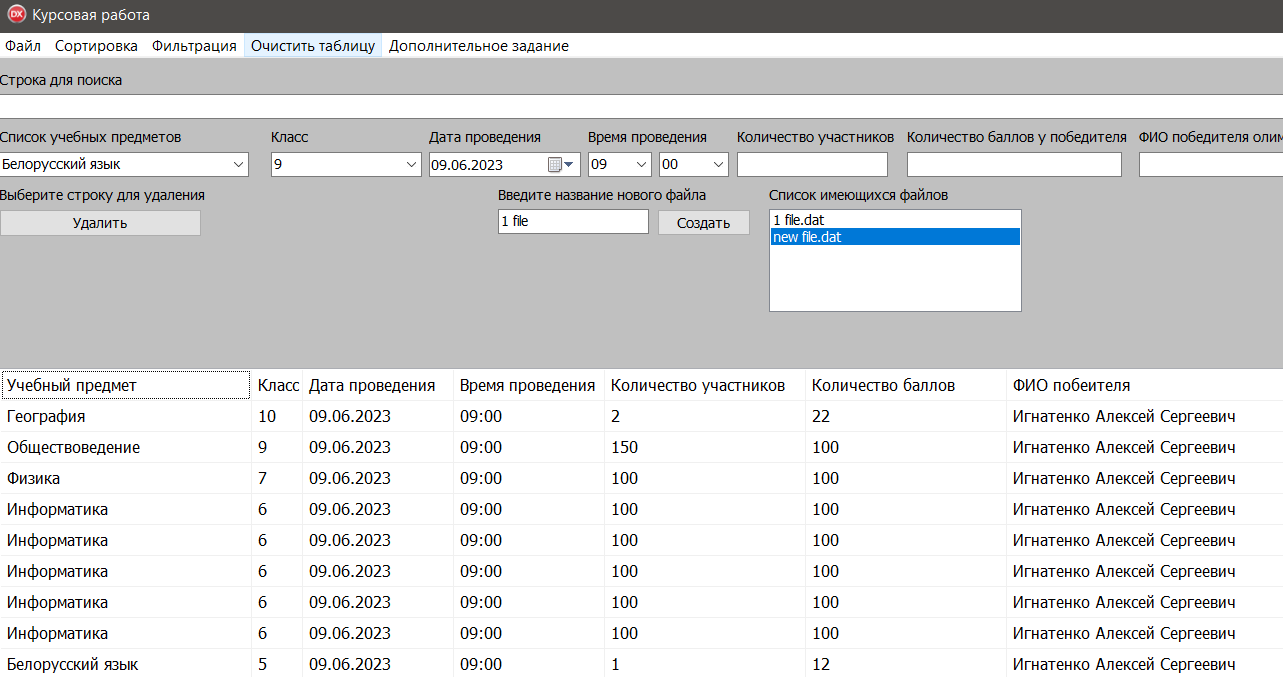


Рисунок 4.24 — Фрагмент таблицы до очистки

На рисунке 4.25 показан фрагмент информации после очистки таблицы от всех данных.

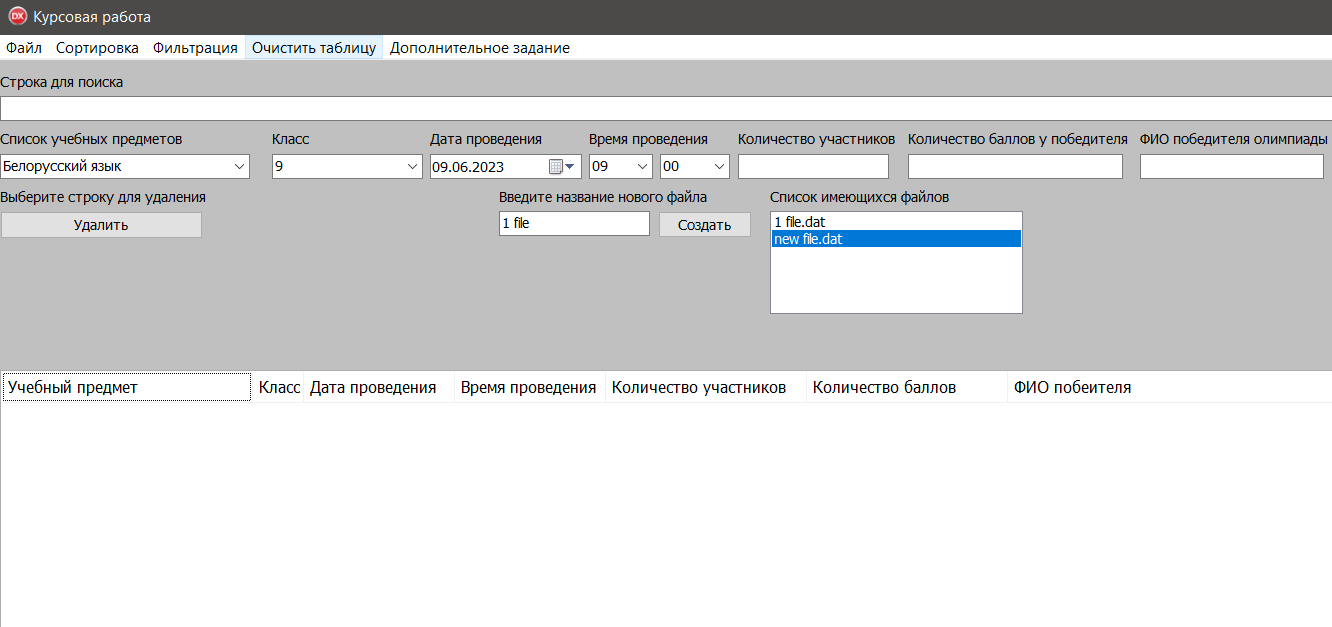


Рисунок 4.25 — Фрагмент таблицы после очистки от всех данных

Для удаления участников ниже введенного бала необходимо в главном меню в разделе «Дополнительное задание» нажать на кнопку «Удаление участников ниже определенного балла». После чего появится поле для ввода баллов и кнопка «Удалить всех участников ниже введенного балла», в поле «Введите бал» необходимо указать бал и нажать на кнопку «Удалить всех участников ниже введенного балла». На рисунке 4.26 показан фрагмент таблицы до удаления участников ниже введенного балла.

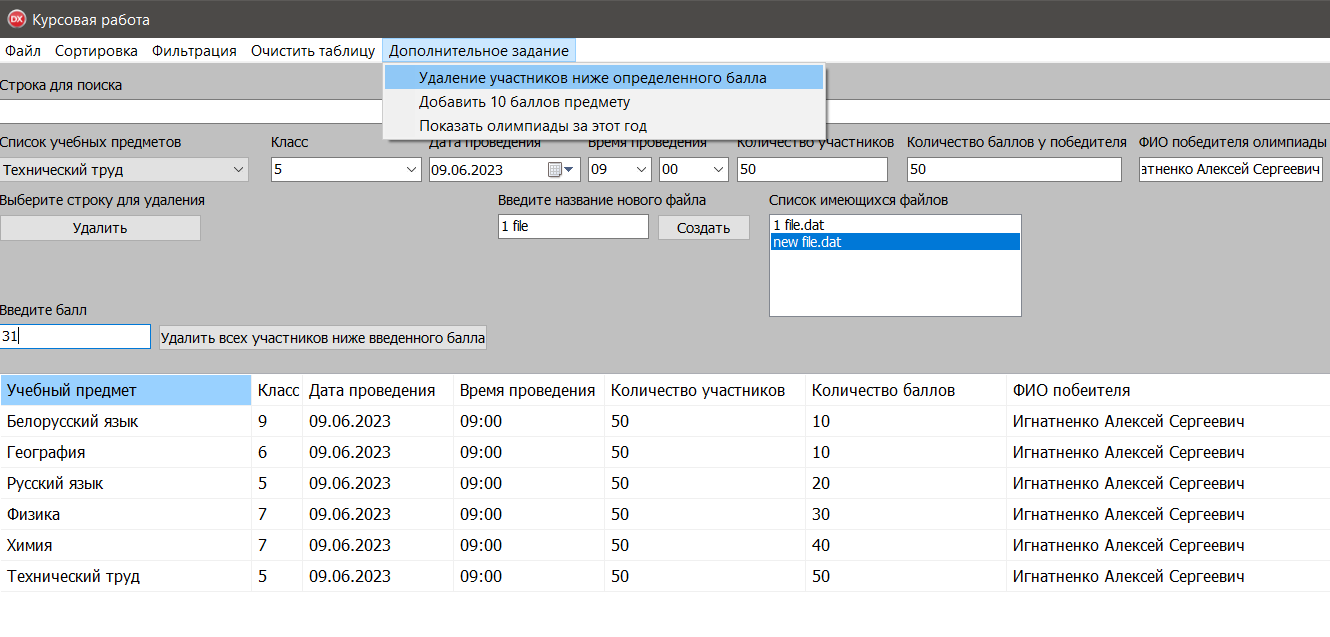


Рисунок 4.26 — Фрагмент таблицы до удаления участников ниже введенного балла

На рисунке 4.27 показан фрагмент таблицы после удаления участников ниже введенного балла.

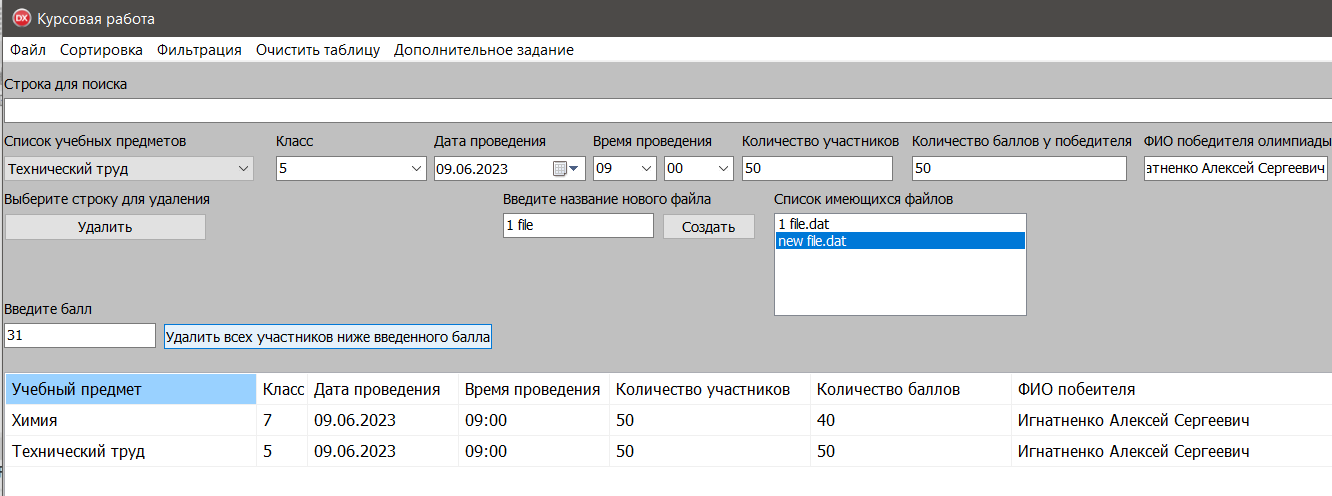


Рисунок 4.27 — Фрагмент таблицы после удаления участников ниже введённого балла

Для добавления 10 баллов определённому предмету необходимо в главном меню в разделе «Дополнительное задание» нажать на кнопку «Добавить 10 баллов предмету», выбрать предмет и нажать на кнопку «Добавить 10 баллов». На рисунке 4.28 показан фрагмент таблицы до добавления 10 баллов предмету.

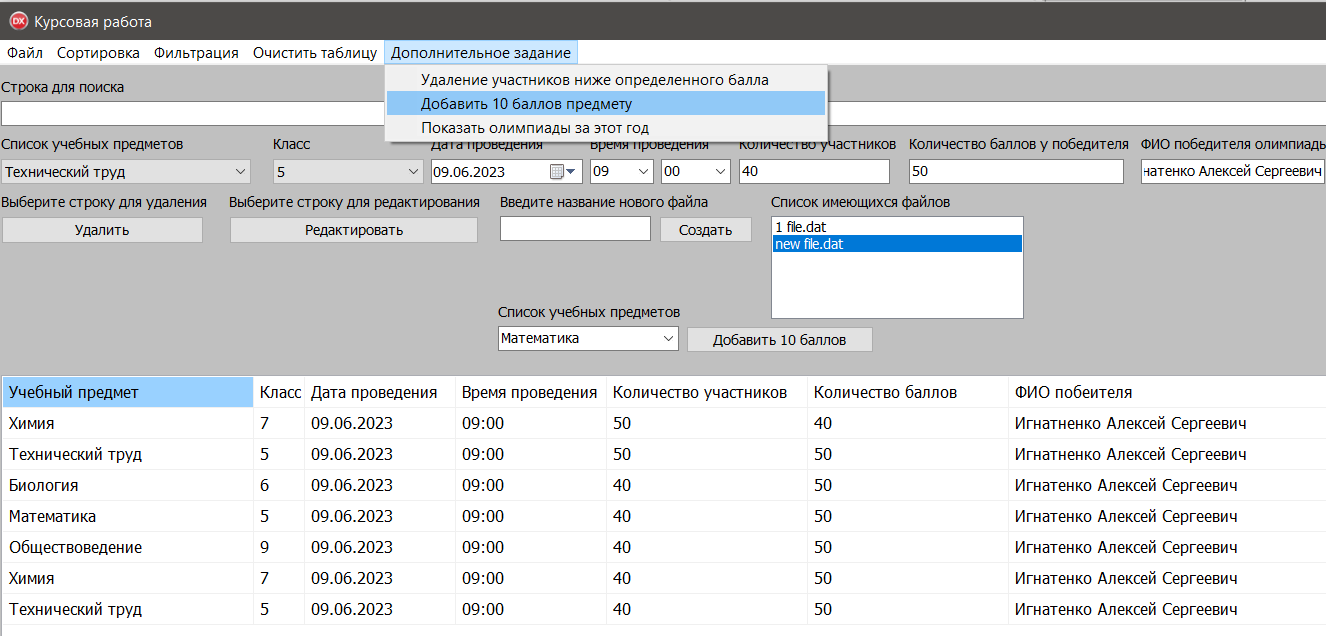


Рисунок 4.28 — Фрагмент таблицы до добавления 10 баллов предмету

На рисунке 4.29 показан фрагмент таблицы после добавления 10 баллов предмету.

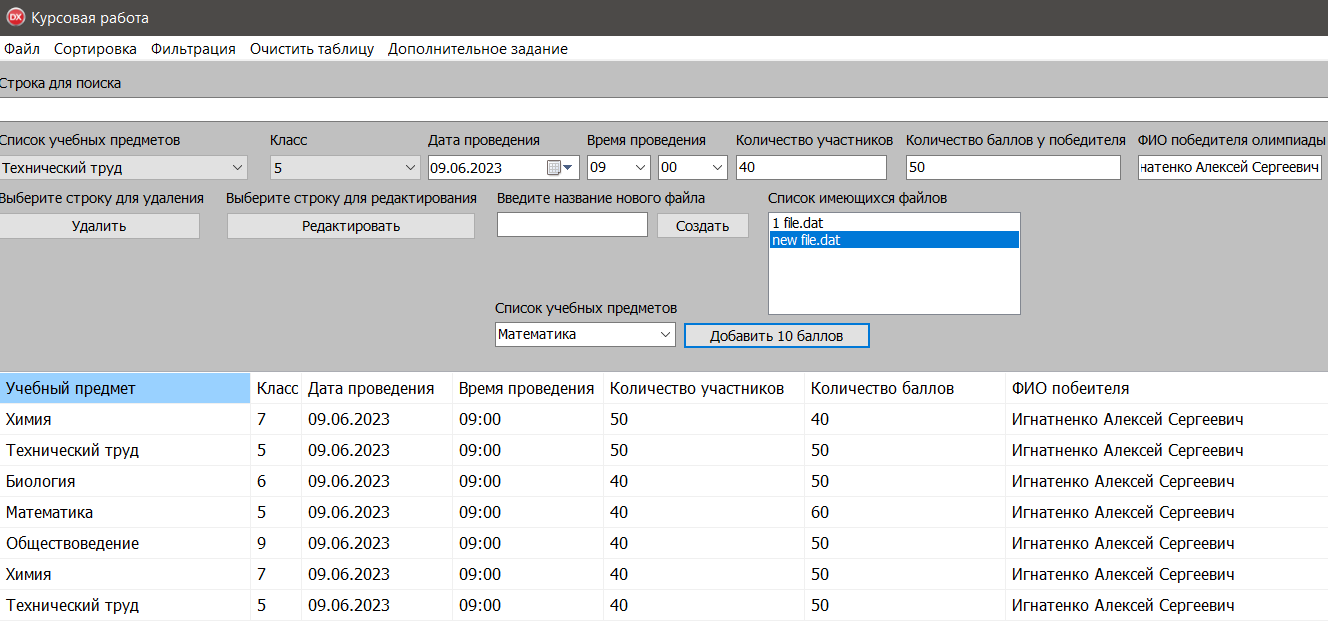


Рисунок 4.29 — Фрагмент таблицы после добавления 10 баллов предмету

Для просмотра информации об олимпиадах, проходивших в текущем году необходимо в главном меню в разделе «Дополнительное задание» нажать на кнопку «Показать олимпиады за этот год» после чего отобразится информация о проводившихся олимпиадах за текущий год. На рисунке 4.30 показан фрагмент таблицы до вывода информации об олимпиадах за текущий год.

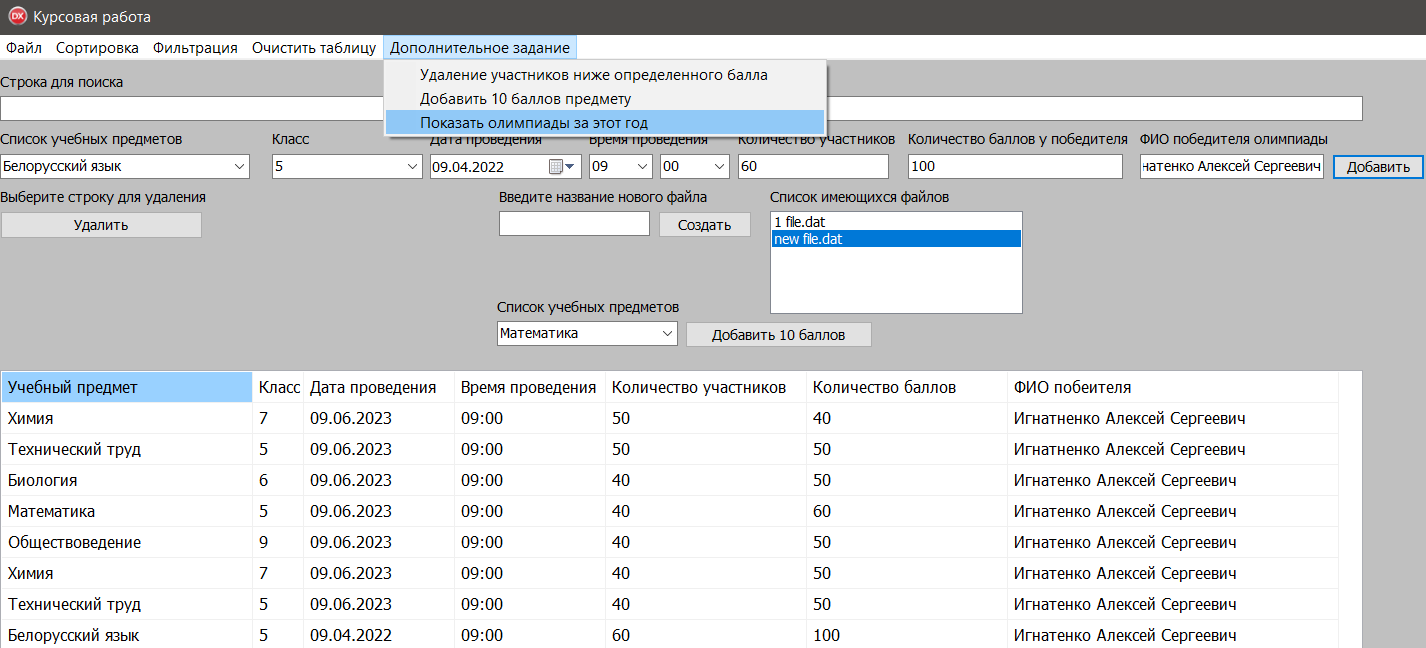


Рисунок 4.30 — Фрагмент таблицы до вывода информации об олимпиадах за текущий год

На рисунке 4.31 показан фрагмент таблицы после вывода информации о олимпиадах проводившихся в текущем году.

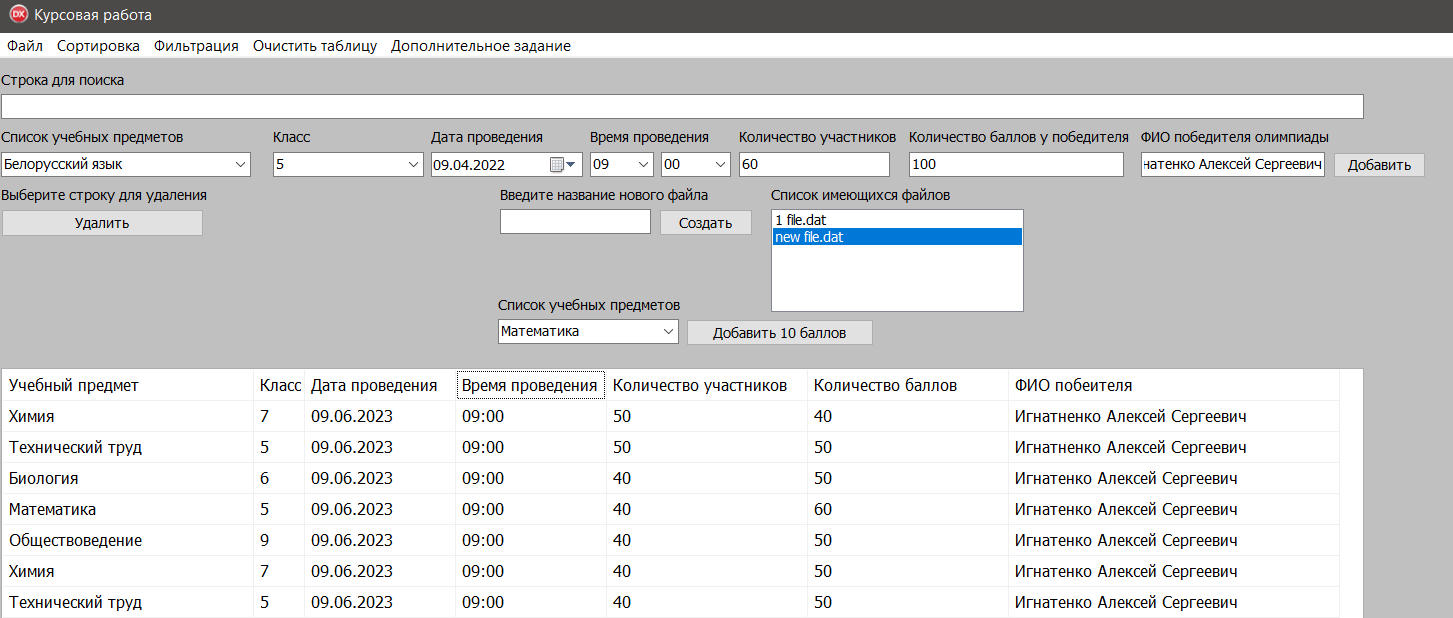


Рисунок 4.31 — Фрагмент таблицы после вывода информации о олимпиадах проводившихся в текущем году

Для поиска информации в таблице необходимо в строку «Строка для поиска» внести значение для поиска после чего результаты поиска отобразятся зеленым цветом. На рисунке 4.32 показан фрагмент таблицы до поиска.

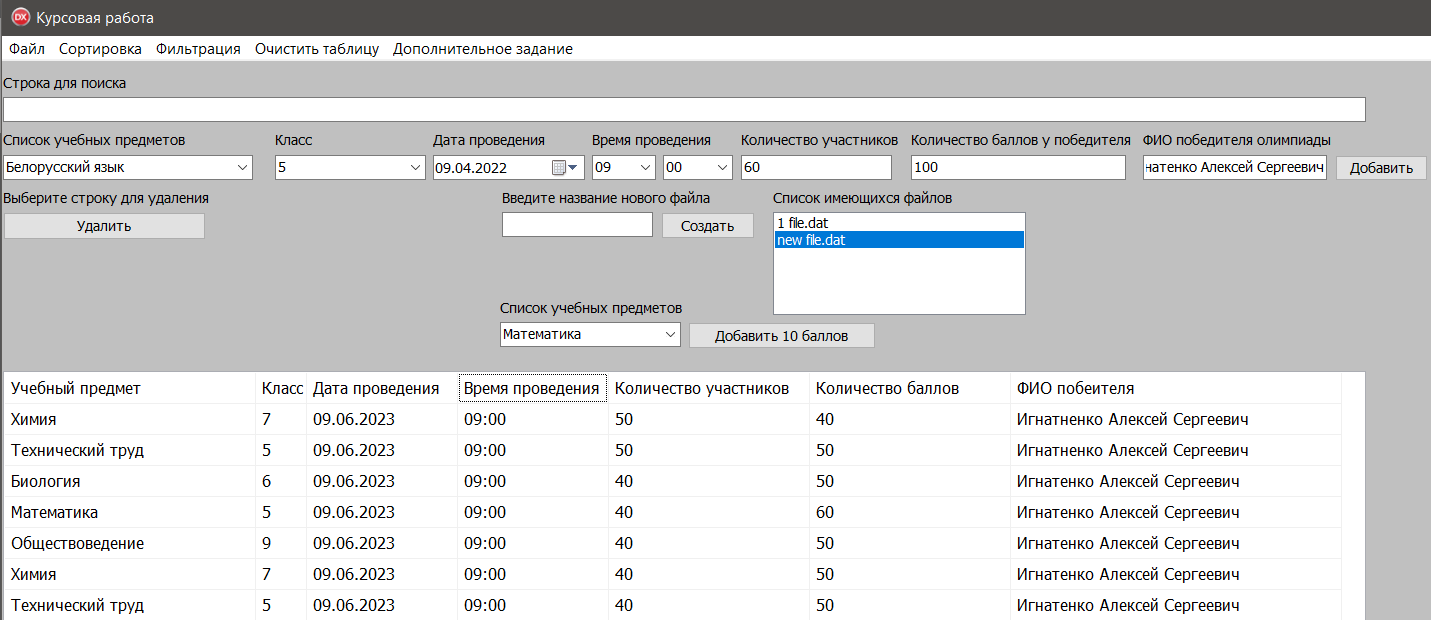


Рисунок 4.32 — Фрагмент таблицы до поиска

На рисунке 4.33 показан фрагмент таблицы после поиска.

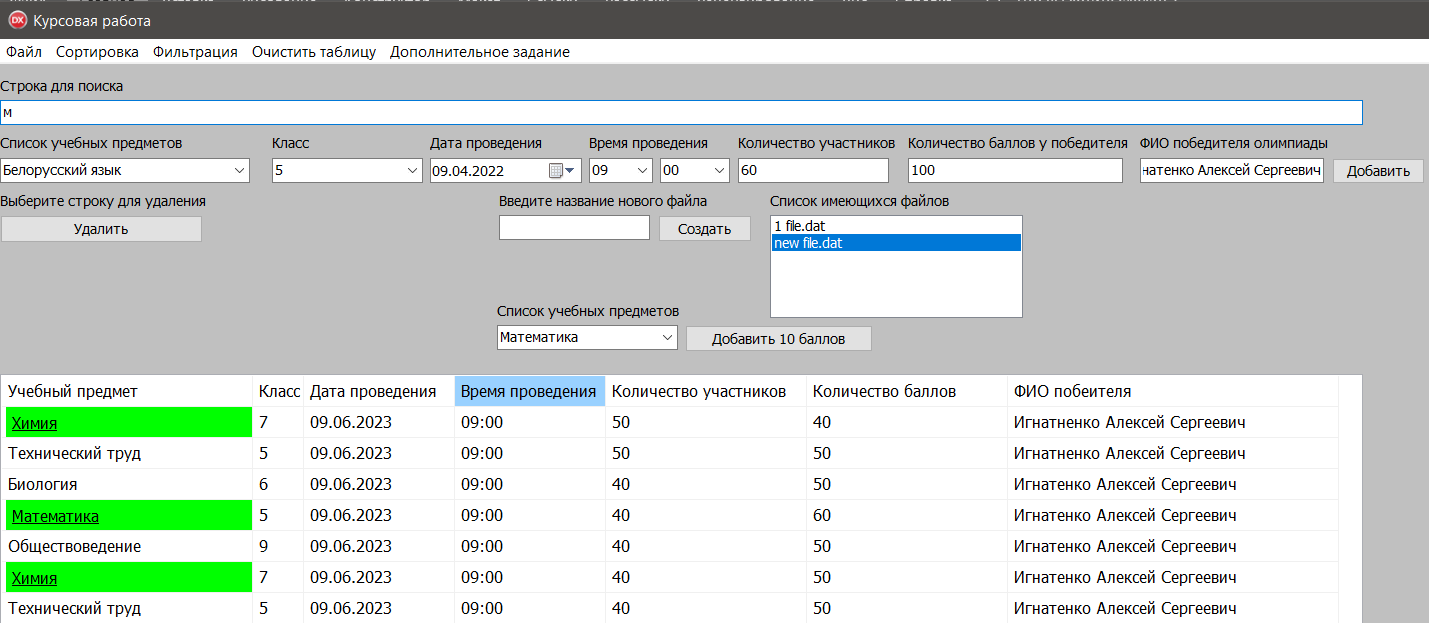


Рисунок 4.33 — Фрагмент таблицы после поиска

Для выхода из приложения необходимо нажать на кнопку «Выход» после чего приложение закроется.

## ЗАКЛЮЧЕНИЕ

В результате выполнения курсовой работы была разработана информационно—поисковая система на тему «Проведение учебных олимпиад», которая позволяет вести учёт олимпиад в виде таблицы.

При написании курсового проекта были изучены теоретические сведения на тему «Процедуры и функции. Формальные и фактические параметры», были закреплены знания о структурных типах данных, была организована работа с типизированными файлами.

В ходе выполнения курсового проекта были решены поставленные задачи, был изучен и описан теоретический вопрос на тему «Процедуры и функции. Формальные и фактические параметры», спроектированы структуры типизированного файла, спроектирован макет интерфейса приложения, определены алгоритмы работы функций, разработан графический пользовательский интерфейс, подключены файлы к приложению, разработан основной функционал программы и индивидуальные функции, организована проверка корректности вводимых данных, проведены тесты и отладки и написана пояснительная записка.

В программе был создан простой и понятный пользователю интерфейс информационно—поисковой системы.

Функциональные возможности программы позволяют добавлять записи в таблицу, выполнять сортировку записей по возрастанию и убыванию, фильтрацию, удаление выбранной записи в таблице, поиск по таблице и редактирование.

По ходу разработки программы были использованы компоненты:

* «StringGrid» — для вывода информации в виде таблицы;
* «Edit» — строка ввода;
* «Button» — функциональная кнопка;
* «ComboBox» — для выбора предложенных вариантов;
* «Image» — для дизайна формы;
* «Lable» — для пояснений;
* «MainMenu» — для верхнего меню программы;
* «DataTimePicker» — для выбора даты;
* «FileListBox» — для отображения имеющихся файлов.

## Список использованных источников

1. Информационно—поисковая система [Электронный ресурс]. —Электронные данные. — Режим доступа: https://science.fandom.com/ru/wiki/Информационно\_поисковые\_системы.
2. Возможности Rad Studio [Электронный ресурс]. —Электронные данные.—Режим доступа: https://www.embarcadero.com/ru/products/rad—studio/features/extend.
3. Задачи информационно—поисковых систем [Электронный ресурс]. —Электронные данные.—Режим доступа: http://asknet.ru/Technology/searchtask.htm.
4. Сухарев М.В. Основы Delphi. Профессиональный подход. — СПб.: Наука и Техника, 2004. — 600 с.: ил.
5. Комлев Н.Ю. Самоучитель игры на Паскаль.—СПб.: СОЛОН—ПРЕСС, 2021. — 252.:ил

## ПРИЛОЖЕНИЕ А

**Листинг информационно — поисковой системы на тему «Проведение учебных олимпиад»**

Текст модуля unit1.pas

unit Unit1;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.Menus, Vcl.Grids,

Vcl.ComCtrls,ShellAPI, Vcl.ExtCtrls, Vcl.Buttons, Vcl.FileCtrl;

type

olimpiada=record

predmet:String[30];

klass:String[10];

data:String[10];

time:String[10];

kolichestvo\_ushenikov:String[5];

kolichestvo\_ballov:String[5];

pobeditel:String[100];

end;

TForm1 = class(TForm)

StringGrid1: TStringGrid;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Label5: TLabel;

Label6: TLabel;

Label7: TLabel;

ComboBox1: TComboBox;

ComboBox2: TComboBox;

DateTimePicker1: TDateTimePicker;

Edit2: TEdit;

Edit3: TEdit;

Edit5: TEdit;

Label8: TLabel;

Button1: TButton;

Button2: TButton;

Button4: TButton;

ComboBox3: TComboBox;

ComboBox4: TComboBox;

Edit1: TEdit;

MainMenu1: TMainMenu;

ComboBox5: TComboBox;

Edit7: TEdit;

OpenDialog1: TOpenDialog;

N3: TMenuItem;

N4: TMenuItem;

N5: TMenuItem;

N6: TMenuItem;

Label9: TLabel;

N7: TMenuItem;

N8: TMenuItem;

Button6: TButton;

N9: TMenuItem;

N10: TMenuItem;

N11: TMenuItem;

N101: TMenuItem;

N12: TMenuItem;

Label10: TLabel;

SaveDialog1: TSaveDialog;

RadioGroup1: TRadioGroup;

N14: TMenuItem;

N15: TMenuItem;

Button7: TButton;

Label11: TLabel;

Button8: TButton;

N17: TMenuItem;

N18: TMenuItem;

BitBtn1: TBitBtn;

RadioGroup2: TRadioGroup;

N19: TMenuItem;

Button3: TButton;

N21: TMenuItem;

N22: TMenuItem;

N23: TMenuItem;

N24: TMenuItem;

N25: TMenuItem;

N26: TMenuItem;

N20: TMenuItem;

N27: TMenuItem;

N28: TMenuItem;

N29: TMenuItem;

N30: TMenuItem;

N31: TMenuItem;

N32: TMenuItem;

N33: TMenuItem;

N34: TMenuItem;

Button9: TButton;

Label12: TLabel;

FileListBox1: TFileListBox;

Edit4: TEdit;

Button10: TButton;

Label14: TLabel;

Label15: TLabel;

procedure FormCreate(Sender: TObject);

procedure Button4Click(Sender: TObject);

procedure Button1Click(Sender: TObject);

procedure Edit1KeyPress(Sender: TObject; var Key: Char);

procedure N1Click(Sender: TObject);

procedure N2Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure N4Click(Sender: TObject);

procedure N5Click(Sender: TObject);

procedure ComboBox1Change(Sender: TObject);

procedure ComboBox2Change(Sender: TObject);

procedure N8Click(Sender: TObject);

procedure Button6Click(Sender: TObject);

procedure N9Click(Sender: TObject);

procedure N12Click(Sender: TObject);

procedure ComboBox5Change(Sender: TObject);

procedure N11Click(Sender: TObject);

procedure N101Click(Sender: TObject);

procedure Edit4KeyPress(Sender: TObject; var Key: Char);

procedure NewFIle1Click(Sender: TObject);

procedure N13Click(Sender: TObject);

procedure RadioGroup1Click(Sender: TObject);

procedure N14Click(Sender: TObject);

procedure N15Click(Sender: TObject);

procedure ComboBox3Change(Sender: TObject);

procedure ComboBox4Change(Sender: TObject);

procedure Button7Click(Sender: TObject);

procedure N17Click(Sender: TObject);

procedure Button8Click(Sender: TObject);

procedure N18Click(Sender: TObject);

procedure BitBtn1Click(Sender: TObject);

procedure RadioGroup2Click(Sender: TObject);

procedure N19Click(Sender: TObject);

procedure Button3Click(Sender: TObject);

procedure N20Click(Sender: TObject);

procedure N21Click(Sender: TObject);

procedure N22Click(Sender: TObject);

procedure N23Click(Sender: TObject);

procedure N26Click(Sender: TObject);

procedure N27Click(Sender: TObject);

procedure N28Click(Sender: TObject);

procedure N30Click(Sender: TObject);

procedure N31Click(Sender: TObject);

procedure N34Click(Sender: TObject);

procedure N33Click(Sender: TObject);

procedure Edit7Change(Sender: TObject);

procedure Button9Click(Sender: TObject);

procedure Button10Click(Sender: TObject);

procedure StringGrid1DrawCell(Sender: TObject; ACol, ARow: Integer;

Rect: TRect; State: TGridDrawState);

procedure FileListBox1Change(Sender: TObject);

procedure ListBox1Click(Sender: TObject);

procedure FileListBox1DblClick(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

i,n,j,r,m:integer;

a: Array[1..100] of olimpiada;

f:file of olimpiada;

s,link\_save,link\_open:string;

implementation

{$R \*.dfm}

procedure TForm1.BitBtn1Click(Sender: TObject);

begin

form1.Close;

end;

procedure TForm1.Button10Click(Sender: TObject);

var k,btn:integer;

begin

if (edit4.Text='') then

messageDlg('Некоректное название файла', mtError, [mbOK],0)

else

begin

link\_open := Edit4.Text+'.dat';

for i:=0 to filelistbox1.Count—1 do

if(link\_open=filelistbox1.Items[i]) then

btn:=messageDlg('Файл с таким названием уже существует. Вы хотите заменить его на новый?',mtConfirmation, mbOKCancel, 0);

if(btn=mrOK) then

begin

AssignFile(F, link\_open);

ReWrite(F);

CloseFile(F);

filelistbox1.Update;

end

else

begin

AssignFile(F, link\_open);

ReWrite(F);

CloseFile(F);

filelistbox1.Update;

end;

end;

end;

procedure TForm1.Button1Click(Sender: TObject);

var

n,i,k:integer;

begin

n:=StrToInt(Edit5.Text);

if ((n>—1) and (n<101)) then

begin

for i:=StringGrid1.RowCount—1 downto 1 do

begin

if StrToInt(StringGrid1.Cells[5,i])<n then

begin

for j:=i to StringGrid1.RowCount—1 do

begin

StringGrid1.Cells[0,j]:=StringGrid1.Cells[0,j+1];

StringGrid1.Cells[1,j]:=StringGrid1.Cells[1,j+1];

StringGrid1.Cells[2,j]:=StringGrid1.Cells[2,j+1];

StringGrid1.Cells[3,j]:=StringGrid1.Cells[3,j+1];

StringGrid1.Cells[4,j]:=StringGrid1.Cells[4,j+1];

StringGrid1.Cells[5,j]:=StringGrid1.Cells[5,j+1];

StringGrid1.Cells[6,j]:=StringGrid1.Cells[6,j+1];

end;

StringGrid1.RowCount:=StringGrid1.RowCount—1;

end;

end;

end

else messageDlg('Некоректное значение', mtError, [mbOK],0);

end;

procedure TForm1.Button2Click(Sender: TObject);

var n: integer;

begin

for i:=1 to StringGrid1.RowCount—1 do

if (StrToInt(StringGrid1.Cells[5,i])<101) then

begin

if StringGrid1.Cells[0,i]= combobox5.Text then

begin

n:=StrToInt(StringGrid1.Cells[5,i]);

n:=n+10;

StringGrid1.Cells[5,i]:=IntToStr(n);

if (StrToInt(StringGrid1.Cells[5,i])>101) then

StringGrid1.Cells[5,i]:='100';

end;

end

else messageDlg('Максимум 100 баллов', mtError, [mbOK],0);

end;

procedure TForm1.Button3Click(Sender: TObject);

begin

RadioGroup1.Visible:=false;

RadioGroup2.Visible:=false;

Button3.Visible:=false;

RadioGroup1.ItemIndex:=—1;

RadioGroup2.ItemIndex:=—1;

AssignFile(f,link\_open);

Reset(f);

n:=filesize(f);

StringGrid1.RowCount:=n+1;

for i:=1 to StringGrid1.RowCount—1 do

begin

StringGrid1.RowCount:=n+1;

read(f,a[i]);

StringGrid1.Cells[0,i]:=a[i].predmet;

StringGrid1.Cells[1,i]:=a[i].klass;

StringGrid1.Cells[2,i]:=a[i].data;

StringGrid1.Cells[3,i]:=a[i].time;

StringGrid1.Cells[4,i]:=a[i].kolichestvo\_ushenikov;

StringGrid1.Cells[5,i]:=a[i].kolichestvo\_ballov;

StringGrid1.Cells[6,i]:=a[i].pobeditel;

end;

closefile(f);

stringgrid1.FixedRows:=1;

end;

procedure TForm1.Button4Click(Sender: TObject);

var i,k,q: integer;

fio, inputStr,cheslo:string;

begin

if ((edit2.Text<>'')and(edit3.Text<>'')) then

begin

cheslo:=edit2.Text;

q:=cheslo.Length;

for i := 1 to q do

begin

if(cheslo[1]='0')then

begin

delete(cheslo,1,1);

end;

end;

edit2.Text:=cheslo;

cheslo:=edit3.Text;

q:=cheslo.Length;

for i := 1 to q do

begin

if(cheslo[1]='0')then

begin

delete(cheslo,1,1);

end;

end;

edit3.Text:=cheslo;

end;

k:=0;

//DateTimePicker1.MaxDate := Date();

DateTimePicker1.MinDate := EncodeDate(2000, 1, 1);

inputStr:=Edit1.Text ;

fio := '';

for i := 1 to Length(inputStr) do

begin

if (i = 1) or(inputStr[i—1] = ' ') then

begin

fio := fio + AnsiUpperCase(inputStr[i]);

k:=k+1;

end

else

begin

fio := fio + AnsiLowercase(inputStr[i]);

end;

end;

if(Combobox1.Text='') then

messageDlg('Выберите предмет', mtError, [mbOK],0);

if(Combobox2.Text='') then

messageDlg('Выберите класс', mtError, [mbOK],0);

if((Combobox3.Text='')or(Combobox4.Text='')) then

messageDlg('Выберите время', mtError, [mbOK],0);

if (edit2.Text='') then

messageDlg('Невероне количество участников', mtError, [mbOK],0);

if (edit3.Text='') then

messageDlg('Невероне количество баллов (минимальный бал — 0, максимальный — 100)', mtError, [mbOK],0);

if ((k<>3)and(k<>4)) then

messageDlg('Невероне ФИО', mtError, [mbOK],0);

if ((edit2.Text<>'')and(edit3.Text<>'')) then

begin

if ((Combobox1.Text<>'')and(Combobox2.Text<>'')and(Combobox3.Text<>'')and (Combobox4.Text<>'') and((StrToint(edit2.text)>0)and(StrToint(edit2.text)<251)) and ((StrToint(edit3.text)>—1)and (StrToint(edit3.text)<101))and((k=3)or(k=4))) then

begin

StringGrid1.RowCount:=StringGrid1.RowCount+1;

StringGrid1.cells[0,StringGrid1.RowCount—1]:=combobox1.Text;

StringGrid1.cells[1,StringGrid1.RowCount—1]:=combobox2.Text;

StringGrid1.cells[2,StringGrid1.RowCount—1]:=FormatDateTime('',DateTimePicker1.Date);

StringGrid1.cells[3,StringGrid1.RowCount—1]:=combobox3.Text+':'+combobox4.Text;

StringGrid1.cells[4,StringGrid1.RowCount—1]:=edit2.Text;

StringGrid1.cells[5,StringGrid1.RowCount—1]:=edit3.Text;

StringGrid1.cells[6,StringGrid1.RowCount—1]:=fio;

stringgrid1.FixedRows:=1;

end;

end;

edit1.Clear;

edit2.Clear;

edit3.Clear;

AssignFile(f,link\_open);

Rewrite(f);

for I:=1 to StringGrid1.RowCount—1 do

begin

a[i].predmet:=StringGrid1.Cells[0,i];

a[i].klass:=StringGrid1.Cells[1,i];

a[i].data:=StringGrid1.Cells[2,i];

a[i].time:=StringGrid1.Cells[3,i];

a[i].kolichestvo\_ushenikov:=StringGrid1.Cells[4,i];

a[i].kolichestvo\_ballov:=StringGrid1.Cells[5,i];

a[i].pobeditel:=StringGrid1.Cells[6,i];

write(f,a[i]);

end;

closefile(f);

end;

procedure TForm1.Button6Click(Sender: TObject);

var btn:integer;

str:string;

begin

n:=Stringgrid1.Row;

str:=InttoStr(n);

btn:=messageDlg('Вы действительно хотите удалить строку: '+str,mtConfirmation, [mbYes,mbNo], 0);

if(btn=mrYes) then

begin

for i:=n to StringGrid1.RowCount—1 do

begin

StringGrid1.Cells[0,i]:=StringGrid1.Cells[0,i+1];

StringGrid1.Cells[1,i]:=StringGrid1.Cells[1,i+1];

StringGrid1.Cells[2,i]:=StringGrid1.Cells[2,i+1];

StringGrid1.Cells[3,i]:=StringGrid1.Cells[3,i+1];

StringGrid1.Cells[4,i]:=StringGrid1.Cells[4,i+1];

StringGrid1.Cells[5,i]:=StringGrid1.Cells[5,i+1];

StringGrid1.Cells[6,i]:=StringGrid1.Cells[6,i+1];

end;

StringGrid1.RowCount:=StringGrid1.RowCount—1;

AssignFile(f,link\_open);

Rewrite(f);

for I:=1 to StringGrid1.RowCount—1 do

begin

a[i].predmet:=StringGrid1.Cells[0,i];

a[i].klass:=StringGrid1.Cells[1,i];

a[i].data:=StringGrid1.Cells[2,i];

a[i].time:=StringGrid1.Cells[3,i];

a[i].kolichestvo\_ushenikov:=StringGrid1.Cells[4,i];

a[i].kolichestvo\_ballov:=StringGrid1.Cells[5,i];

a[i].pobeditel:=StringGrid1.Cells[6,i];

write(f,a[i]);

end;

closefile(f);

end

else

end;

procedure TForm1.Button7Click(Sender: TObject);

var n:integer;

time:string;

begin

button8.Visible:=true;

button9.Visible:=true;

n:=Stringgrid1.Row;

ComboBox1.Style:=csDropDown;

ComboBox2.Style:=csDropDown;

ComboBox3.Style:=csDropDown;

ComboBox4.Style:=csDropDown;

combobox1.Text:=StringGrid1.cells[0,n];

combobox2.Text:=StringGrid1.cells[1,n];

DateTimePicker1.Date:=StrToDateTime(StringGrid1.cells[2,n]);

time:=Stringgrid1.Cells[3,n];

delete(time,3,5);

Combobox3.Text:=time;

time:=StringGrid1.cells[3,n];

delete(time,1,3);

Combobox4.Text:=time;

edit2.Text:=StringGrid1.cells[4,n];

edit3.Text:=StringGrid1.cells[5,n];

edit1.Text:=StringGrid1.cells[6,n];

end;

procedure TForm1.Button8Click(Sender: TObject);

var i,k,q,e: integer;

fio, inputStr,cheslo:string;

begin

k:=0;

q:=Stringgrid1.Row;

//DateTimePicker1.MaxDate := Date();

if ((edit2.Text<>'')and(edit3.Text<>'')) then

begin

cheslo:=edit2.Text;

e:=cheslo.Length;

for i := 1 to e do

begin

if(cheslo[1]='0')then

begin

delete(cheslo,1,1);

end;

end;

edit2.Text:=cheslo;

cheslo:=edit3.Text;

e:=cheslo.Length;

for i := 1 to e do

begin

if(cheslo[1]='0')then

begin

delete(cheslo,1,1);

end;

end;

edit3.Text:=cheslo;

end;

k:=0;

DateTimePicker1.MinDate := EncodeDate(2000, 1, 1);

inputStr:=Edit1.Text ;

fio := '';

for i := 1 to Length(inputStr) do

begin

if (i = 1) or(inputStr[i—1] = ' ') then

begin

fio := fio + AnsiUpperCase(inputStr[i]);

k:=k+1;

end

else

begin

fio := fio + AnsiLowercase(inputStr[i]);

end;

end;

if(Combobox1.Text='') then

messageDlg('Выберите предмет', mtError, [mbOK],0);

if(Combobox2.Text='') then

messageDlg('Выберите класс', mtError, [mbOK],0);

if((Combobox3.Text='')or(Combobox4.Text='')) then

messageDlg('Выберите время', mtError, [mbOK],0);

if (edit2.Text='') then

messageDlg('Невероне количество участников', mtError, [mbOK],0);

if (edit3.Text='') then

messageDlg('Невероне количество баллов (минимальный бал — 0, максимальный — 100)', mtError, [mbOK],0);

if ((k<>3)and(k<>4)) then

messageDlg('Невероне ФИО', mtError, [mbOK],0);

if ((edit2.Text<>'')and(edit3.Text<>'')) then

begin

if ((Combobox1.Text<>'')and(Combobox2.Text<>'')and(Combobox3.Text<>'')and (Combobox4.Text<>'') and((StrToint(edit2.text)>0)and(StrToint(edit2.text)<251)) and ((StrToint(edit3.text)>—1)and (StrToint(edit3.text)<101))and((k=3)or(k=4))) then

begin

StringGrid1.cells[0,q]:=combobox1.Text;

StringGrid1.cells[1,q]:=combobox2.Text;

StringGrid1.cells[2,q]:=FormatDateTime('',DateTimePicker1.Date);

StringGrid1.cells[3,q]:=combobox3.Text+':'+combobox4.Text;

StringGrid1.cells[4,q]:=edit2.Text;

StringGrid1.cells[5,q]:=edit3.Text;

StringGrid1.cells[6,q]:=fio;

Stringgrid1.UpdateControlState;

AssignFile(f,link\_open);

Rewrite(f);

for I:=1 to StringGrid1.RowCount—1 do

begin

a[i].predmet:=StringGrid1.Cells[0,i];

a[i].klass:=StringGrid1.Cells[1,i];

a[i].data:=StringGrid1.Cells[2,i];

a[i].time:=StringGrid1.Cells[3,i];

a[i].kolichestvo\_ushenikov:=StringGrid1.Cells[4,i];

a[i].kolichestvo\_ballov:=StringGrid1.Cells[5,i];

a[i].pobeditel:=StringGrid1.Cells[6,i];

write(f,a[i]);

end;

closefile(f);

end;

end;

end;

procedure TForm1.Button9Click(Sender: TObject);

begin

ComboBox1.Style:=csDropDown;

ComboBox2.Style:=csDropDown;

ComboBox3.Style:=csDropDown;

ComboBox4.Style:=csDropDown;

button4.Visible:=true;

button8.Visible:=false;

button9.Visible:=false;

button7.Visible:=false;

label11.Visible:=false;

combobox1.Text:=combobox1.Items[1];

combobox2.Text:=combobox2.Items[0];

DateTimePicker1.Date:=StrToDateTime('09.06.2023');

combobox3.Text:=combobox3.Items[0];

combobox4.Text:=combobox4.Items[0];

edit1.Clear;

edit2.Clear;

edit3.Clear;

end;

procedure TForm1.ComboBox1Change(Sender: TObject);

begin

ComboBox2.Style:=csDropDown;

ComboBox1.Style := csDropDownList;

if ComboBox1.ItemIndex = —1 then

messageDlg('Выберите предмет', mtError, [mbOK],0);

if((ComboBox1.text='Химия')or(ComboBox1.text='Физика')or(ComboBox1.text='Обществоведение')or(ComboBox1.text='Информатика')or(ComboBox1.text='История Беларуси')or (ComboBox1.text='География') or (ComboBox1.text='Астрономия') or (ComboBox1.text='Технический труд') or (ComboBox1.text='Биология')) then

begin

if (ComboBox1.text='Астрономия') then

begin

ComboBox2.Items.clear;

ComboBox2.text:='11';

ComboBox2.Items.add('11');

end;

if (ComboBox1.text='Технический труд') then

begin

ComboBox2.Items.clear;

ComboBox2.Items.add('5');

ComboBox2.Items.add('6');

ComboBox2.Items.add('7');

ComboBox2.Items.add('8');

ComboBox2.Items.add('9');

ComboBox2.text:='5';

end;

if ((ComboBox1.text='Биология')or(ComboBox1.text='История Беларуси')or(ComboBox1.text='Информатика')or(ComboBox1.text='География')) then

begin

ComboBox2.Items.clear;

ComboBox2.Items.add('6');

ComboBox2.Items.add('7');

ComboBox2.Items.add('8');

ComboBox2.Items.add('9');

ComboBox2.Items.add('10');

ComboBox2.Items.add('11');

ComboBox2.text:='6';

end;

if (ComboBox1.text='Обществоведение') then

begin

ComboBox2.Items.clear;

ComboBox2.Items.add('9');

ComboBox2.Items.add('10');

ComboBox2.Items.add('11');

ComboBox2.text:='9';

end;

if ((ComboBox1.text='Химия')or (ComboBox1.text='Физика')) then

begin

ComboBox2.Items.clear;

ComboBox2.Items.add('7');

ComboBox2.Items.add('8');

ComboBox2.Items.add('9');

ComboBox2.Items.add('10');

ComboBox2.Items.add('11');

ComboBox2.text:='7';

end;

end

else

begin

ComboBox2.Items.clear;

ComboBox2.Items.add('5');

ComboBox2.Items.add('6');

ComboBox2.Items.add('7');

ComboBox2.Items.add('8');

ComboBox2.Items.add('9');

ComboBox2.Items.add('10');

ComboBox2.Items.add('11');

ComboBox2.Text:='5';

end;

end;

procedure TForm1.ComboBox2Change(Sender: TObject);

begin

ComboBox2.Style := csDropDownList;

if ComboBox2.ItemIndex = —1 then

messageDlg('Выберите класс', mtError, [mbOK],0);

end;

procedure TForm1.ComboBox3Change(Sender: TObject);

begin

ComboBox3.Style := csDropDownList;

if ComboBox3.ItemIndex = —1 then

messageDlg('Выберите время', mtError, [mbOK],0);

end;

procedure TForm1.ComboBox4Change(Sender: TObject);

begin

ComboBox4.Style := csDropDownList;

if ComboBox4.ItemIndex = —1 then

messageDlg('Выберите время', mtError, [mbOK],0);

end;

procedure TForm1.ComboBox5Change(Sender: TObject);

begin

ComboBox2.Style := csDropDownList;

end;

procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);

begin

if (Key in ['0'..'9', '.', ',', '+', '\*', '/', '(', ')', '[', ']', '{', '}', '@','<', '>', '?', '\','!','#','$',';','%',':','^','&','\_','"','|','№']) then

Key := #0;

end;

procedure TForm1.Edit4KeyPress(Sender: TObject; var Key: Char);

begin

if not (Key in ['0'..'9', #8])then Key:=#0;

end;

procedure TForm1.Edit7Change(Sender: TObject);

begin

StringGrid1.Invalidate;

for i := 0 to StringGrid1.ColCount — 1 do

for j := 1 to StringGrid1.RowCount — 1 do

if Pos(AnsiLowerCase(Trim(Edit7.Text)), AnsiLowerCase(Trim(StringGrid1.Cells[i, j]))) <> 0 then

StringGrid1.Invalidate;

end;

procedure TForm1.FileListBox1Change(Sender: TObject);

begin

SendMessage(fileListBox1.Handle,LB\_DIR,0,integer(pchar('E:\Ytcheba\2 kurs\kursovai\Project\Win32\Debug\.dat')));

end;

procedure TForm1.FileListBox1DblClick(Sender: TObject);

var btn:integer;

begin

link\_open:=filelistbox1.FileName;

AssignFile(f,link\_open);

Reset(f);

n:=filesize(f);

StringGrid1.RowCount:=n+1;

for i:=1 to StringGrid1.RowCount—1 do

begin

StringGrid1.RowCount:=n+1;

read(f,a[i]);

StringGrid1.Cells[0,i]:=a[i].predmet;

StringGrid1.Cells[1,i]:=a[i].klass;

StringGrid1.Cells[2,i]:=a[i].data;

StringGrid1.Cells[3,i]:=a[i].time;

StringGrid1.Cells[4,i]:=a[i].kolichestvo\_ushenikov;

StringGrid1.Cells[5,i]:=a[i].kolichestvo\_ballov;

StringGrid1.Cells[6,i]:=a[i].pobeditel;

end;

closefile(f);

end;

procedure TForm1.FormCreate(Sender: TObject);

begin

StringGrid1.ColWidths[6] := 330;// Размер колонки с фио

StringGrid1.ColWidths[5] := 200;

StringGrid1.ColWidths[4] := 200;

StringGrid1.ColWidths[3] := 150;

StringGrid1.ColWidths[2] := 150;

StringGrid1.ColWidths[1] := 50;

StringGrid1.ColWidths[0] := 250;// Размер колонки с предметом

StringGrid1.cells[0,0]:='Учебный предмет';

StringGrid1.cells[1,0]:='Класс';

StringGrid1.cells[2,0]:='Дата проведения';

StringGrid1.cells[3,0]:='Время проведения';

StringGrid1.cells[4,0]:='Количество участников';

StringGrid1.cells[5,0]:='Количество баллов';

StringGrid1.cells[6,0]:='ФИО побеителя';

link\_open:='new file.dat';

AssignFile(f,link\_open);

Reset(f);

n:=filesize(f);

StringGrid1.RowCount:=n+1;

for i:=1 to StringGrid1.RowCount—1 do

begin

StringGrid1.RowCount:=n+1;

read(f,a[i]);

StringGrid1.Cells[0,i]:=a[i].predmet;

StringGrid1.Cells[1,i]:=a[i].klass;

StringGrid1.Cells[2,i]:=a[i].data;

StringGrid1.Cells[3,i]:=a[i].time;

StringGrid1.Cells[4,i]:=a[i].kolichestvo\_ushenikov;

StringGrid1.Cells[5,i]:=a[i].kolichestvo\_ballov;

StringGrid1.Cells[6,i]:=a[i].pobeditel;

end;

closefile(f);

stringgrid1.FixedRows:=1;

end;

procedure TForm1.ListBox1Click(Sender: TObject);

begin

SendMessage(fileListBox1.Handle,LB\_DIR,0,integer(pchar('E:\Ytcheba\2 kurs\kursovai\Project\Win32\Debug\\*.dat')))

end;

procedure TForm1.N101Click(Sender: TObject);

begin

Label9.Visible:=true;

Combobox5.Visible:=true;

Button2.Visible:=true;

end;

procedure TForm1.N11Click(Sender: TObject);

begin

Label8.Visible:=true;

Edit5.Visible:=true;

Button1.Visible:=true;

end;

procedure TForm1.N12Click(Sender: TObject);

var data: string;

begin

for i:=StringGrid1.RowCount—1 downto 1 do

begin

data:=StringGrid1.Cells[2,i];

delete(data,1,6);

if (data<>'2023') then

begin

StringGrid1.Cells[0,i]:=StringGrid1.Cells[0,i+1];

StringGrid1.Cells[1,i]:=StringGrid1.Cells[1,i+1];

StringGrid1.Cells[2,i]:=StringGrid1.Cells[2,i+1];

StringGrid1.Cells[3,i]:=StringGrid1.Cells[3,i+1];

StringGrid1.Cells[4,i]:=StringGrid1.Cells[4,i+1];

StringGrid1.Cells[5,i]:=StringGrid1.Cells[5,i+1];

StringGrid1.Cells[6,i]:=StringGrid1.Cells[6,i+1];

StringGrid1.RowCount:=StringGrid1.RowCount—1;

end;

end;

end;

procedure TForm1.N13Click(Sender: TObject);

begin

if Savedialog1.Execute then

link\_save:=PChar(Savedialog1.FileName);

AssignFile(f,link\_open);

Rewrite(f);

for I:=1 to StringGrid1.RowCount—1 do

begin

a[i].predmet:=StringGrid1.Cells[0,i];

a[i].klass:=StringGrid1.Cells[1,i];

a[i].data:=StringGrid1.Cells[2,i];

a[i].time:=StringGrid1.Cells[3,i];

a[i].kolichestvo\_ushenikov:=StringGrid1.Cells[4,i];

a[i].kolichestvo\_ballov:=StringGrid1.Cells[5,i];

a[i].pobeditel:=StringGrid1.Cells[6,i];

write(f,a[i]);

end;

closefile(f);

end;

procedure TForm1.N14Click(Sender: TObject);

begin

RadioGroup1.Visible:=true;

RadioGroup2.Visible:=false;

Button3.Visible:=true;

AssignFile(f,link\_open);

Rewrite(f);

for I:=1 to StringGrid1.RowCount—1 do

begin

a[i].predmet:=StringGrid1.Cells[0,i];

a[i].klass:=StringGrid1.Cells[1,i];

a[i].data:=StringGrid1.Cells[2,i];

a[i].time:=StringGrid1.Cells[3,i];

a[i].kolichestvo\_ushenikov:=StringGrid1.Cells[4,i];

a[i].kolichestvo\_ballov:=StringGrid1.Cells[5,i];

a[i].pobeditel:=StringGrid1.Cells[6,i];

write(f,a[i]);

end;

closefile(f);

end;

procedure TForm1.N15Click(Sender: TObject);

begin

AssignFile(f,link\_open);

Rewrite(f);

for I:=1 to StringGrid1.RowCount—1 do

begin

a[i].predmet:=StringGrid1.Cells[0,i];

a[i].klass:=StringGrid1.Cells[1,i];

a[i].data:=StringGrid1.Cells[2,i];

a[i].time:=StringGrid1.Cells[3,i];

a[i].kolichestvo\_ushenikov:=StringGrid1.Cells[4,i];

a[i].kolichestvo\_ballov:=StringGrid1.Cells[5,i];

a[i].pobeditel:=StringGrid1.Cells[6,i];

write(f,a[i]);

end;

closefile(f);

end;

procedure TForm1.N17Click(Sender: TObject);

begin

AssignFile(f,link\_open);

Reset(f);

n:=filesize(f);

StringGrid1.RowCount:=n+1;

for i:=1 to StringGrid1.RowCount—1 do

begin

StringGrid1.RowCount:=n+1;

read(f,a[i]);

StringGrid1.Cells[0,i]:=a[i].predmet;

StringGrid1.Cells[1,i]:=a[i].klass;

StringGrid1.Cells[2,i]:=a[i].data;

StringGrid1.Cells[3,i]:=a[i].time;

StringGrid1.Cells[4,i]:=a[i].kolichestvo\_ushenikov;

StringGrid1.Cells[5,i]:=a[i].kolichestvo\_ballov;

StringGrid1.Cells[6,i]:=a[i].pobeditel;

end;

closefile(f);

end;

procedure TForm1.N19Click(Sender: TObject);

begin

RadioGroup1.Visible:=false;

RadioGroup2.Visible:=true;

Button3.Visible:=true;

AssignFile(f,link\_open);

Rewrite(f);

for I:=1 to StringGrid1.RowCount—1 do

begin

a[i].predmet:=StringGrid1.Cells[0,i];

a[i].klass:=StringGrid1.Cells[1,i];

a[i].data:=StringGrid1.Cells[2,i];

a[i].time:=StringGrid1.Cells[3,i];

a[i].kolichestvo\_ushenikov:=StringGrid1.Cells[4,i];

a[i].kolichestvo\_ballov:=StringGrid1.Cells[5,i];

a[i].pobeditel:=StringGrid1.Cells[6,i];

write(f,a[i]);

end;

closefile(f);

end;

procedure TForm1.N1Click(Sender: TObject);

var i:integer;

begin

AssignFile(f,link\_open);

Reset(f);

n:=filesize(f);

StringGrid1.RowCount:=n+1;

for I:=1 to StringGrid1.RowCount—1 do

begin

read(f,a[i]);

StringGrid1.Cells[0,i]:=a[i].predmet;

StringGrid1.Cells[1,i]:=a[i].klass;

StringGrid1.Cells[2,i]:=a[i].data;

StringGrid1.Cells[3,i]:=a[i].time;

StringGrid1.Cells[4,i]:=a[i].kolichestvo\_ushenikov;

StringGrid1.Cells[5,i]:=a[i].kolichestvo\_ballov;

StringGrid1.Cells[6,i]:=a[i].pobeditel;

end;

closefile(f);

end;

procedure TForm1.N20Click(Sender: TObject);

begin

n:=stringgrid1.RowCount; //сортировка от а до я

for I := 1 to n—2 do

for j := i+1 to n—1 do

if stringgrid1.Cells[0,j] < stringgrid1.Cells[0,i] then

begin

stringgrid1.Cols[0].Exchange(j,i);

stringgrid1.Cols[1].Exchange(j,i);

stringgrid1.Cols[2].Exchange(j,i);

stringgrid1.Cols[3].Exchange(j,i);

stringgrid1.Cols[4].Exchange(j,i);

stringgrid1.Cols[5].Exchange(j,i);

stringgrid1.Cols[6].Exchange(j,i);

end;

end;

procedure TForm1.N21Click(Sender: TObject);

begin

n:=stringgrid1.RowCount; //сортировка от я до а

for I := 1 to n—2 do

for j := i+1 to n—1 do

if stringgrid1.Cells[0,j] > stringgrid1.Cells[0,i] then

begin

stringgrid1.Cols[0].Exchange(j,i);

stringgrid1.Cols[1].Exchange(j,i);

stringgrid1.Cols[2].Exchange(j,i);

stringgrid1.Cols[3].Exchange(j,i);

stringgrid1.Cols[4].Exchange(j,i);

stringgrid1.Cols[5].Exchange(j,i);

stringgrid1.Cols[6].Exchange(j,i);

end;

end;

procedure TForm1.N22Click(Sender: TObject);

begin

procedure TForm1.N27Click(Sender: TObject);

begin

n:=StringGrid1.RowCount;

for i:=n—2 downto 1 do

for j:=1 to i do

if (StrToInt(StringGrid1.Cells[5,j])>StrToInt(StringGrid1.Cells[5,j+1])) then

begin

s:=(StringGrid1.Cells[0,j]);

StringGrid1.Cells[0,j]:=StringGrid1.Cells[0,j+1];

StringGrid1.Cells[0,j+1]:=s;

s:=(StringGrid1.Cells[1,j]);

StringGrid1.Cells[1,j]:=StringGrid1.Cells[1,j+1];

StringGrid1.Cells[1,j+1]:=s;

s:=(StringGrid1.Cells[2,j]);

StringGrid1.Cells[2,j]:=StringGrid1.Cells[2,j+1];

StringGrid1.Cells[2,j+1]:=s;

s:=(StringGrid1.Cells[3,j]);

StringGrid1.Cells[3,j]:=StringGrid1.Cells[3,j+1];

StringGrid1.Cells[3,j+1]:=s;

s:=(StringGrid1.Cells[4,j]);

StringGrid1.Cells[4,j]:=StringGrid1.Cells[4,j+1];

StringGrid1.Cells[4,j+1]:=s;

s:=(StringGrid1.Cells[5,j]);

StringGrid1.Cells[5,j]:=StringGrid1.Cells[5,j+1];

StringGrid1.Cells[5,j+1]:=s;

s:=(StringGrid1.Cells[6,j]);

StringGrid1.Cells[6,j]:=StringGrid1.Cells[6,j+1];

StringGrid1.Cells[6,j+1]:=s;

end;

end;

procedure TForm1.N28Click(Sender: TObject);

begin

n:=StringGrid1.RowCount;

for i:=n—2 downto 1 do

for j:=1 to i do

if (StrToInt(StringGrid1.Cells[5,j])<StrToInt(StringGrid1.Cells[5,j+1])) then

begin

s:=(StringGrid1.Cells[0,j]);

StringGrid1.Cells[0,j]:=StringGrid1.Cells[0,j+1];

StringGrid1.Cells[0,j+1]:=s;

s:=(StringGrid1.Cells[1,j]);

StringGrid1.Cells[1,j]:=StringGrid1.Cells[1,j+1];

StringGrid1.Cells[1,j+1]:=s;

s:=(StringGrid1.Cells[2,j]);

StringGrid1.Cells[2,j]:=StringGrid1.Cells[2,j+1];

StringGrid1.Cells[2,j+1]:=s;

s:=(StringGrid1.Cells[3,j]);

StringGrid1.Cells[3,j]:=StringGrid1.Cells[3,j+1];

StringGrid1.Cells[3,j+1]:=s;

s:=(StringGrid1.Cells[4,j]);

StringGrid1.Cells[4,j]:=StringGrid1.Cells[4,j+1];

StringGrid1.Cells[4,j+1]:=s;

s:=(StringGrid1.Cells[5,j]);

StringGrid1.Cells[5,j]:=StringGrid1.Cells[5,j+1];

StringGrid1.Cells[5,j+1]:=s;

s:=(StringGrid1.Cells[6,j]);

StringGrid1.Cells[6,j]:=StringGrid1.Cells[6,j+1];

StringGrid1.Cells[6,j+1]:=s;

end;

end;

procedure TForm1.N2Click(Sender: TObject);

begin

AssignFile(f,link\_open);

rewrite(f);

for I:=1 to StringGrid1.RowCount—1 do

begin

a[i].predmet:=StringGrid1.Cells[0,i];

a[i].klass:=StringGrid1.Cells[1,i];

a[i].data:=StringGrid1.Cells[2,i];

a[i].time:=StringGrid1.Cells[3,i];

a[i].kolichestvo\_ushenikov:=StringGrid1.Cells[4,i];

a[i].kolichestvo\_ballov:=StringGrid1.Cells[5,i];

a[i].pobeditel:=StringGrid1.Cells[6,i];

write(f,a[i]);

end;

closefile(f);

end;

procedure TForm1.N8Click(Sender: TObject);

begin

if Opendialog1.Execute then

link\_open:=PChar(Opendialog1.FileName);

AssignFile(f,link\_open);

Reset(f);

n:=filesize(f);

StringGrid1.RowCount:=n+1;

for i:=1 to StringGrid1.RowCount—1 do

begin

StringGrid1.RowCount:=n+1;

read(f,a[i]);

StringGrid1.Cells[0,i]:=a[i].predmet;

StringGrid1.Cells[1,i]:=a[i].klass;

StringGrid1.Cells[2,i]:=a[i].data;

StringGrid1.Cells[3,i]:=a[i].time;

StringGrid1.Cells[4,i]:=a[i].kolichestvo\_ushenikov;

StringGrid1.Cells[5,i]:=a[i].kolichestvo\_ballov;

StringGrid1.Cells[6,i]:=a[i].pobeditel;

end;

closefile(f);

end;

procedure TForm1.N9Click(Sender: TObject);

begin

for i:=1 to StringGrid1.RowCount—1 do

begin

StringGrid1.Cells[0,i]:=StringGrid1.Cells[0,i+1];

StringGrid1.Cells[1,i]:=StringGrid1.Cells[1,i+1];

StringGrid1.Cells[2,i]:=StringGrid1.Cells[2,i+1];

StringGrid1.Cells[3,i]:=StringGrid1.Cells[3,i+1];

StringGrid1.Cells[4,i]:=StringGrid1.Cells[4,i+1];

StringGrid1.Cells[5,i]:=StringGrid1.Cells[5,i+1];

StringGrid1.Cells[6,i]:=StringGrid1.Cells[6,i+1];

StringGrid1.RowCount:=StringGrid1.RowCount—1;

end;

end;

procedure TForm1.NewFIle1Click(Sender: TObject);

begin

if Savedialog1.Execute then

link\_open:=PChar(Savedialog1.FileName);

AssignFile(f,link\_open);

Rewrite(f);

for I:=1 to StringGrid1.RowCount—1 do

begin

StringGrid1.Cells[0,i]:=StringGrid1.Cells[0,i+1];

StringGrid1.Cells[1,i]:=StringGrid1.Cells[1,i+1];

StringGrid1.Cells[2,i]:=StringGrid1.Cells[2,i+1];

StringGrid1.Cells[3,i]:=StringGrid1.Cells[3,i+1];

StringGrid1.Cells[4,i]:=StringGrid1.Cells[4,i+1];

StringGrid1.Cells[5,i]:=StringGrid1.Cells[5,i+1];

StringGrid1.Cells[6,i]:=StringGrid1.Cells[6,i+1];

StringGrid1.RowCount:=StringGrid1.RowCount—1;

write(f,a[i]);

filelistbox1.Update;

end;

if Opendialog1.Execute then

link\_open:=PChar(Opendialog1.FileName);

AssignFile(f,link\_open);

AssignFile(f,link\_open);

Rewrite(f);

n:=filesize(f);

StringGrid1.RowCount:=n+1;

for I:=1 to StringGrid1.RowCount—1 do

begin

read(f,a[i]);

StringGrid1.Cells[0,i]:='';

StringGrid1.Cells[1,i]:='';

StringGrid1.Cells[2,i]:='';

StringGrid1.Cells[3,i]:='';

StringGrid1.Cells[4,i]:='';

StringGrid1.Cells[5,i]:='';

StringGrid1.Cells[6,i]:='';

end;

closefile(f);

end;

procedure TForm1.RadioGroup1Click(Sender: TObject);

var x:integer;

begin

Stringgrid1.RowCount:=1;

assignfile(f,link\_open);

reset(f);

n:=filesize(f);

for i := 1 to n do

begin

read(f,a[i]);

if a[i].predmet=RadioGroup1.items[RadioGroup1.ItemIndex] then

begin

Stringgrid1.RowCount:=Stringgrid1.RowCount+1;

stringgrid1.cells[0,Stringgrid1.RowCount—1]:=a[i].predmet;

stringgrid1.cells[1,Stringgrid1.RowCount—1]:=a[i].klass;

stringgrid1.cells[2,Stringgrid1.RowCount—1]:=a[i].data;

stringgrid1.cells[3,Stringgrid1.RowCount—1]:=a[i].time;

stringgrid1.cells[4,Stringgrid1.RowCount—1]:=a[i].kolichestvo\_ushenikov;

stringgrid1.cells[5,Stringgrid1.RowCount—1]:=a[i].kolichestvo\_ballov;

stringgrid1.cells[6,Stringgrid1.RowCount—1]:=a[i].pobeditel;

end;

end;

closefile(f);

end;

procedure TForm1.RadioGroup2Click(Sender: TObject);

begin

Stringgrid1.RowCount:=1;

assignfile(f,link\_open);

reset(f);

n:=filesize(f);

for i := 1 to n do

begin

read(f,a[i]);

if a[i].klass=RadioGroup2.items[RadioGroup2.ItemIndex] then

begin

Stringgrid1.RowCount:=Stringgrid1.RowCount+1;

stringgrid1.cells[0,Stringgrid1.RowCount—1]:=a[i].predmet;

stringgrid1.cells[1,Stringgrid1.RowCount—1]:=a[i].klass;

stringgrid1.cells[2,Stringgrid1.RowCount—1]:=a[i].data;

stringgrid1.cells[3,Stringgrid1.RowCount—1]:=a[i].time;

stringgrid1.cells[4,Stringgrid1.RowCount—1]:=a[i].kolichestvo\_ushenikov;

stringgrid1.cells[5,Stringgrid1.RowCount—1]:=a[i].kolichestvo\_ballov;

stringgrid1.cells[6,Stringgrid1.RowCount—1]:=a[i].pobeditel;

end;

end;

closefile(f);

end;

procedure TForm1.StringGrid1DrawCell(Sender: TObject; ACol, ARow: Integer;

Rect: TRect; State: TGridDrawState);

begin

if (ARow > 0) and (Pos(AnsiLowerCase(Trim(Edit7.Text)),

AnsiLowerCase(Trim(StringGrid1.Cells[ACol, ARow]))) <> 0) then

begin

StringGrid1.Canvas.Brush.Color := $00FF00;

StringGrid1.Canvas.FillRect(Rect);

StringGrid1.Canvas.Font.Color := clBlack;

StringGrid1.Canvas.Font.Style := [fsUnderline];

StringGrid1.Canvas.TextOut(Rect.Left + 6, Rect.Top + 6, StringGrid1.Cells[ACol, ARow]);

end;

end;

end.

Текст модуля unit2.pas

unit Unit2;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.ExtCtrls, Vcl.Buttons;

type

TForm2 = class(TForm)

Button1: TButton;

Image1: TImage;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

procedure Button1Click(Sender: TObject);

procedure FormCreate(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form2: TForm2;

implementation

{$R \*.dfm}

uses Unit1;

procedure TForm2.Button1Click(Sender: TObject);

begin

Form1.ShowModal;

Form2.Close;

end;

procedure TForm2.FormCreate(Sender: TObject);

begin

Image1.Picture.LoadFromFile('E:\Ytcheba\2 kurs\kursovai\Project\image\Form\_1.bmp');

end;

end.