МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное
учреждение высшего образования
"Южно-Уральский государственный университет
(национальный исследовательский университет)"
Высшая школа электроники и компьютерных наук
Кафедра системного программирования

ОТЧЕТ

по учебной практике

бакалавра направления 02.03.02 "Фундаментальная информатика и информационные технологии"

Выполнил:_____
студент группы КЭ-201
Гордеев А.С.

Проверил:_____
Докт. физ.-мат. наук, доцент,
Проф. кафедры СП
Макаровских Т.А.
Дата:_____, Оценка:_____

Челябинск-2022

Министерство науки и высшего образования Российской Федерации
Южно-Уральский государственный университет
Кафедра системного программирования

УТВЕРЖДАЮ

Зав. кафедрой
системного программирования

_____ Л.Б. Соколинский

ЗАДАНИЕ
по учебной практике

1. Цель работы

Разработать GUI-приложение, работающее с входной информацией, вводимой пользователем с помощью управляемых элементов формы, либо из текстового файла.

2. Исходные данные к работе

1. База данных пословиц, поговорок, афоризмов, каламбуров и других словесных курьезов.
2. Хранение данных в файле в формате json.
3. Файл хранит такую информацию как: автор, тема и фраза.

3. Перечень подлежащих разработке вопросов

1. Определение структуры приложения (по модулям), структур данных, используемых для хранения основной пользовательской информации.
2. Дизайн оконного интерфейса, анализ структуры входных данных и их защита от некорректного ввода информации.
3. Разработка основного функционала приложения: основных форм и механизмов получения информации из их компонентов и их файлов; основного алгоритма функционирования приложения; тестирование приложения.
4. Подготовка руководства пользователя и документации для программиста.

4. Сроки

Дата выдачи задания: "27" июня 2022 г.
Срок сдачи законченной работы: "23" июля 2022 г.

Руководитель:

Докт. физ-мат наук, проф. кафедры СП	_____	Макаровских Т.А.
должность, ученая степень	подпись	ФИО руководителя

Задание принял к исполнению:

	_____	Гордеев А.С.
	подпись	ФИО студента

ОГЛАВЛЕНИЕ

1. ПОСТАНОВКА ЗАДАЧИ

Требуется написать справочник пословиц, поговорок, афоризмов, каламбуров, других словесных курьезов. Программа выполняет классификацию по авторам и источникам, поиск по темам и ключевым словам. Программа должна обеспечивать поиск по базе по заданным критериям, позволять редактировать и дополнять базу. Записи базы данных будут иметь следующие типы (табл.1).

Таблица 1. Переменные, используемые для хранения записей базы данных

Переменная	Тип переменной	Содержательный смысл
author	QString	Автор (источник)
theme	QString	Тема
phrase	QString	Фраза

Реализовать решение данной задачи можно с помощью такой известной структуры как префиксное дерево, или же бор. Каждый его элемент — вершина дерева, включающая в себя символ поиска строки, номер вершины, хэш-таблицу с номерами следующих вершин в виде: ключ-символ, значение-номер; булеву переменную, является ли вершина листом и указатель на данные. Само префиксное дерево будет состоять из символов (вершин) строки с названием источника (автора) и темы для их поиска по данным критериям, а каждый лист указатель на строку с фразой.

Листинг 1. Пример структуры вершины и класса, реализующего поиск в префиксном дереве

```
struct Node {
  QMap<QChar, int> nextNodes;
  bool isLeaf = false;
  void* data;
};

class PrefixSearcher {
 public:
  PrefixSearcher() : countNodes_(1), countStrings_(0) {}
  int getCountStrings() { return countStrings_; }
  void insert(QString& str, void* input_data);
```

```
  void erase(QString& str);
  QVector<void*> find(QString& prefix);

 private:
  QMap<int, Node> trie_;
  int countNodes_;
  int countStrings_;

  bool canGoNode(int vertice, QChar sym);
  void createNode(int vertice, QChar sym);
  int getNextVertice(int vertice, QChar sym);
  bool isLeaf(int vretice);
  void depthFirstSearch(QVector<void*>& result, int vertice);
};
```

Разрабатываемое приложение состоит из пяти оконных форм:

1. Главное окно программы.
2. Диалоговое окно добавление записи в базу.
3. Окно редактирования записи.
4. Окно с информацией о приложении.
5. Окно с информацией об авторе.

Каждой из разработанных оконных форм соответствует пара файлов (*.h и *.cpp), приведенные в таблице 2. Там же приведена информация о файле prefixsearcher.h, в котором содержится информация о классе, реализующем работу префиксного дерева.

Таблица 2. Модули создаваемого проекта

Имя файла	Описание информации, содержащейся в нем	Функциональное назначение	Файлы проекта, подключенные к текущему файлу посредством директивы #include
prefixsearcher.cpp	Класс реализации поиска в дереве	Работа с префиксном деревом	prefixsearcher.h
tabledata.cpp	Класс, реализующий базу данных таблицы	Хранение и обработка данных таблицы	tabledata.h
mainwindow.cpp	Класс основного рабочего окна приложения	Работа с основным окном приложения.	mainwindow.h

additemdialog.cpp	Класс окна добавления записи	Ввод данных пользователя и добавление в базу	additemdialog.h
edititemdialog.cpp	Класс окна редактирования записи	Редактирование данных в базе	edititemdialog.h
helpwindow.cpp	Класс окна с информацией о приложении	Вывод информации о приложении	helpwindow.h
aboutwindow.cpp	Класс окна с информацией об авторе	Вывод информации об авторе	aboutwindow.h

На рисунке 1 представлена схема взаимодействия классов и оконных форм в приложении.

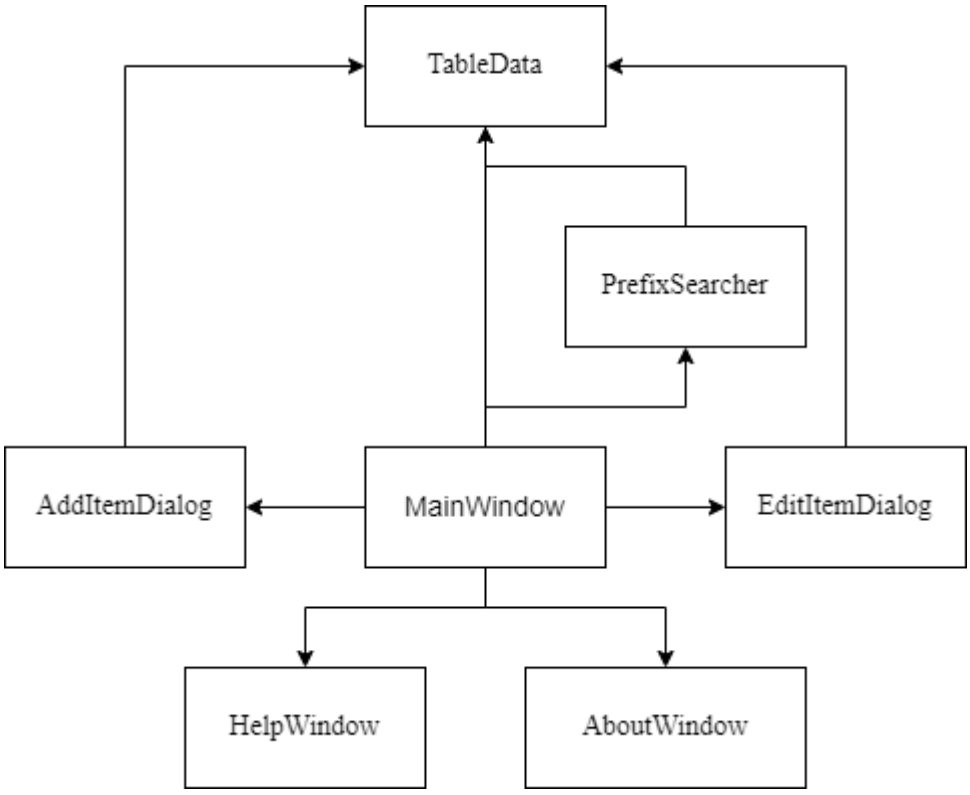

Рисунок 1. Блок-схема взаимодействия классов

2. ДИЗАЙН ОКОННОГО ИНТЕРФЕЙСА

В данном разделе приводится описание всех оконных форм, используемых для функционирования приложения:

- Основная форма для работы с таблицей.
- Форма добавление записи в таблицу.
- Форма редактирования записи таблицы.
- Форма информации о приложении.
- Форма информации об авторе.
- Вспомогательные окна.

При описании интерфейса приводится изображение соответствующей формы и приводится перечень помещенных на нее компонентов для ввода/вывода данных, оформления, и пр.

Для реализации работы с данными можно воспользоваться таблицей, реализованной в классе QTableView. На рисунке 2 представлено оформление основной формы приложения MainWindow класса QMainWindow.

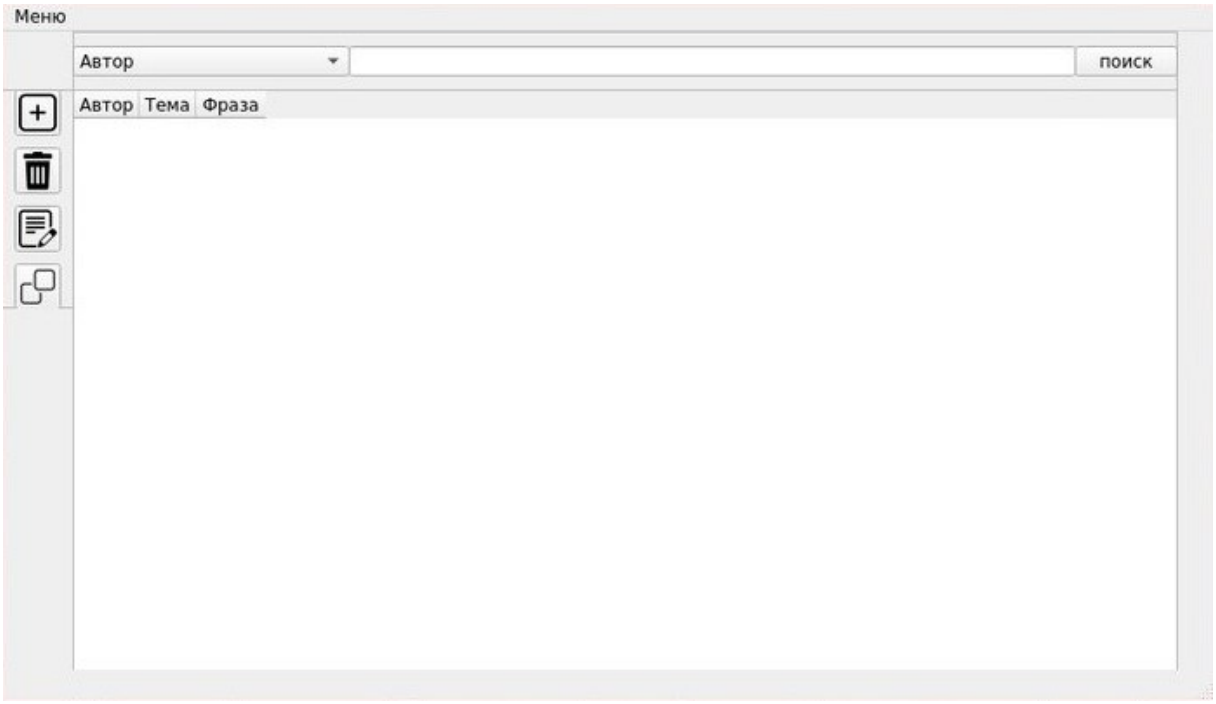

Рисунок 2. Оформление формы основного окна приложения MainWindow

Размещенные на форме компоненты и перечень методов и событий, которые необходимо реализовать приведен в таблице 3.

Таблица 3. Компоненты основного окна приложения

Имя компоненты	Тип	Ограничения для ввода информации	Реализованные события	Функциональное назначение
mainTableView	QTableView	Таблица с данными	Нет	Отображение таблицы из класса TableData
criterionComboBox	QComboBox	Выпадающее меню с пунктами: Автор, Тема	Нет	Выбор критерия для поиска в таблице
searchLineEdit	QLineEdit	Ввод данных в виде строки	Нет	Пользователь вводит строку для поиска значения из таблицы по заданному критерию
searchButton	QPushButton	Инициализация поиска с заданной строкой	onSearchButton	Пользователю выводится таблица с заданным критерием поиска
addButton	QPushButton	Вывод формы для добавления записи в таблицу	onAddButton	При нажатии таблицы открывается форма для заполнения
removeButton	QPushButton	Удаление записи с таблицы	onRemoveButton	При нажатии на кнопку осуществляется удаление выделенной записи таблицы
editButton	QPushButton	Вывод формы для редактирования записи	onEditButton	При нажатии на кнопку пользователь может редактировать выделенную запись таблицы
copyButton	QPushButton	Копирование записи таблицы	onCopyButton	При нажатии на кнопку выделенная запись копируется в буфер обмена
menuBar	QMenuBar	Выбор пункта из выпадающего меню	Нет	Пользователь может обратиться к пунктам меню

Добавление и редактирование записи из таблицы реализованы в похожих формах AddItemDialog и EditItemDialog соответственно и имеют класс QDialog. На рисунке 3 представлено оформление этих форм, а их компоненты в таблице 4. Главные отличия в них, это название кнопок “Добавить” и “Изменить”, и в том, что данные в форме EditItemDialog берутся из таблицы для редактирования.

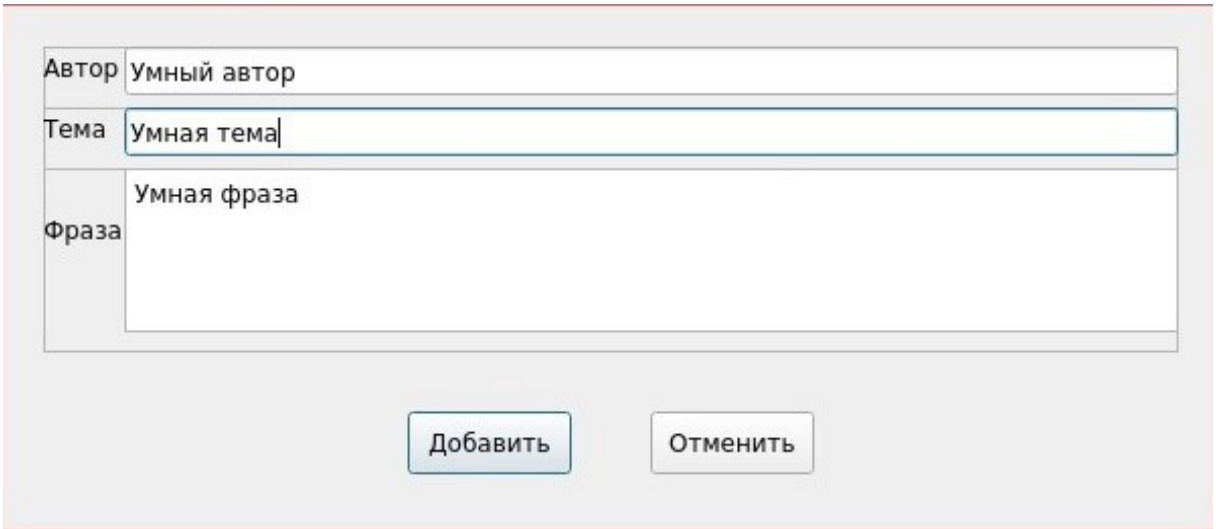

Рисунок 3. Оформление формы добавления записи AddItemDialog

Таблица 4. Компоненты диалоговых окон AddItemDialog и EditItemDialog

Имя компоненты	Тип	Ограничения для ввода информации	Реализованные события	Функциональное назначение
authorLabel	QLabel	Текстовая информация	Нет	Информация о вводимом поле “Автор”
authorLineEdit	QLineEdit	Ввод строки	Нет	Пользователь вводит имя автора или источника
themeLabel	QLabel	Текстовая информация	Нет	Информация о вводимом поле ‘Тема”
themeLineEdit	QLineEdit	Ввод строки	Нет	Пользователь вводит название темы фразы
phraseLabel	QLabel	Текстовая информация	Нет	Информация о вводимом поле “Фраза”
phraseTextEdit	QTextEdit	Ввод много-	Нет	Пользователь

		строчных данных		вводит фразу
acceptButton	QPushButton	Соглашение о вводимых данных	Нет	При нажатии на кнопку пользователь добавляет либо изменяет запись таблицы
cancelButton	QPushButton	Отмена изменений	Нет	При нажатии на кнопку пользователь отказывается от вводимых или измененных данных

Информационная форма о приложении и форма с информацией об авторе реализованы в простом QMessageBox. Информационная форма содержит пояснительную картинку с компонентой infoImage класса QImage, а форма об авторе компоненту textLabel класса QLabel.

Алгоритмы и программная реализация приведенных в этой главе событий и методов приведены в следующем разделе.