Министерство образования и науки Российской Федерации

**Муромский институт (филиал)**

федерального государственного бюджетного образовательного учреждения

высшего профессионального образования

**«Владимирский государственный университет**

**имени Александра Григорьевича и Николая Григорьевича Столетовых»**

(МИ (филиал) ВлГУ)

Факультет ИТР

Кафедра ФПМ

*КУРСОВАЯ РАБОТА*

по дисциплине Базы данных

Тема АИС учета успеваемости и достижений учащихся школы

Руководитель

Колпаков А.А.

(оценка)  (уч. степень, звание)

(подпись) (дата)

Члены комиссии Студент ПМИ-122

(группа)

Штырлов Д.М.

(подпись) (дата)

(подпись) (дата) (подпись) (дата)

Муром 2024 г.

В работе представлена автоматизированная информационная система (АИС). Приложение разработано на платформе Windows. Цель работы - рассмотреть принципы разработки АИС и разработать программу на основе этих принципов.

Для разработки были выбраны язык программирования – Python 3.13 и среда разработки – Visual Studio Code. Для создания базы данных использовалась СУБД SQLite.

В разработанном приложении доступны возможность добавления, изменения и поиска записей в базе данных.

The paper presents an automated information system (AIS). The application is developed on the Windows platform. The purpose of the work is to consider the principles of AIS development and develop a program based on these principles.

For the development were chosen programming language - Python 3.13 and development environment - Visual Studio Code. SQLite DBMS was used to create the database.

The developed application provides the ability to add, modify and search records in the database.

Содержание

[Введение 6](#_Toc187343445)

[Постановка проблемы и формулировка целей и задач 6](#_Toc187343446)

[1.Анализ технического задания 8](#_Toc187343447)

[2.Разработка моделей данных 10](#_Toc187343448)

[2.1. Концептуальная модель 10](#_Toc187343449)

[2.2. Разработка физической модели 11](#_Toc187343450)

[3. Разработка и реализация АИС 12](#_Toc187343451)

[3.1. Разработка серверной части 12](#_Toc187343452)

[3.2 SQL запросы 13](#_Toc187343453)

[3.3 Разработка пользовательской части 16](#_Toc187343454)

[3.4 Руководство программисту: 21](#_Toc187343455)

[3.5 Руководство пользователю. 22](#_Toc187343456)

[4. Тестирование АИС 23](#_Toc187343457)

[Заключение 29](#_Toc187343458)

[Список используемой литературы 30](#_Toc187343459)

[Приложение 1. Модели данных 31](#_Toc187343460)

[Приложение 2. Текст программы 33](#_Toc187343465)

[Приложение 3. Снимки окон программы 33](#_Toc187343466)

# Введение

Современные образовательные учреждения стремятся к оптимизации процесса учета успеваемости и достижений учащихся. Автоматизированные информационные системы (АИС) играют важную роль в этом процессе, позволяя автоматизировать сбор, хранение и анализ данных о результатах обучения.

АИС учета успеваемости и достижений учащихся в школе обеспечивает эффективный мониторинг успеваемости. Это способствует созданию более прозрачной и эффективной образовательной среды.

Веб-интерфейсы и мобильные приложения, входящие в состав АИС, упрощают доступ к информации и способствуют вовлечению всех участников образовательного процесса.

Таким образом, АИС становится незаменимым инструментом для повышения качества образования и формирования успешной образовательной стратегии в условиях современной школы.

## Постановка проблемы и формулировка целей и задач

Современные образовательные учреждения сталкиваются с множеством сложностей в учете успеваемости и достижений учащихся. Важные трудности включают:

1. Сложность ручного учета успеваемости, что приводит к ошибкам и потере данных.

2. Недостаток прозрачности для родителей и учащихся в отношении текущих результатов и достижения целей.

3. Невозможность оперативно анализировать данные и делать выводы о прогрессе учеников.

Для решения указанных проблем необходимо разработать автоматизированную информационную систему (АИС) с конкретными задачами:

1. Автоматизация процесса учета успеваемости, включая выставление оценок.

2. Создание единой базы данных, которая будет доступна для всех участников образовательного процесса.

3. Разработка инструментов аналитики, позволяющих отслеживать динамику успеваемости и выявлять проблемные области.

4. Интеграция системы с существующими образовательными ресурсами для улучшения взаимодействия.

Цель: разработка функциональной и удобной автоматизированной информационной системы (АИС) для пользователей школьных учреждений, обеспечивающей автоматизацию ключевых процессов учета успеваемости и оценивания.

# 1.Анализ технического задания

В данной курсовой работе поставлена задача создания АИС учета успеваемости и достижений учащихся школы. Для разработки такой системы необходимы СУБД для создания базы данных и среда разработки для создания интерфейса и функций по работе с БД.

Согласно заданию, в программе нужно учесть следующие особенности:

1) вести учет учащихся и их достижений и успеваемости;

2) для указанной дисциплины получить количество часов и ведущего преподавателя;

предоставить возможность добавления и изменения информации об учащихся, об учебных планах, о журнале успеваемости при этом предусмотреть обработку исключительных ситуаций;

3) для указанного ученика получить его успеваемость по всем дисциплинам за указанный срок и количество пропусков занятий;

4) предоставить возможность добавления и изменения информации в журнале успеваемости.

• В БД предусмотреть хранение изображений (минимум в одном поле) в соответствии с тематикой курсовой работы.

• В БД должны быть реализованы хранимые процедуры и/или триггеры.

• в БД должно быть занесено суммарно не менее 50 записей.

Исходя из этих данных, БД должна содержать 5 таблиц: учащиеся, учителя, дисциплины, оценивание, достижения. Для разработки БД в рамках поставленной задачи можно использовать SQLite в качестве СУБД, и SQLite Studio в качестве графического приложения для отслеживания успешности работы в самом начале написания программного кода.

Достоинством SQLite является:

1. Легковесность: SQLite — это небольшой, самодостаточный движок базы данных, что позволяет легко интегрировать его в приложения.

2. Простота использования: не требует установки и настройки, достаточно просто включить библиотеку в проект.

3. Файловая структура: все данные хранятся в одном файле, что облегчает резервное копирование и перенос.

4.Отсутствие серверной архитектуры: работает без необходимости настройки серверной части, что снижает нагрузку и делает его более подходящим для небольших приложений.

5. Высокая производительность: оптимизирован для быстрого выполнения запросов, особенно в условиях небольших и средних объемов данных.

6. Поддержка транзакций: обеспечивает ACID-соответствие, гарантируя целостность данных.

7. Кросс-платформенность: работает на различных операционных системах, включая Windows, macOS, Linux и мобильные платформы.

8. Широкая поддержка языков: поддерживается множеством языков программирования благодаря доступным библиотекам.

Для написания программы будет использоваться язык Python 3.13. Для разработки графической части будет использоваться библиотека tkinter. Для того чтобы создать программу, необходимо учесть то, что она создается, прежде всего, для пользователя, и поэтому немаловажным требованием к программе должен стать удобный и интуитивно понятный интерфейс. Необходимо предусмотреть все возможности управления приложением, чтобы упростить работу пользователя и максимально обеспечить эффективность работы.

Программа должна правильно работать с данными. Приложение должно мгновенно реагировать на действия пользователя и в зависимости от запроса с его стороны формировать выходной результат.

# 2.Разработка моделей данных

Этот этап играет ключевую роль в разработке приложения для работы с базой данных. На этом этапе выделяются сущности, их атрибуты и взаимосвязи между ними. Создание и настройка базы данных в выбранной системе управления осуществляется на основе логической модели, которая затем преобразуется в физическую. Этот процесс включает в себя определение структуры таблиц, индексов, ключей и других объектов базы данных.

## 2.1. Концептуальная модель

Создание концептуальной модели начинается с анализа предметной области и выделения сущностей.

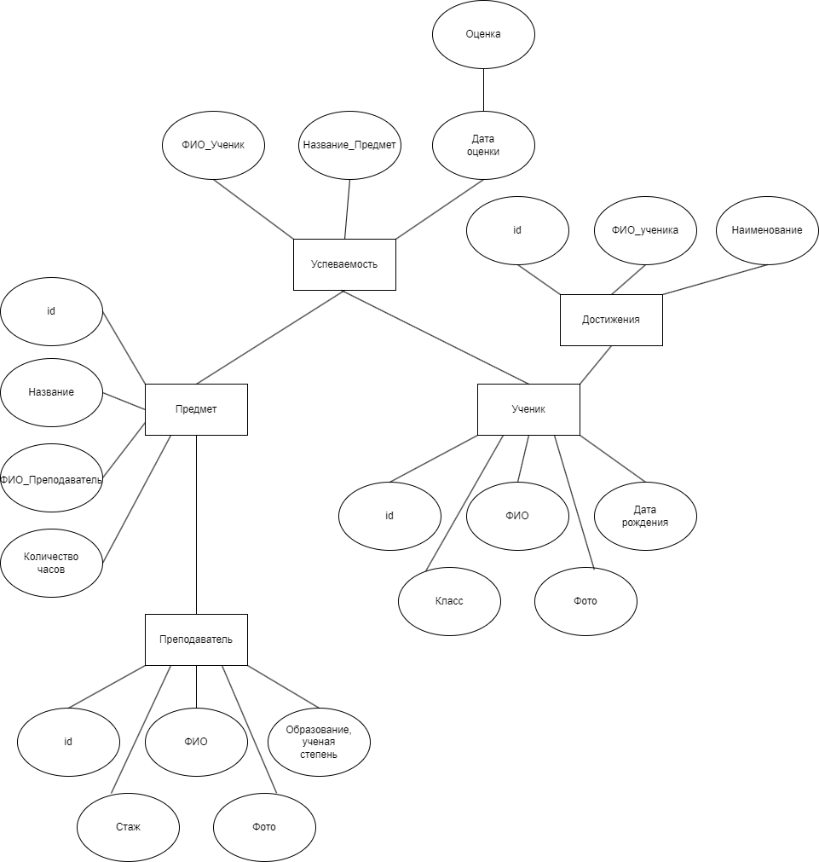


Рис.1 концептуальная модель данных

## 2.2. Разработка физической модели

Физическая модель отражает конкретный принцип работы базы данных, включая типизацию данных и связи между таблицами.

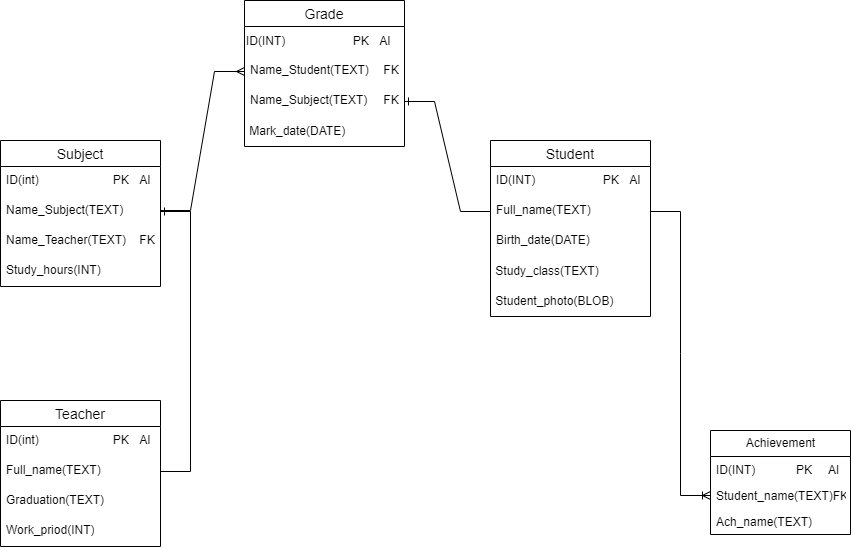


Рис.2 физическая модель

В результате разработки этих моделей достигается понимание оптимизации структуры данных и обеспечение эффективного хранения и обработки информации. Каждая модель играет свою роль в процессе создания базы данных, начиная с общего представления данных и заканчивая их конкретной реализацией.

# 3. Разработка и реализация АИС

## 3.1. Разработка серверной части

На данном этапе с помощью программного обеспечения SQLiteStudio будет создана сама база данных или иначе серверная часть создаваемой информационной системы.

Для работы будет использоваться созданная ранее физическая модель, которая дает полное представление о том как должна быть построена логика работы базы данных. Исходя из модели были созданы следующие таблицы:

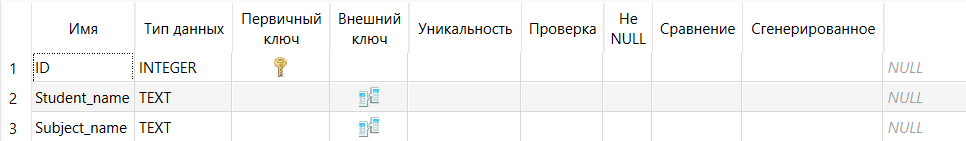


Рис. 3.1 – таблица учета успеваемости

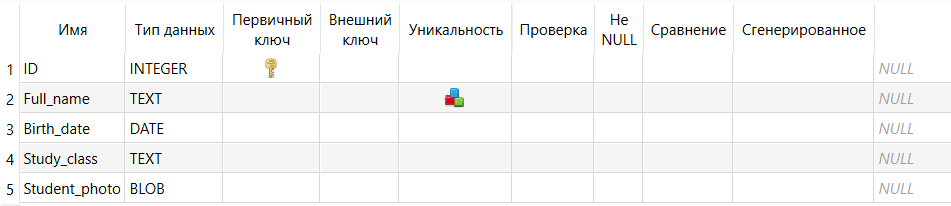


Рис. 3.2 – таблица учащихся

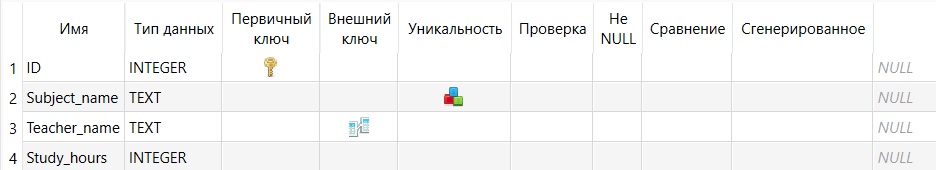


Рис. 3.3 – таблица дисциплин

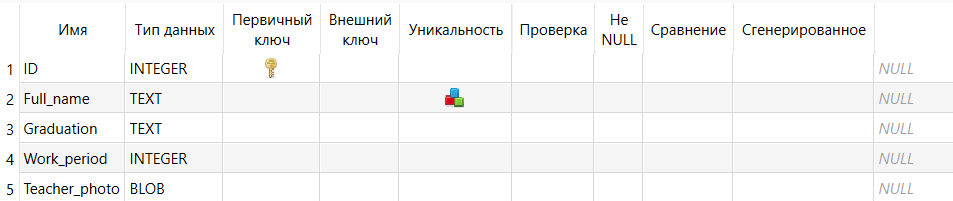


Рис. 3.4 – таблица преподавателей

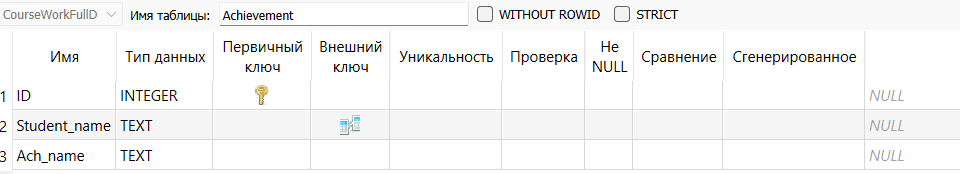


Рис. 3.5 – таблица достижений учащихся

На рисунке 3.1 предсатленна таблица учета успеваемости, основной целью которой будет хранение собственно оценок отнощасихся к определенному ученику и предмету.

На рисунке 3.2 представлена таблица учащихся, имеющая такие столбцы как, полное имя учащегося, дата рождения, класс или группа прохождения обучения и его фото.

На рисунке 3.3 представлена таблица предметов обучения, содержит такую информацию как, название дисциплины, имя преподавателя проводящего занятия по указанной дисциплине, и количество часов обучения.

На рисунке 3.4 представлена таблица преподавателей, которая содержит такую информацию как, имя преподавателя, его ученая степень, стаж работы и фото.

На рисунке 3.5 показана таблица достижений учащихся хранящая информацию о ученике и присужденных ему достижениях в учебе.

## 3.2 SQL запросы

SQL запросы для создания таблиц соответственно по порядку упоменания:

CREATE TABLE Grade (

ID INTEGER PRIMARY KEY AUTOINCREMENT,

Student\_name TEXT REFERENCES Student (Full\_name) ON DELETE CASCADE

ON UPDATE CASCADE,

Subject\_name TEXT REFERENCES Subject (Subject\_name) ON DELETE CASCADE

ON UPDATE CASCADE,

);

CREATE TABLE Student (

ID INTEGER PRIMARY KEY AUTOINCREMENT,

Full\_name TEXT UNIQUE,

Birth\_date DATE,

Study\_class TEXT,

Student\_photo BLOB

);

CREATE TABLE Subject (

ID INTEGER PRIMARY KEY AUTOINCREMENT,

Subject\_name TEXT UNIQUE,

Teacher\_name TEXT REFERENCES Teacher (Full\_name) ON DELETE CASCADE

ON UPDATE CASCADE,

Study\_hours INTEGER

);

CREATE TABLE Teacher (

ID INTEGER PRIMARY KEY AUTOINCREMENT,

Full\_name TEXT UNIQUE,

Graduation TEXT,

Work\_period INTEGER,

Teacher\_photo BLOB

);

CREATE TABLE Achievement (

ID INTEGER PRIMARY KEY AUTOINCREMENT,

Student\_name TEXT REFERENCES Student (Full\_name),

Ach\_name TEXT

);

В работе же с уже созданными таблицами будут использованы следующие основные запросы:

1. SELECT возвращает определенный набор, данный из таблицы. Синтаксис:

SELECT Столбец1, Столбец2, Столбец3, … FROM Таблица

В случае рассматриваемой АИС будет использован повсеместно для просмотра данных, к примеры для получения списка имен учащихся.

2.WHERE не является запросом сам по себе и используется для фильтрации наборов данный вместе с SELECT.

Точно также как и SELECT будет использован практически везде, для сортировки и отбора получаемых из запросов данных.

3. INSERT используется для вставки строк в таблицу.

Так же часто используемый запрос будет применяться для заполнения пользователем таблицы, например добавление учащихся или выставление оценок.

4. ALTER TABLE … ADD конструкция, используемая для **добавления, удаления или изменения столбцов в существующей таблице**.

В данной работе имеет конкретную цель так как выставляемая оценка привязана конкретно к определенной дате необходимо при каждый раз добавлять дополнительный столбец даты.

## 3.3 Разработка пользовательской части

Пользовательский интерфейс представляет собой способ взаимодействия человека, в данном случае преподавателя или работника образовательно ечреждения с функционалом предоставляемым приложением.

Интерфейс должен быть удобен, понятен и главное прост в использование. В данном случае для создания интерфейса была использована библиотека tkinter для языка программирования python, отличающаяся простотой разработки и удобством взаимодействия. Дополнительно сам язык Python имеет встроенную библиотеку sqlite3 для работы с базами данных SQLite. Внешний вид разработанного интерфейса:

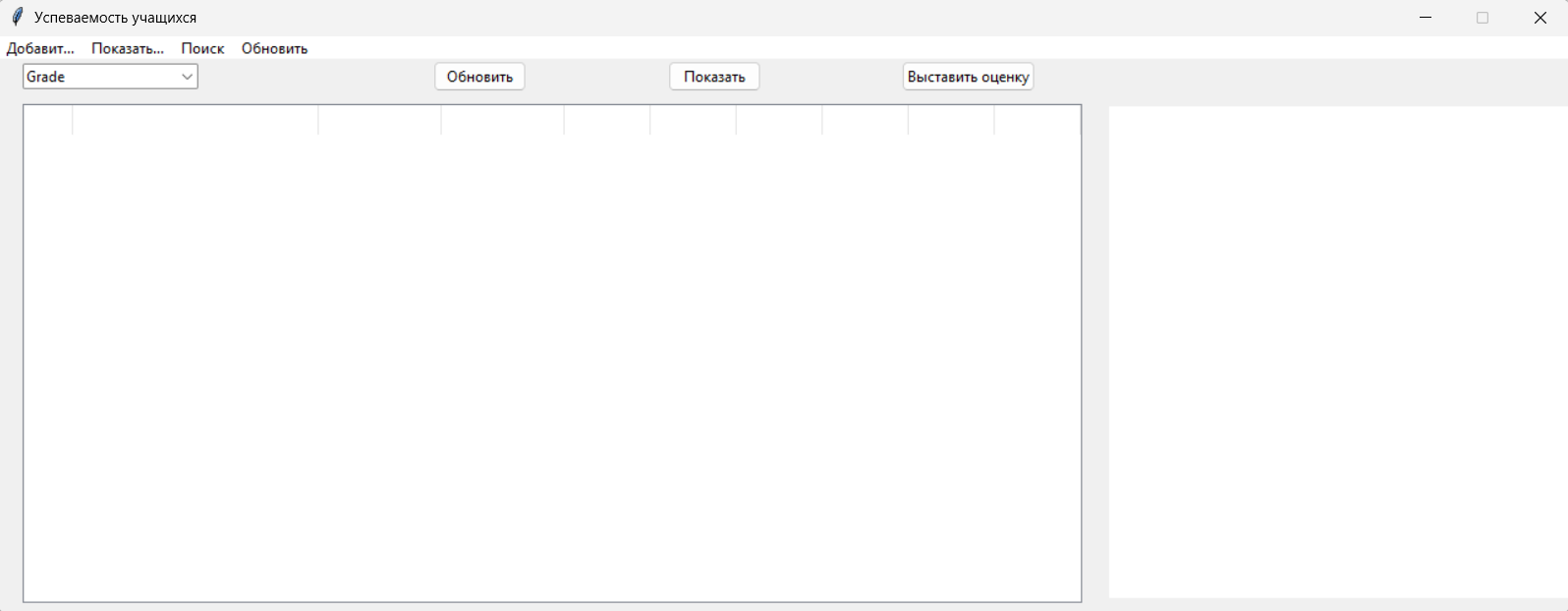


Рис. 4.1 – основное окно приложения



Рис 4.2

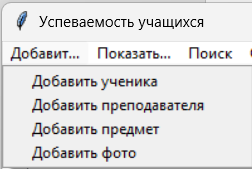


Рис 4.3

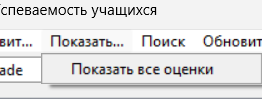


Рис 4.4

На рисунке 4.1 мы можем видеть основные элементы программы которые будут встречаться и на последующих рисунках:

1. Меню окна – рисунок 4.2, дублирует некоторые элементы управления, но также и содержит в себе следующие функции:

1. Добавить… (Рисунок 4.3)

1.1. Добавить ученика

1.2. Добавить преподавателя

1.3. Добавить предмет

1.4. Добавить фото

2. Показать… (Рисунок 4.4)

2.1. Показать все оценки

2.2. Показать достижения

3. Поиск

4. Отчет

5. Удалить оценку

6. Обновить

2. Табличное представление базы данных

3. Кнопки управления:

1. Обновить – обновляет данные и содержимое табличного представления

2. Показать – отображает хранящееся изображение на расположенном элементе холста

3. Выставить оценку – открывает окно управления оценками

4. Элемент выбора отображаемой таблицы

5. Элемент отображения изображения (холст)

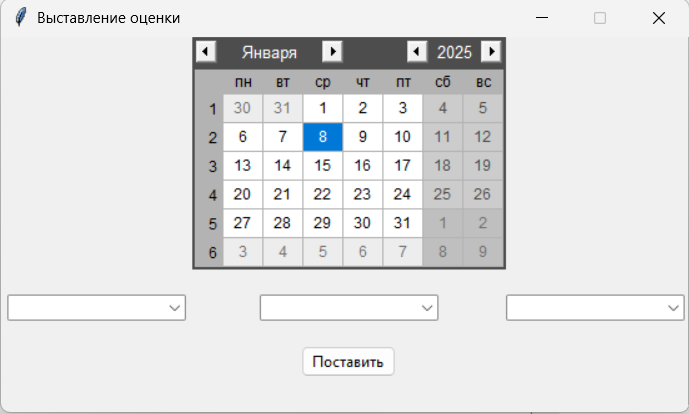


Рис. 4.5 – окно выставления оценки

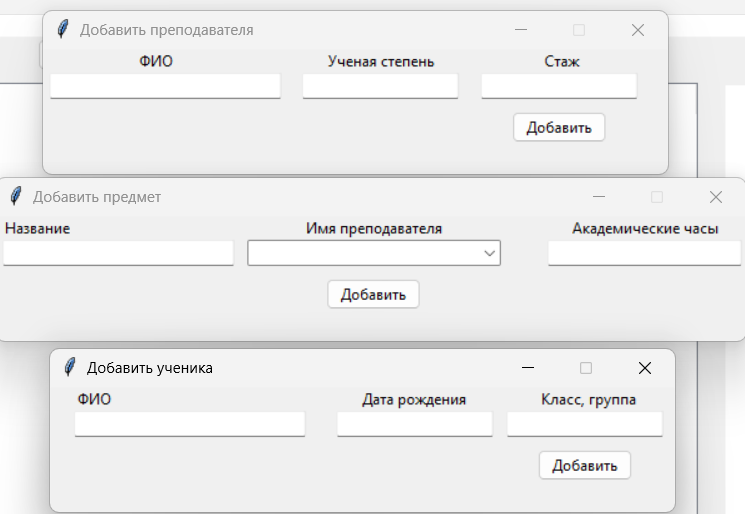


Рис. 4.6 – окна добавления преподавателя, предмета и учащегося (снизу-вверх)

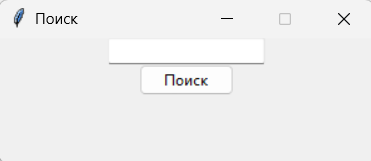


Рис. 4.7 – Окно поиска

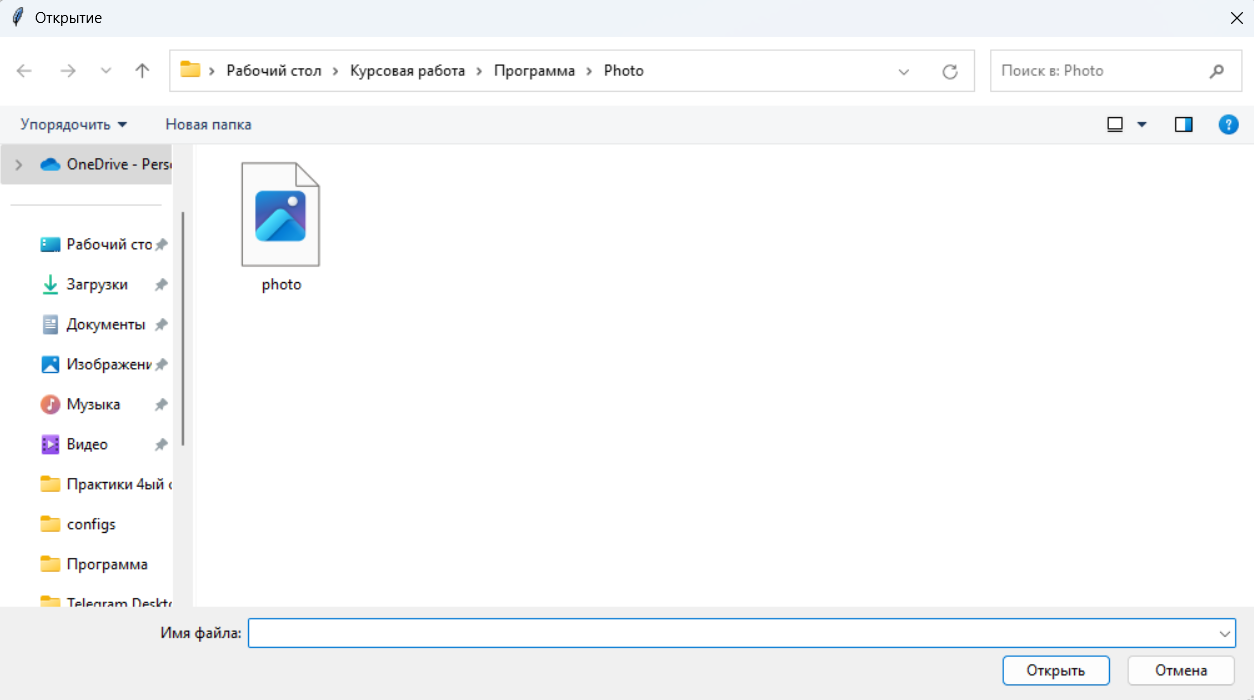


Рис. 4.8 – окно добавления фото

## 3.4 Руководство программисту:

def add\_current\_date() – функция добавления текущей даты как колонки в таблицу оценивания.

def write\_to\_file(data, filename), def convert\_to\_binary\_data(filename), def add\_photo\_to\_chosen(), def get\_image\_from\_database(id), def show\_image\_in\_canvas() – функции для хранения фотографий в базе данных сначала фото в формате PNG или GIF переводиться в бинарный вид и записываются в базу, затем в обратном порядке изымаются и отображаться в окне программы.

def grade\_complition() – функция заполняет таблицу оценивания, создавая пары ученик-предмет.

def add\_student\_data\_to\_database() – функция добавления ученика в базу данных.

def add\_teacher\_to\_database() – функция добавления преподавателя в базу данных.

def add\_subject\_to\_database() – функция добавления предмета в базу данных.

def refresh\_viewport\_data() – функция обновления таблицы отображения базы данных.

def mark\_set() – функция выставления оценки учащимуся.

def delete\_mark\_from\_date() – функция удаления оценки учащемуся.

def show\_all\_marks() – функция открытия окна для показа всех оценок.

def avg\_mark() – функция расчета средней оценки у учащегося.

def search\_viewport() – функция поиска в таблице отображения базы данных.

def show\_achievements() – функция открытия окна показа достижений учащихся.

## 3.5 Руководство пользователю.

После запуска приложения откроется графический интерфейс приложения, для корректной работы следует произвести следующие шаги:

1. Добавить учителя используя элемент меню “Добавить преподавателя”, проверить корректность добавления выбрав таблицу из выпадающего списка и нажав кнопку обновить

2. Добавить фото преподавателю, выбрав его нажатием левой кнопки мыши и выбрав элемент меню “Добавить фото”, проверить правильность нажатием кнопки “Показать”

3. Добавить предмет выбрав элемент меню “Добавить предмет”, указав его название и выбрать имеющегося преподавателя в открывшемся окне

4. Добавить ученика выбрав элемент меню “Добавить ученика”, проверить корректность добавления и также добавить при необходимости фото, как указано в пункте 2

5. Перейти в таблицу “Grade” выбрав пункт выпадающего списка и проверить корректность добавления пар ученик-предмет

6. Выставить оценку нажав соответствующую кнопку, выбрать ученика, предмет, дату и оценку к выставлению

7. При необходимости просмотреть все оценки, удалить оценку или посмотреть отчет можно нажав соответствующие пункты меню

8. Выбрав пункт меню “Показать достижения” просмотреть или добавить достижения каждого ученика

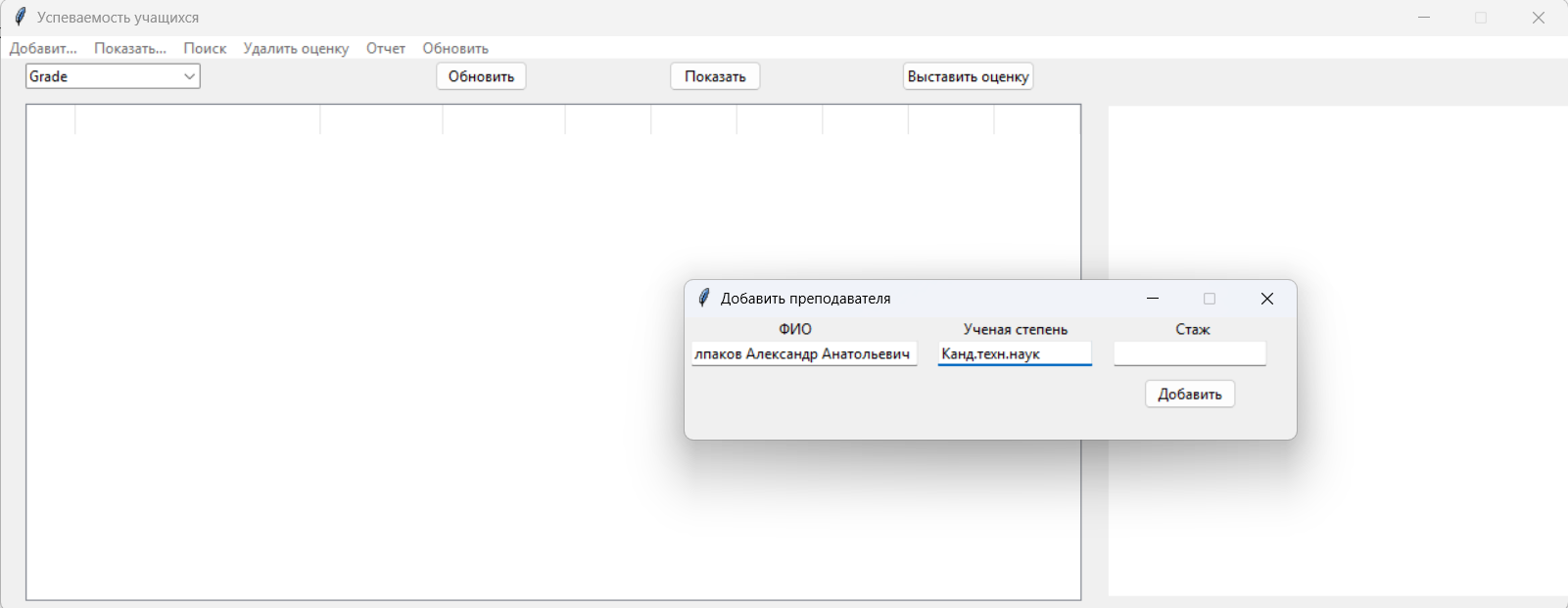
# 4. Тестирование АИС

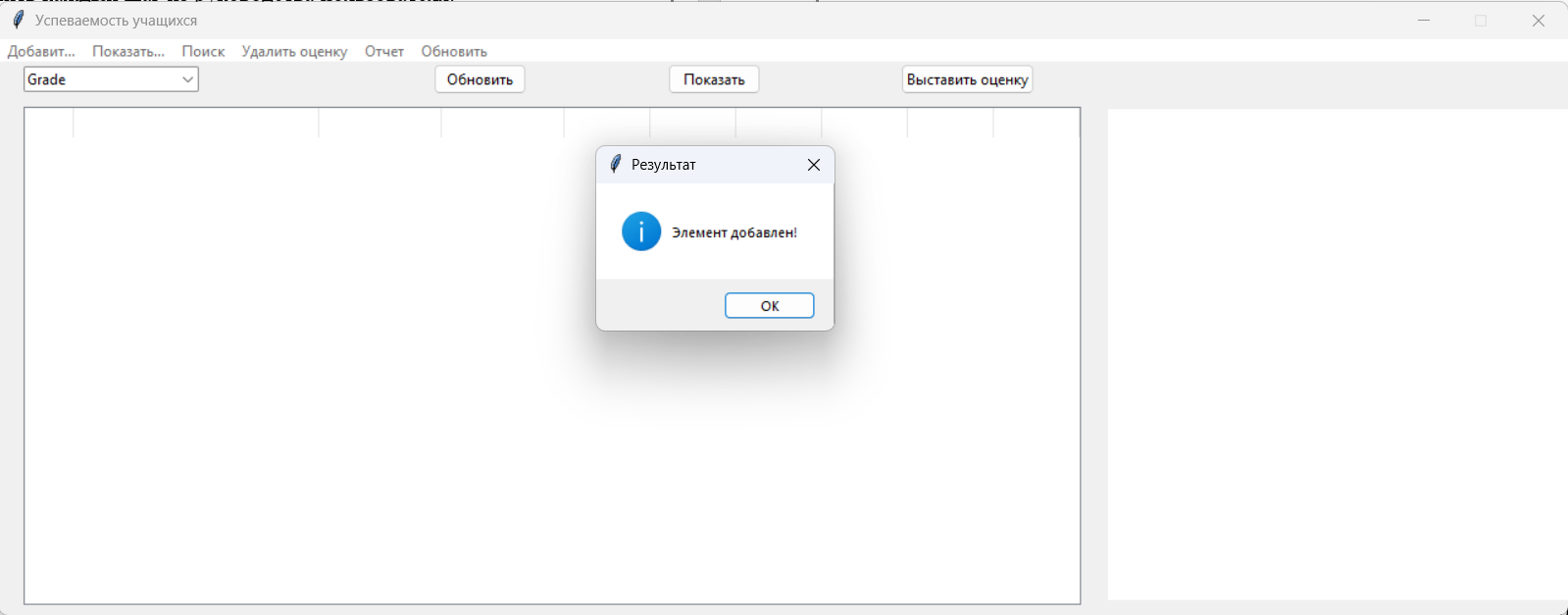
После написания приложения необходимо проверить, как будет вести себя при различных действиях пользователя.

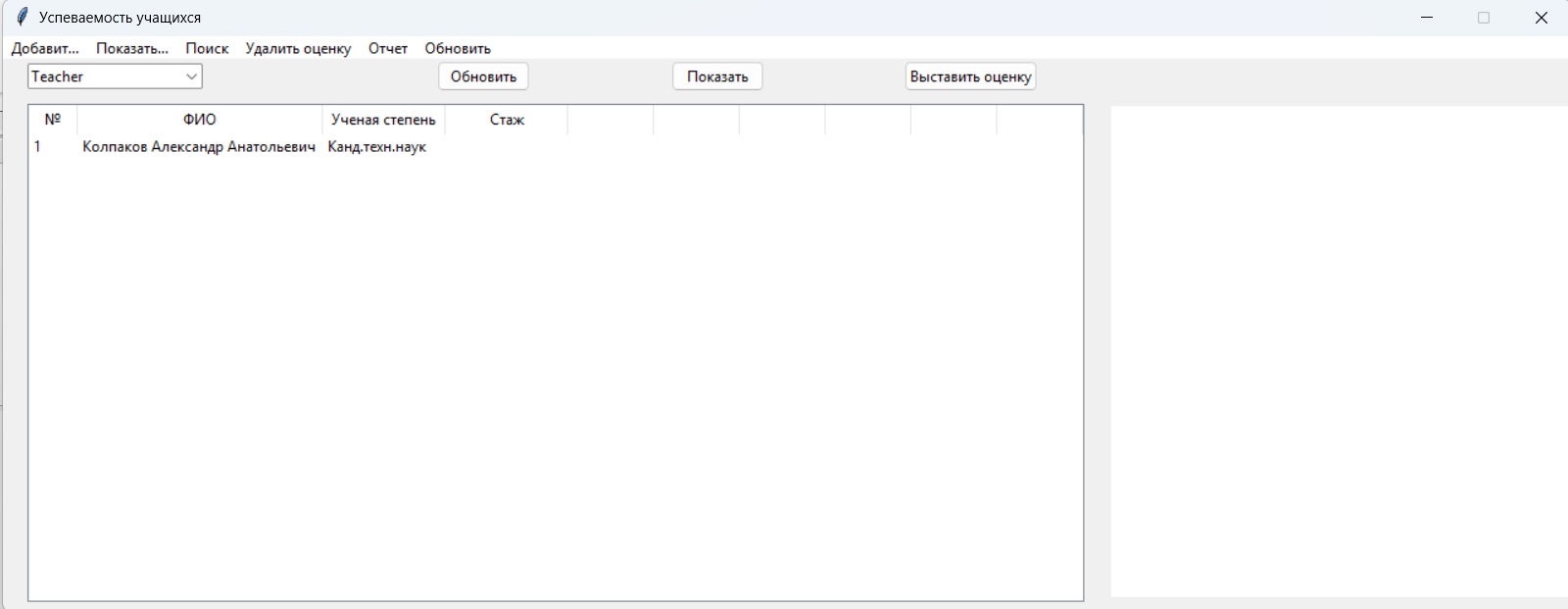
Стоит выполнить каждый шаг из руководства пользователя.

Принцип работы по шагам:

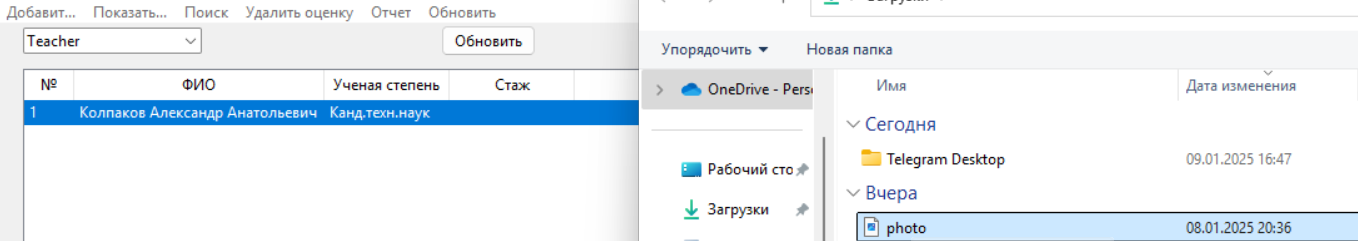
1.

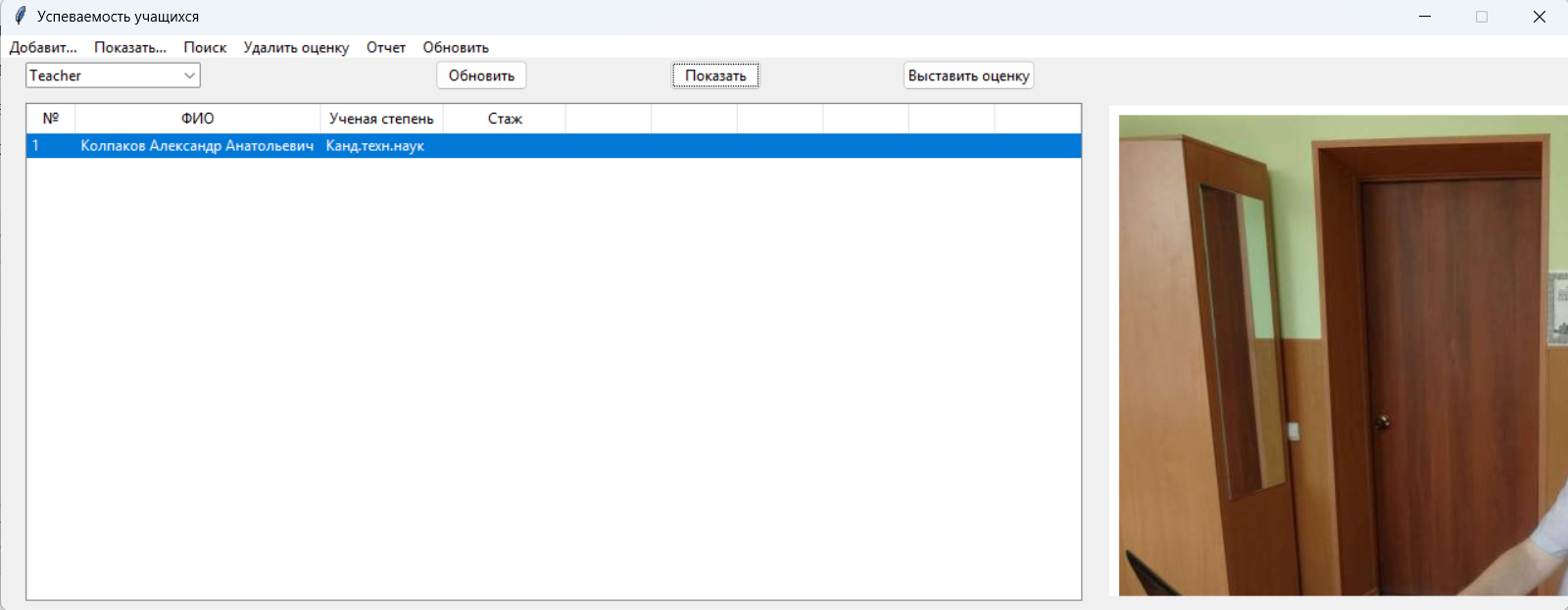




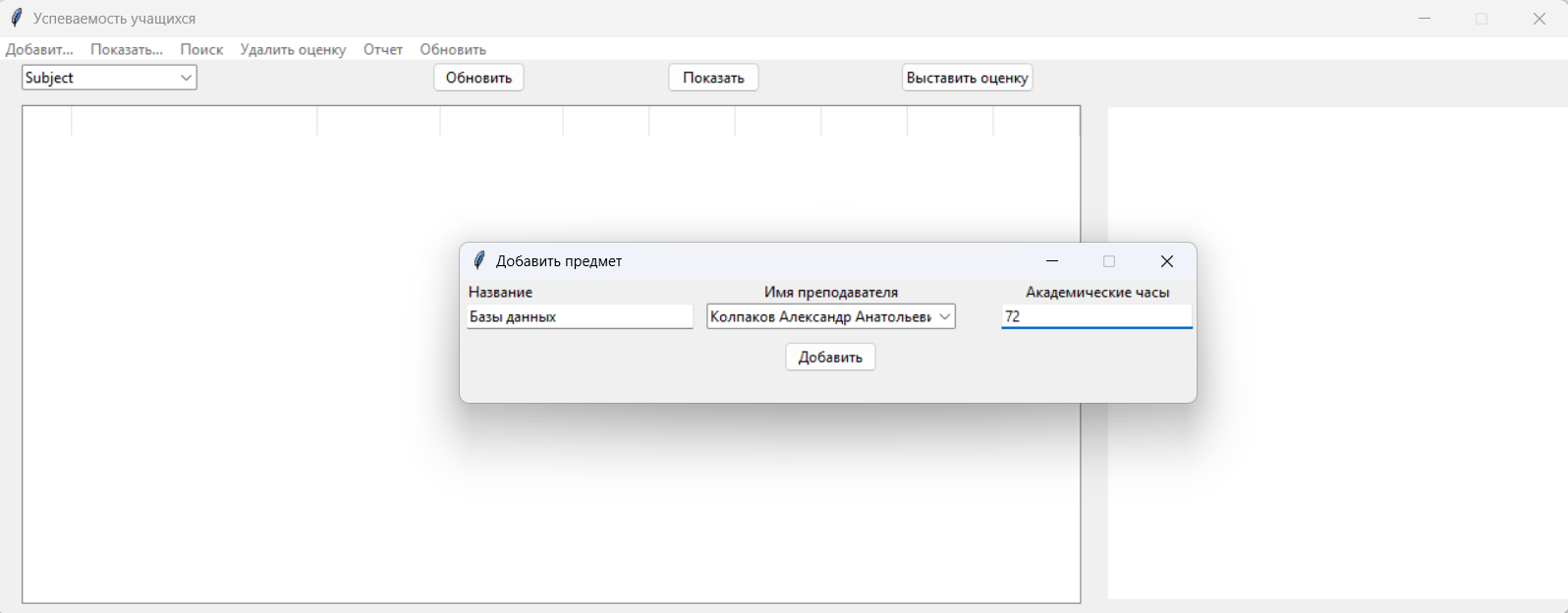


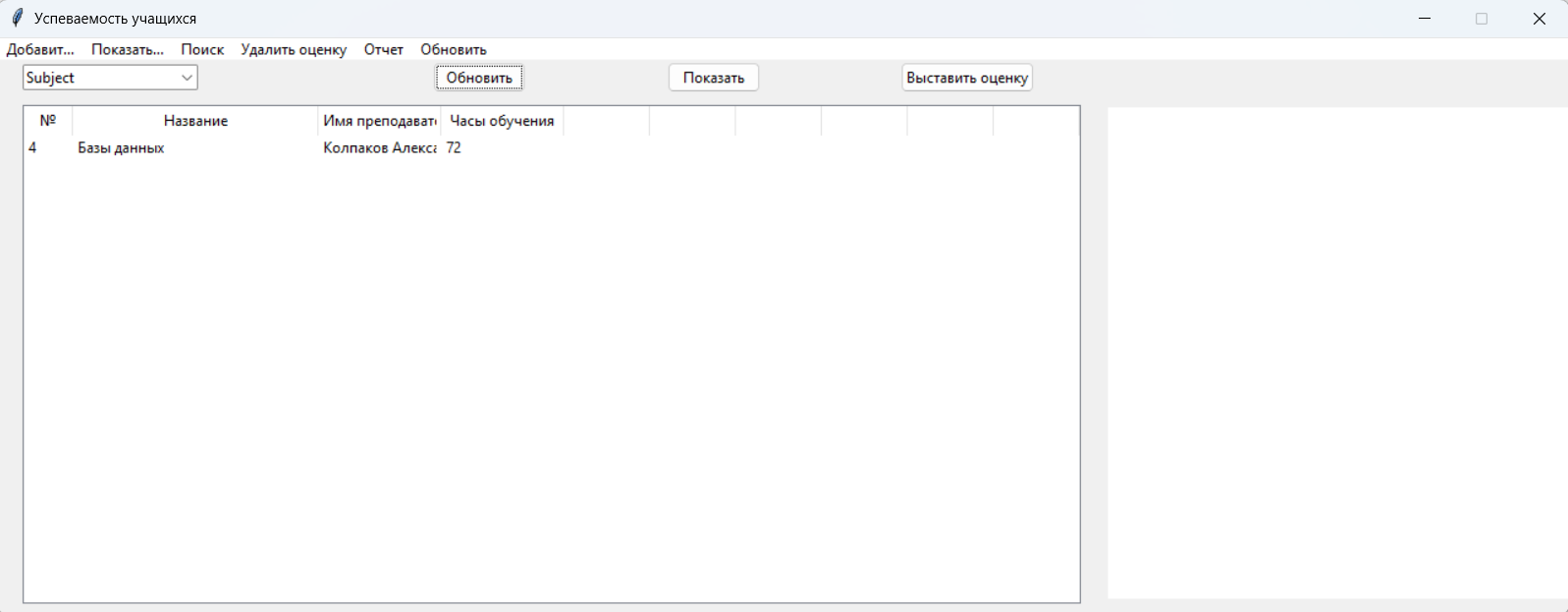
2.



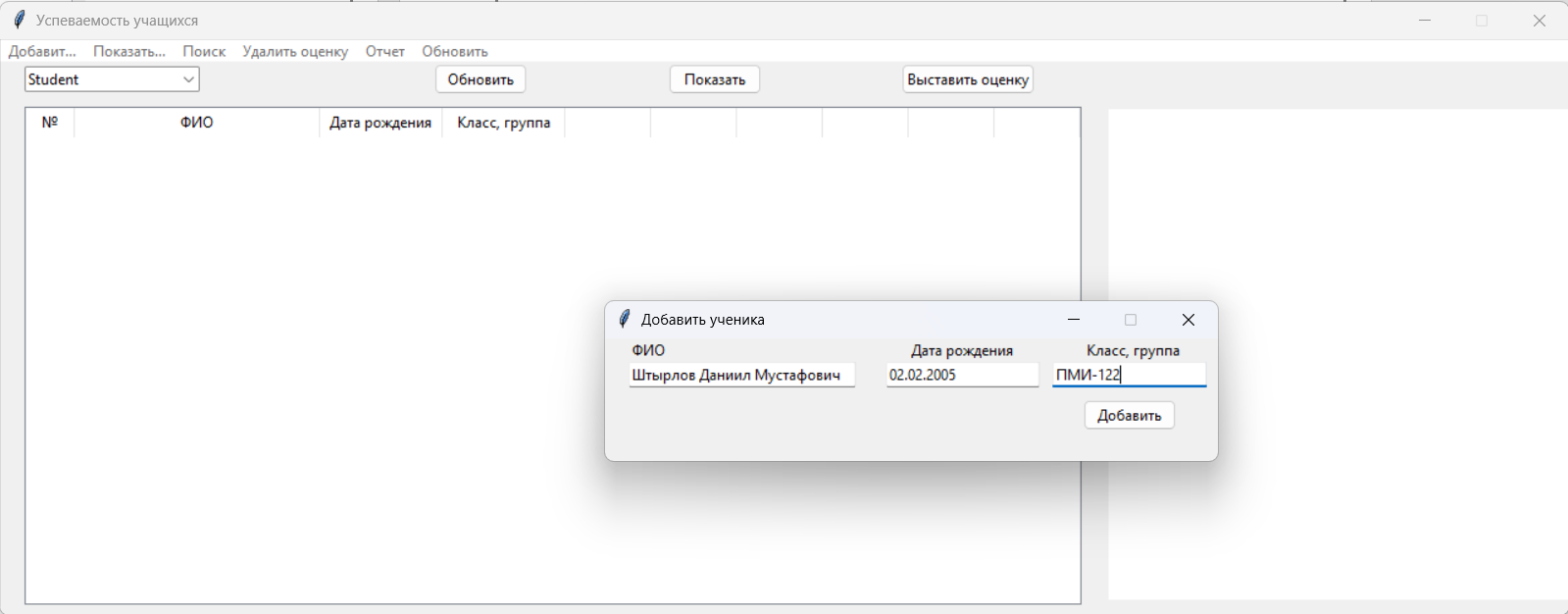


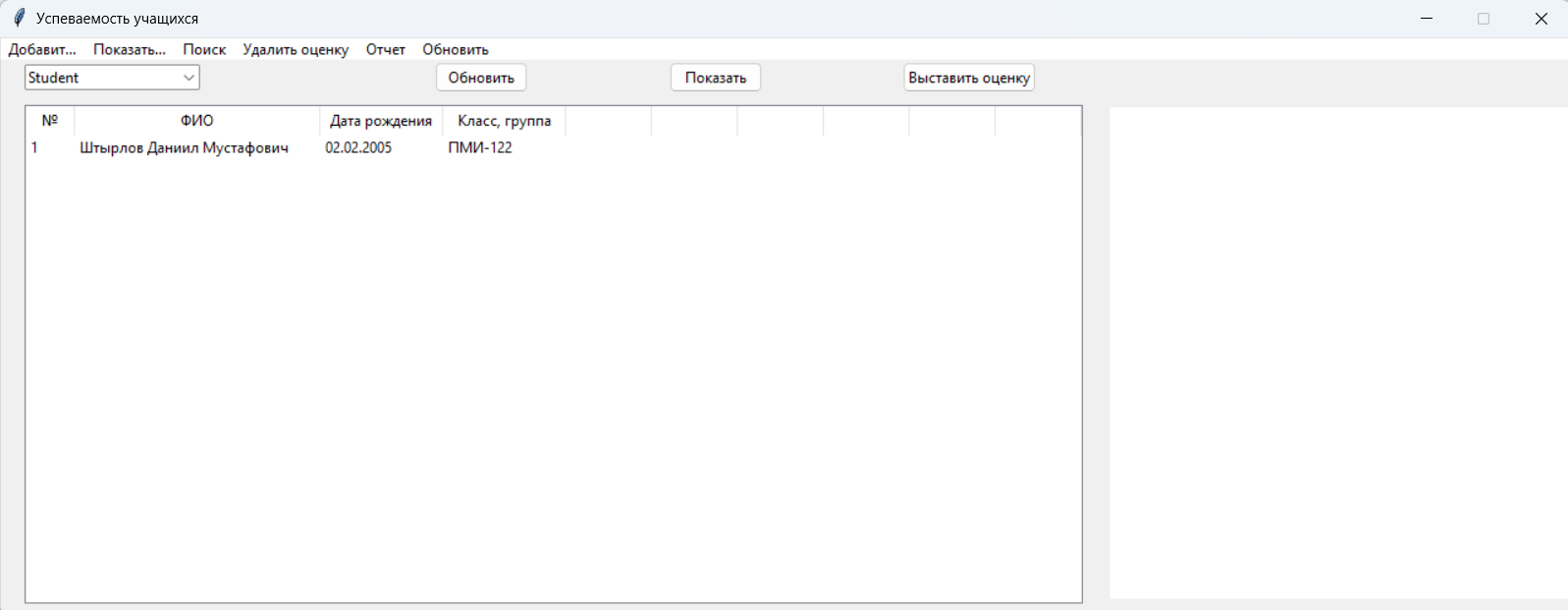
3.



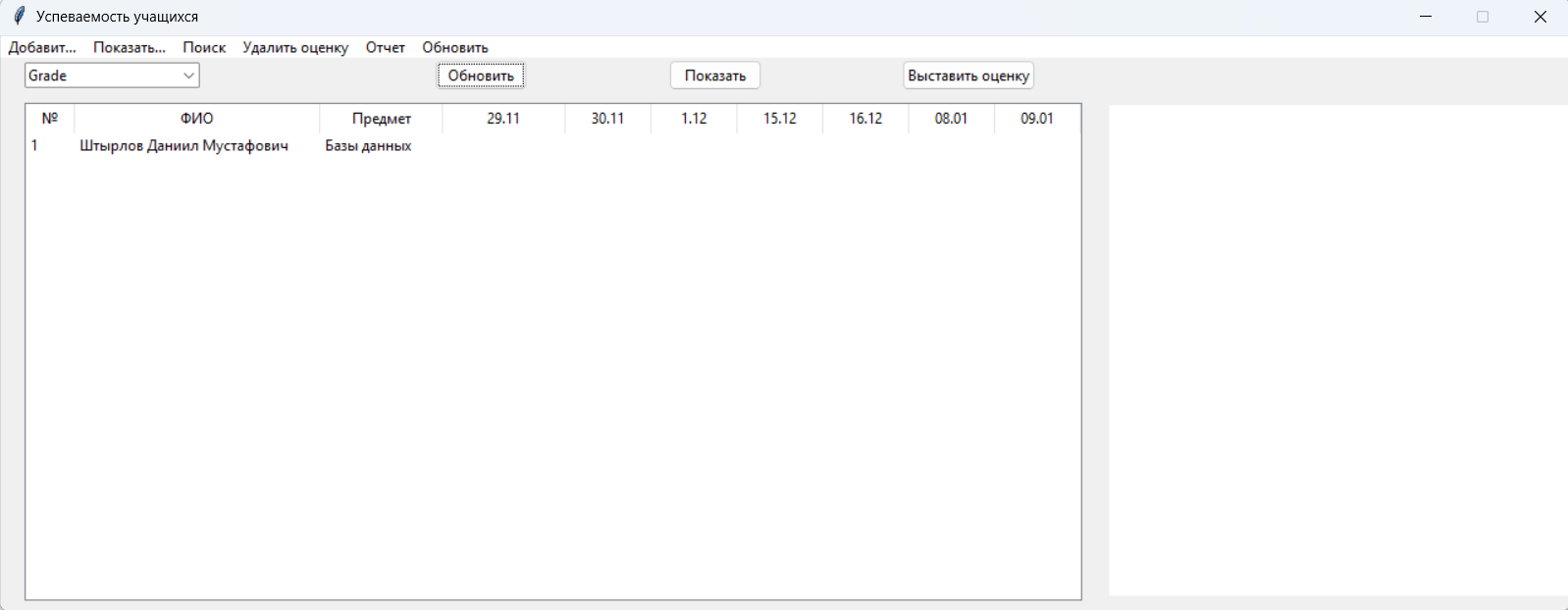


4.

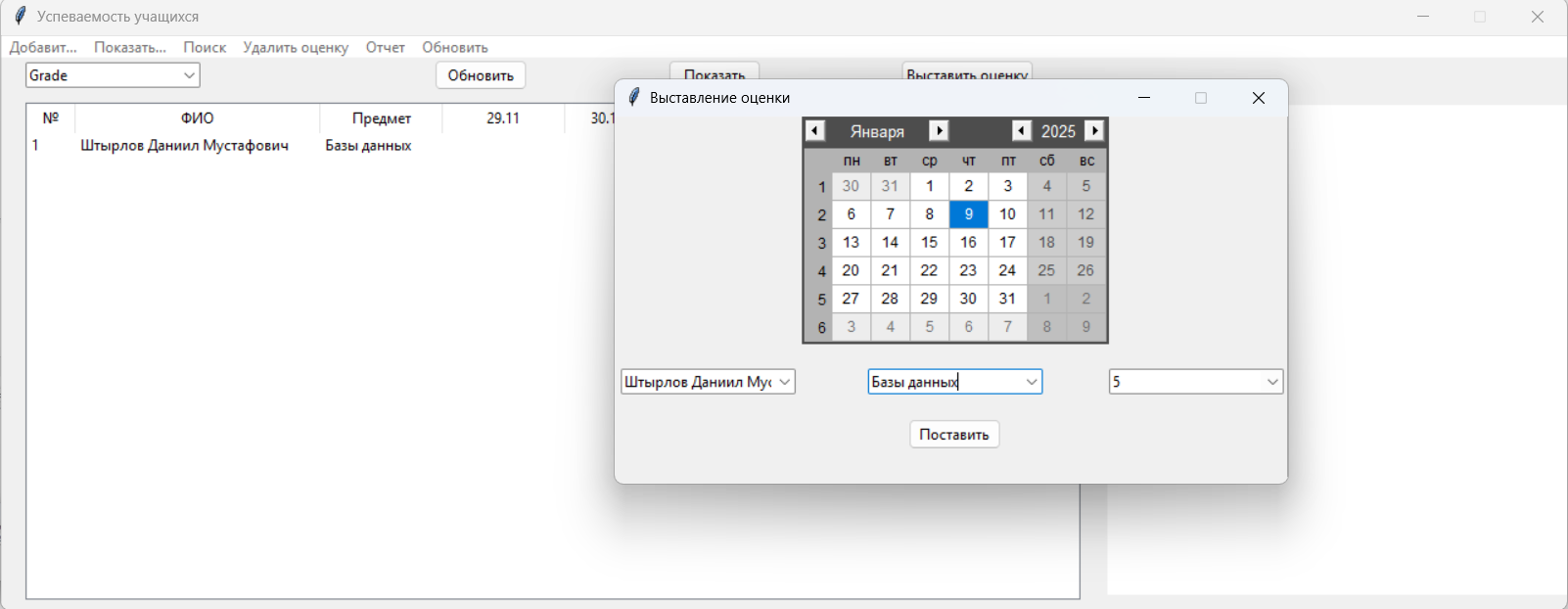


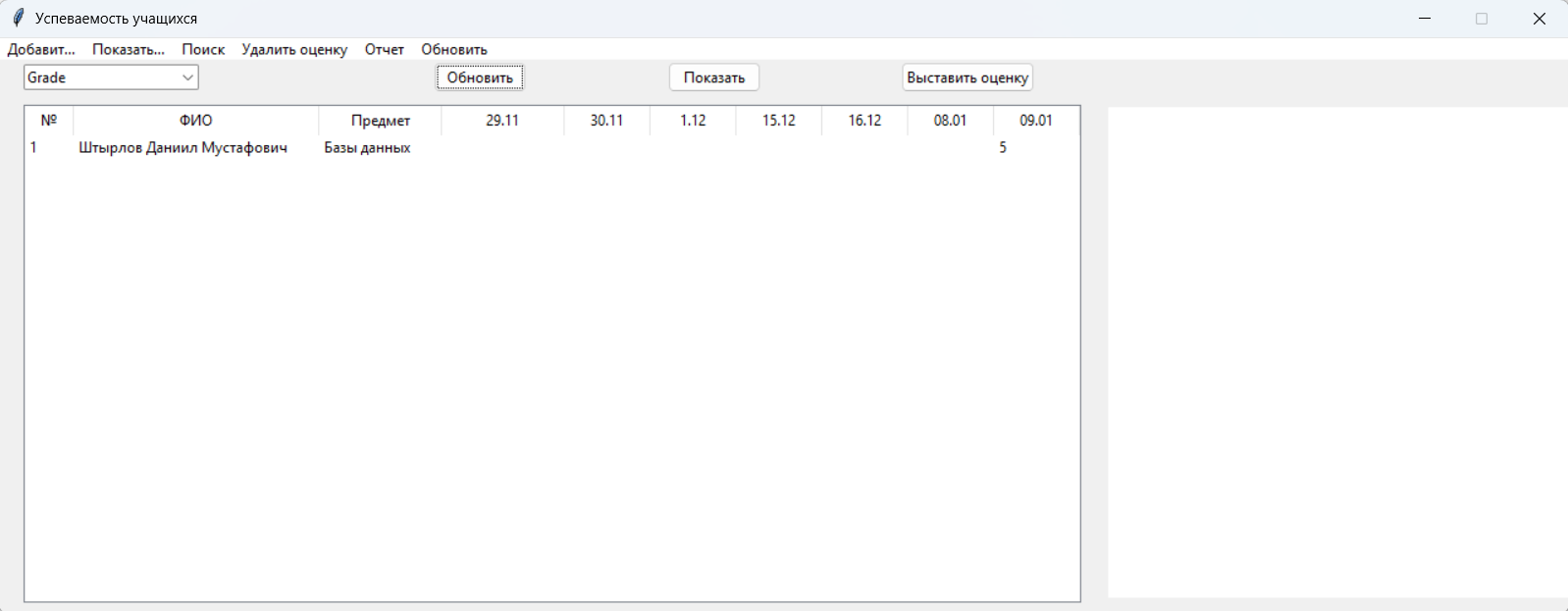


5.

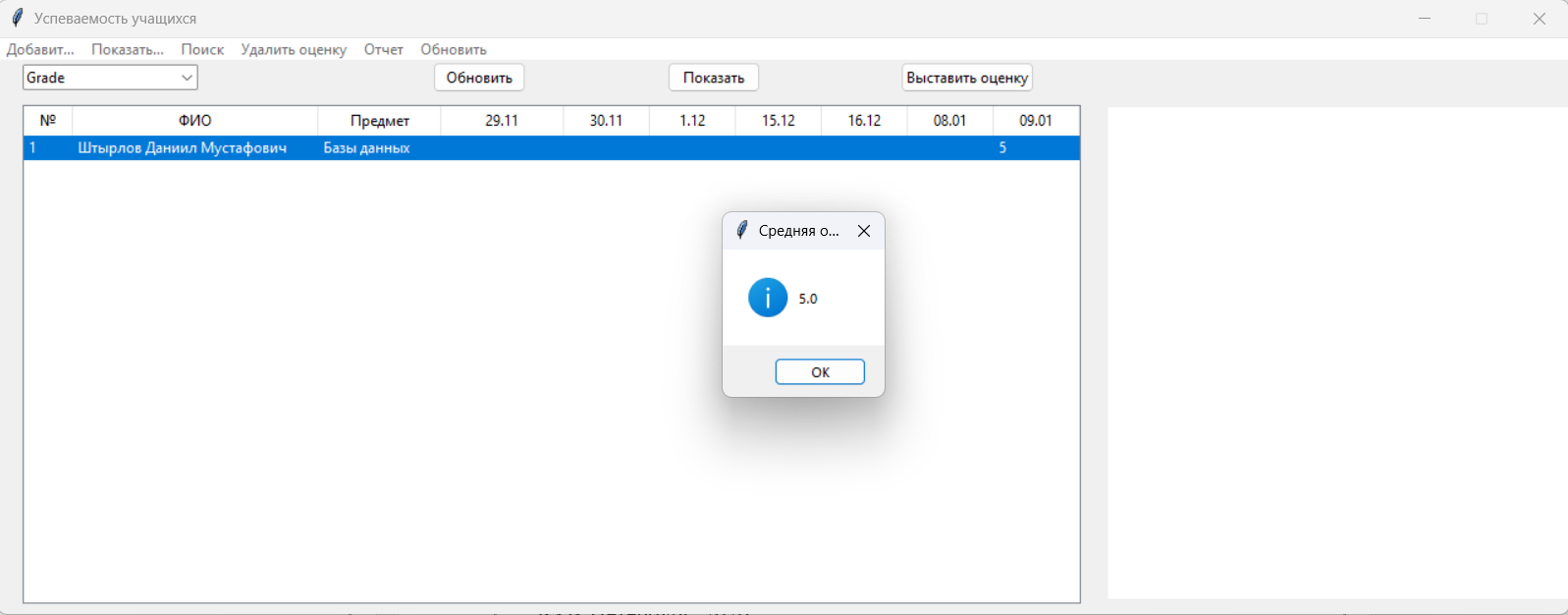


6.

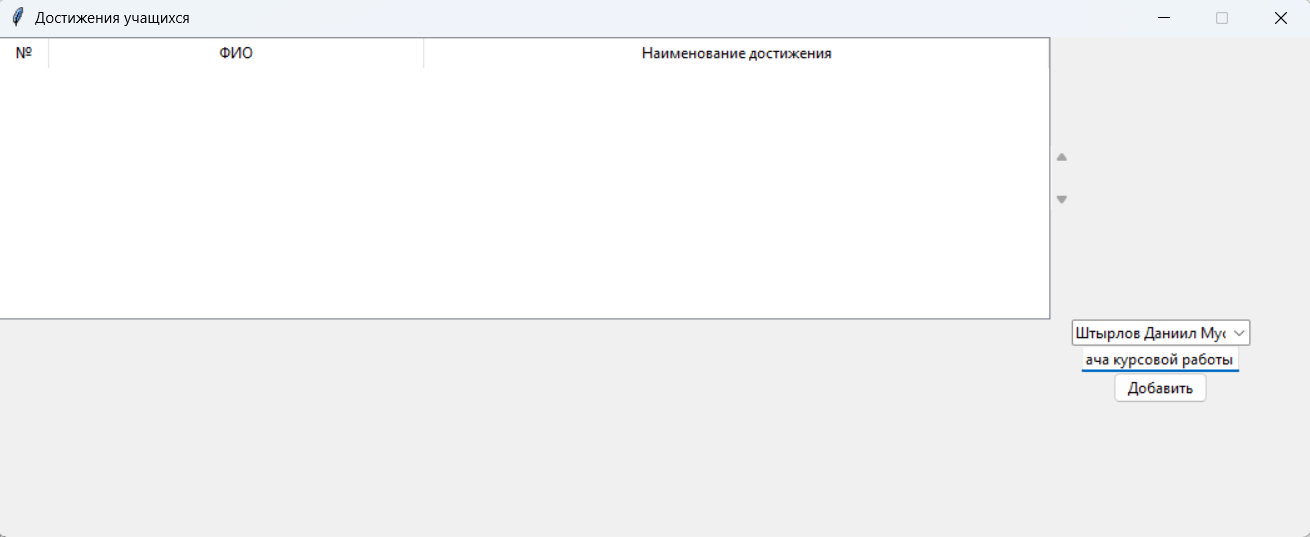


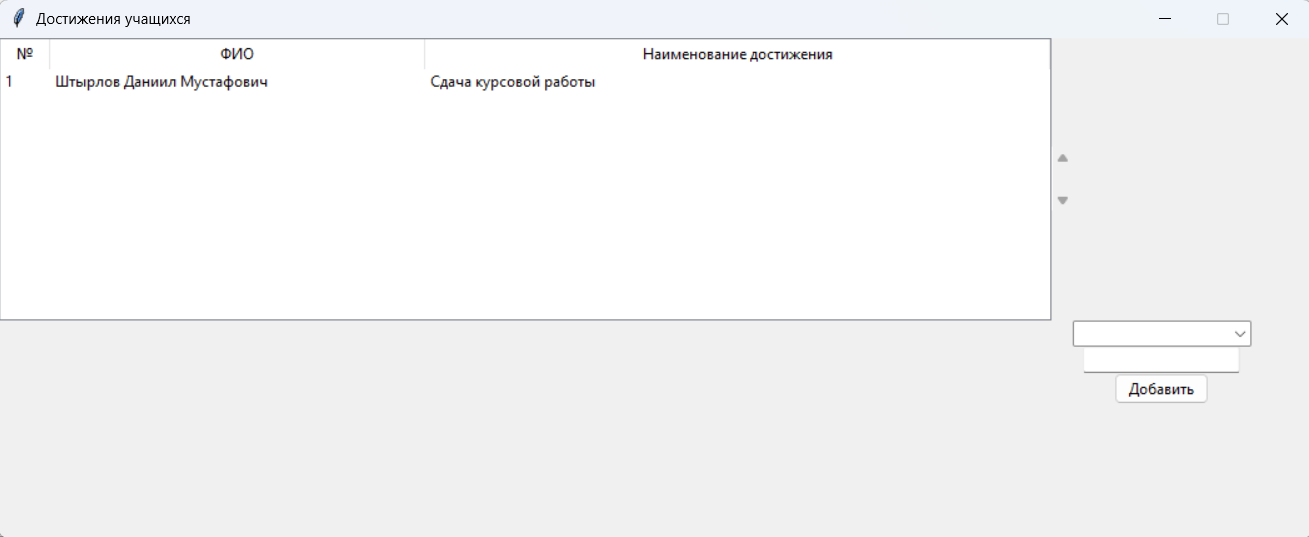


7.



8.





Как можно увидеть на изображениях все этапы из пользовательского руководства работают корректно и приложение полностью функционируют.

# Заключение

В ходе выполнения курсовой работы по теме Автоматизированной информационной системы учета успеваемости и достижений учащихся школы были достигнуты основные цели и задачи.

Система обеспечивает эффективный учет и анализ успеваемости, позволяет автоматизировать рутинные процессы, что значительно облегчает работу преподавателей и администраторов. В результате внедрения АИС становится возможным оперативно получать необходимые отчеты и данные, что способствует более глубокому пониманию уровня знаний учащихся и упрощает взаимодействие между всеми участниками образовательного процесса.

Кроме того, разработанная система демонстрирует гибкость и масштабируемость, что позволит адаптировать ее под конкретные нужды школы в будущем.

Таким образом, результаты работы подтверждают актуальность и необходимость внедрения АИС для повышения качества образовательного процесса и оптимизации управления.

# Список используемой литературы

1) *Тарасов, С. В. СУБД для программиста. Базы данных изнутри / С. В. Тарасов. - Москва : СОЛОН-Пресс, 2018. — 320 c. - ISBN 978-2-7466-7383-0.*

*- Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт].*

*- URL: https://www.iprbookshop.ru/90409.html (дата обращения: 23.08.2022). — Режим доступа: для авторизир. пользователей*

2) *Прокушев, Я. Е. Базы данных : учебник с практикумом / Я. Е. Прокушев. — 2-е изд. - Санкт-Петербург :*

*Интермедия, 2022. - 264 с. — ISBN 978-5-4383-0250-6. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. - URL: https://www.iprbookshop.ru/120171.html (дата обращения: 23.08.2022). — Режим доступа: для авторизир. пользователей*

3) *Маркин, А. В. СУБД «Ред База Данных». Основы SQL : учебное пособие / А. В. Маркин. - Москва : Ай Пи Ар Медиа, 2022. - 460 с. - ISBN 978-5-4497-1605-7. - Текст: электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/119617.html (дата обращения: 23.08.2022). — Режим доступа: для авторизир, пользователей*

# Приложение 1. Модели данных

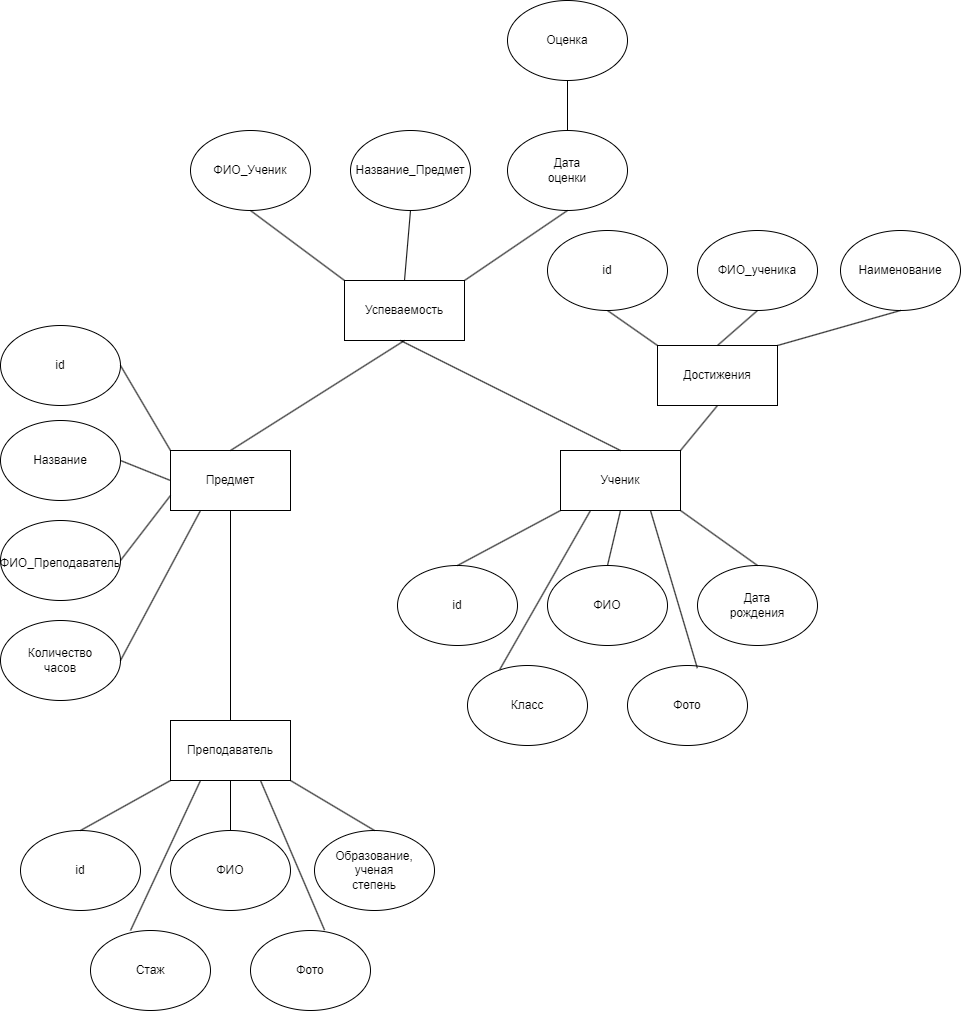


Рис. 1 – концептуальная модель данных

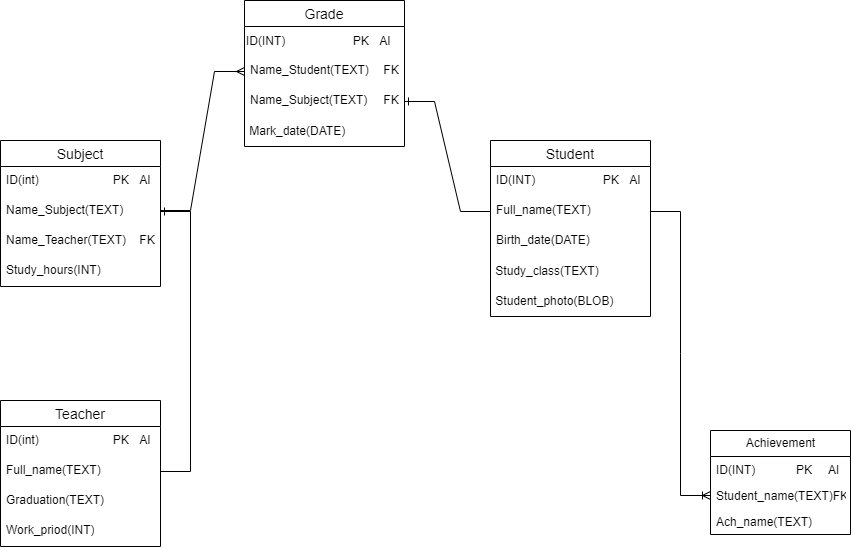


Рис. 2 - физическая модель данных

Приложение 2. Текст программы

Текст программы и все использованные в курсовой работе файлы прикреплены в репозиторий. Ссылка:

<https://github.com/DanyaShtyrlov/CourseWorkDB>

Приложение 3. Снимки окон программы

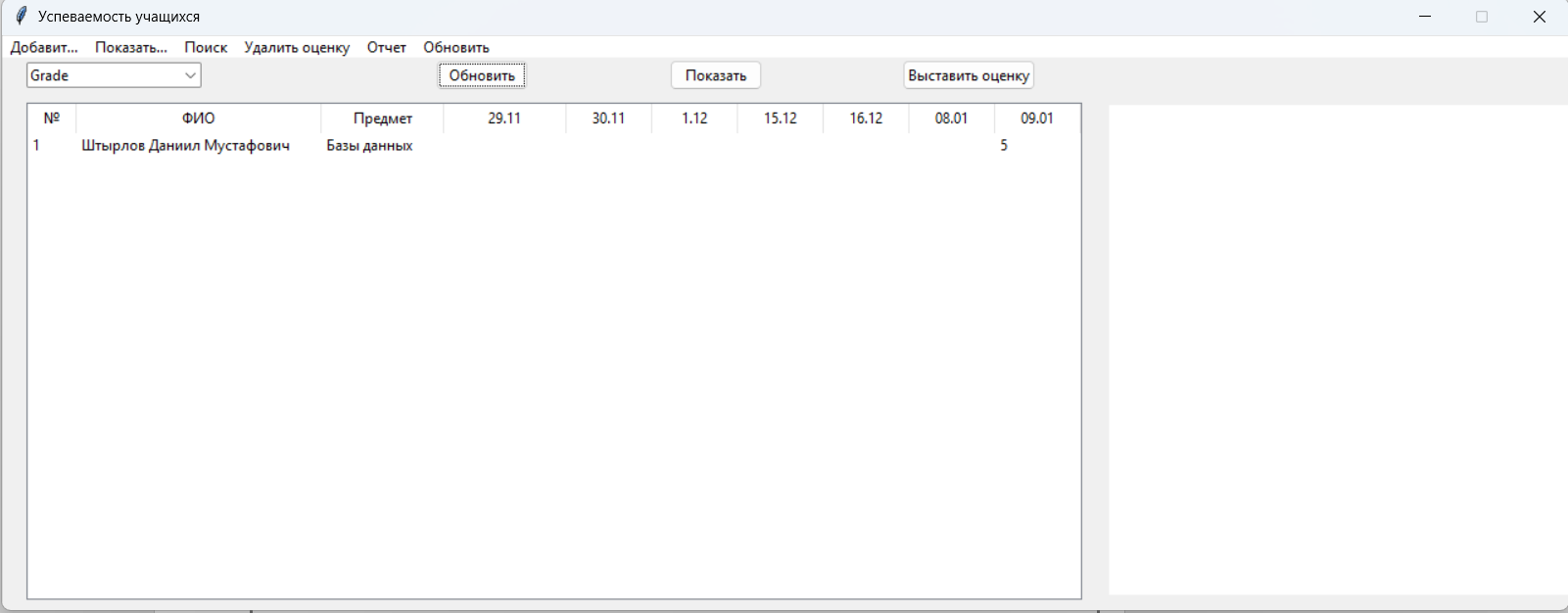


Рис 1 – основное окно

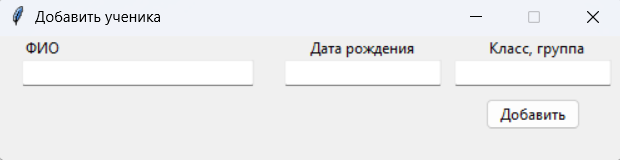


Рис. 2 – окно добавления ученика

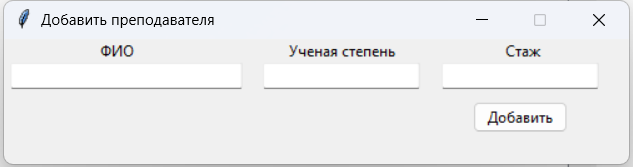


Рис. 3 – окно добавления преподавателя

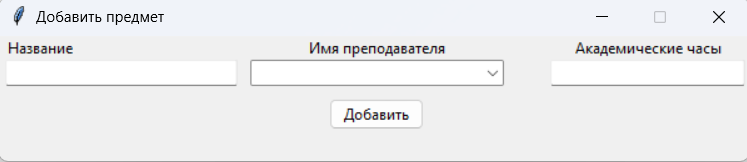


Рис. 4 – окно добавления предмета

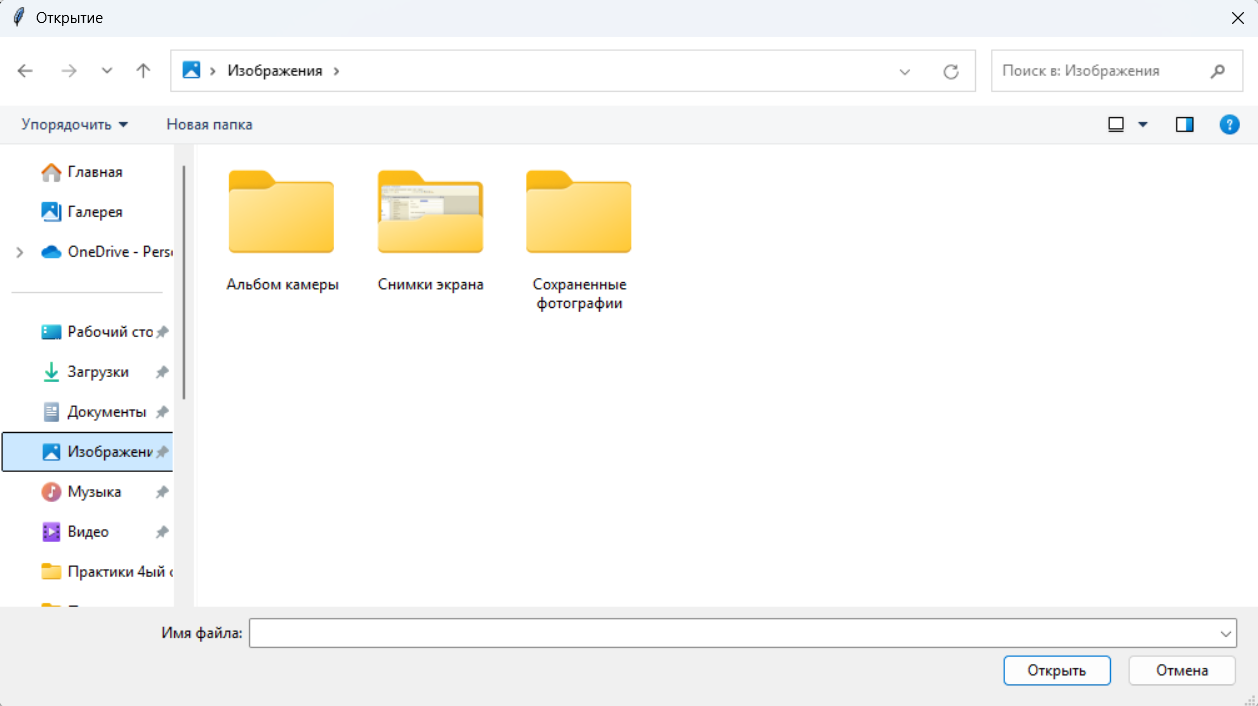


Рис. 5 – окно добавления фото

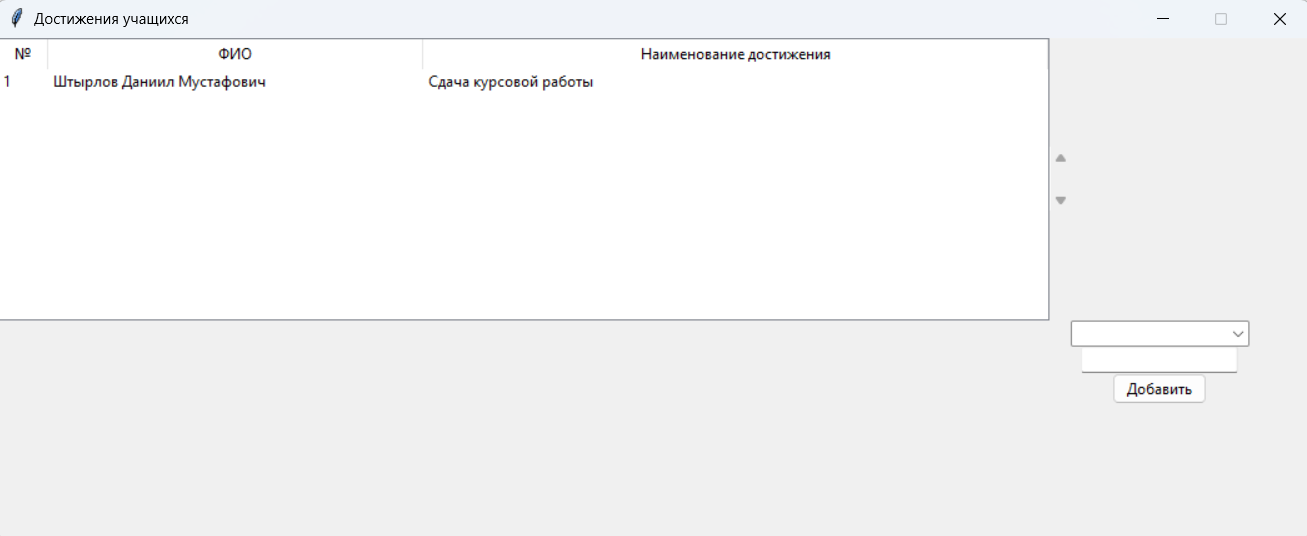


Рис. 6 – окно достижений учащихся

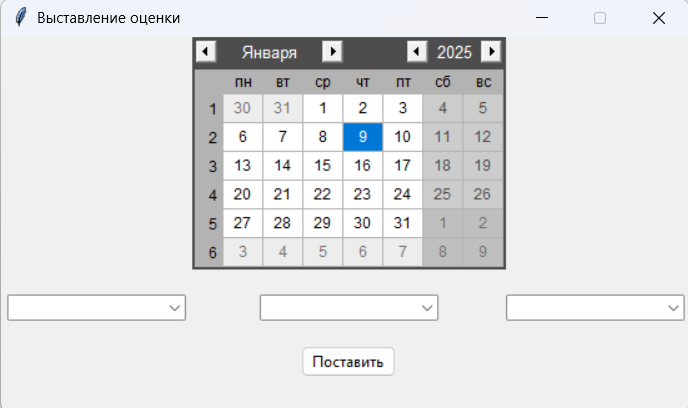


Рис. 7 – окно выставления оценки

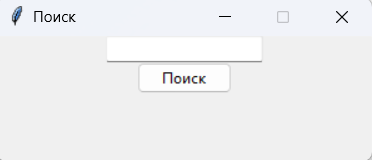


Рис. 8 – окно поиска