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.

1. **Исходные данные к работе**
2. База данных пословиц, поговорок, афоризмов, каламбуров и других словесных курьезов.
3. Хранение данных в файле в формате json.
4. Файл хранит такую информацию как: автор, тема и фраза.
5. **Перечень подлежащих разработке вопросов**
6. Определение структуры приложения (по модулям), структур данных, используемых для хранения основной пользовательской информации.
7. Дизайн оконного интерфейса, анализ структуры входных данных и их защита от некорректного ввода информации.
8. Разработка основного функционала приложения: основных форм и механизмов получения информации из их компонентов и их файлов; основного алгоритма функционирования приложения; тестирование приложения.
9. Подготовка руководства пользователя и документации для программиста.
10. **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 4](#_Toc107416469)

[2. ДИЗАЙН ОКОННОГО ИНТЕРФЕЙСА 7](#_Toc107416470)

[3. РАЗРАБОТКА ФУНКЦИОНАЛА ОСНОВНЫХ ФОРМ И МЕХАНИЗМОВ ПОЛУЧЕНИЯ ИНФОРМАЦИИ 9](#_Toc107416471)

[4. РАЗРАБОТКА ОСНОВНОГО МЕХАНИЗМА ФУНКЦИОНИРОВАНИЯ ПРИЛОЖЕНИЯ 11](#_Toc107416472)

[5. ТЕСТИРОВАНИЕ 12](#_Toc107416473)

[5.1. Автономное тестирование 12](#_Toc107416474)

[5.2. Комплексное тестирование 12](#_Toc107416475)

[6. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 14](#_Toc107416476)

[7. РУКОВОДСТВО РАЗРАБОТЧИКА 15](#_Toc107416477)

[ЗАКЛЮЧЕНИЕ 16](#_Toc107416478)

[ЛИТЕРАТУРА 17](#_Toc107416479)

# 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 информаци об авторе | aboutwindow.h |

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

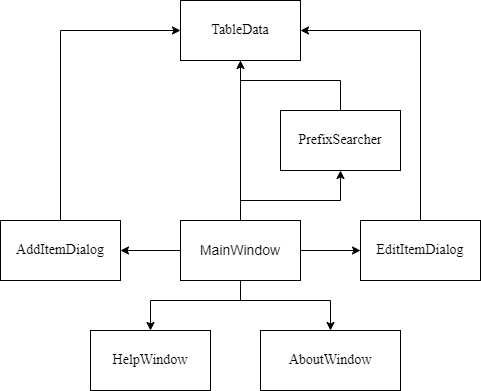


Рисунок 1. Блок-схема взаимодействия классов

# 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.

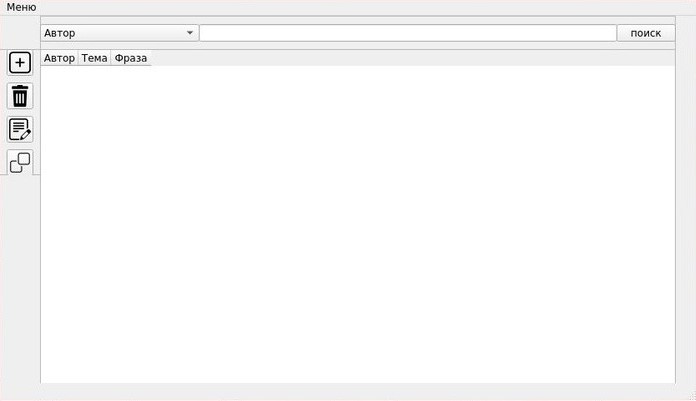


Рисунок 2. Оформление формы основного окна приложения MainWindow

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.

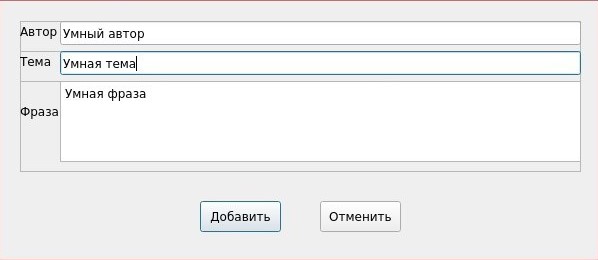


Рисунок 3. Оформление формы добавления записи AddItemDialog

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 многострочных данных | Нет | Пользователь 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.

# РАЗРАБОТКА ФУНКЦИОНАЛА ОСНОВНЫХ ФОРМ И МЕХАНИЗМОВ ПОЛУЧЕНИЯ ИНФОРМАЦИИ

В данном разделе приводится структурная схема, указывающая связь между вводимыми/выводимыми данными (на форме) и объектами, используемыми для их хранения. Если для хранения объекта используется тип, отличный от String^, в схеме необходимо отметить наличие операции приведения типов.

Для каждой формы пишется отдельный подраздел, в котором естественным языком описываются (возможно, с приведением программного кода) все события, указанные в табл.3. Слово «возможно» в предыдущем предложении не освобождает вас от необходимости приводить код совсем. Не приводится только код большого объема (в этом случае он ибо размещается в приложении, либо приводится краткий алгоритм функционирования), а также однотипный код (здесь приводится одна функция, про остальные указывается, что они реализованы аналогичным образом).

**Пример.** При выборе пользователем марки автомобиля из выпадающего списка, вызывается метод SelectedIndexChanged, который записывает соответствующее значение типа String^ во временную переменную <имя переменной> того же типа. Эта переменная используется в дальнейшем при обработке события Click при нажатии на кнопку «Поиск по марке» и передается в качестве параметра в метод Find разработанного пользовательского класса. Исходный код функции приведен в Листинге 1 (далее вставляете текст рассмотренной функции).

**Замечания**

При использовании диалогов для выбора входных/выходных файлов либо диалога печати, следует описать функции, отвечающие за получение информации посредством чтения/записи из/в файла.

Системные диалоги – это только элементы интерфейса, позволяющие получить корректное имя файла. Данные компоненты не отвечают за процесс чтения и записи файлов. Чтение и запись файлов реализуется стандартными средствами языка. Рекомендуется использовать потоки, поскольку они используют объектно-ориентированный подход. Операции чтения и записи данных рекомендуется реализовать в качестве методов пользовательского класса.

Для корректного функционирования программы рекомендуется устанавливать флаг Enabled=true только для тех компонент формы, для которых в текущий момент возможен корректный запуск разработанных методов.

# РАЗРАБОТКА ОСНОВНОГО МЕХАНИЗМА ФУНКЦИОНИРОВАНИЯ ПРИЛОЖЕНИЯ

В данном разделе описываются методы классов для работы со списками, в которых хранятся данные приложения:

* чтение;
* редактирование;
* вставка (вставка в хвост);
* поиск по разным ключам (перегруженная функция);
* удаление элемента.

Если вы решаете свою задачу, и перечень не соответствует реализуемому алгоритму, то вам необходимо выделить 5 подзадач, для которых вы опишете механизмы их решения и приведете код соответствующих функций.

Для каждой решаемой подзадачи необходимо записать алгоритм в словесной (либо в формульно-словесной) форме и привести листинг функции, используемой для реализации работы алгоритма. Если текст функции превышает 1 страницу, то код целесообразно привести в приложении со ссылкой на него в основном тексте отчета.

# ТЕСТИРОВАНИЕ

Данный раздел следует разбить на следующие пункты:

## Автономное тестирование

В данном разделе приводятся протоколы тестирования всех основных механизмов, приведенных в главе 4: чтение, редактирование, вставка (вставка в хвост), поиск по разным ключам, удаление элемента. Необходимо для каждого случая предусмотреть несколько принципиально различных наборов входных данных, не обязательно корректных. Количество наборов данных зависит от количества возвращаемых кодов ошибок и операторов возврата из функции.

Табл.4. Тестирование операции … (пример заполнения таблицы)

|  |  |  |
| --- | --- | --- |
| **Входные данные** | **Выходные данные** | **Верно ли?** |
| Перечисляются все параметры, передаваемые в функцию (корректные) | Приводится возвращаемое значение | **Да** |
| Приводится набор данных, приводящих к аварийному завершению функции | Возвращаемый код ошибки | **Да** |
| Приводится набор данных, позволяющий добраться до каждого размещенного в функции оператора return. | Приводится возвращаемое значение | **Да** |

## Комплексное тестирование

В данном разделе приводятся протоколы тестирования приложения. В таблице необходимо отразить следующие ситуации. Для каждой из них возможно указать несколько тестов.

1. Программа работает корректно для введенных данных.
2. Пользователь ввел некорректные данные (либо попытался ввести некорректные данные)
3. Пользователь завершил работу системного диалога и отказался от выбора имени файла.
4. Имя входного/выходного файла задано некорректно. Входной/выходной файл не открывается.
5. Невозможно загрузить файл помощи.
6. Невозможно открыть заставку приложения.
7. Прочие критические и аварийные ситуации.

Полученный результат описывается текстом, но допускается приведение скриншотов оконных форм.

Табл.5. Тестирование головного модуля (примеры заполнения таблицы).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | Описание ситуации | Входные данные | Выходные данные | Тест пройден? |
| 1. | Программа работает корректно | Источник входных данных | Описание полученного результата | Да |
| 2. | Невозможно открыть заставку приложения | Путь к файлу с заставкой | Описание полученного результата, текст выводимого сообщения | Да |
| 3. | Невозможно загрузить файл помощи | Путь к файлу помощи | текст выводимого сообщения об ошибке | Да |

# РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Данный раздел следует разбить на следующие пункты:

1. Запуск приложения, приветственные экраны, какие ресурсы необходимы для выполнения программы.
2. Различные способы ввода и вывода информации.
3. Описание всех возможностей программы.
4. Перечень ошибок и способов их устранения.

Следует отметить, что пользователь может быть не знаком с языками программирования и объектно-ориентированным подходом. Пользователь – это человек, который запускает исполняемый файл и работает с ним. От пользователя следует скрывать все детали реализации, поскольку эта информация для него не актуальна. Чтобы было легче написать данный раздел представьте, что вы рассказываете о функционировании своей программы потенциальному покупателю и показываете ему все возможности вашей разработки.

# РУКОВОДСТВО РАЗРАБОТЧИКА

Данный раздел отдельно не оформляется, поскольку финальная версия отчета и является руководством разработчика.

# ЗАКЛЮЧЕНИЕ

Данная работа посвящена разработке оконного приложения <вставить название либо описать основной функционал> с использованием языка C++. Для хранения данных использованы объекты класса <вставить название>, являющегося наследником класса List, приведенного в задании. При организации функционирования класса была создана перегруженная функция <вставить название>, используемая для поиска элемента списка с ключами разных типов, а также с различными наборами ключевых полей.

В ходе выполнения работы были пройдены все основные этапы разработки программного обеспечения: анализ, написание спецификации, проектирование, разработка алгоритмов, кодирование, тестирование и сопровождение. Проведено автономное тестирование основных методов разработанного класса и комплексное тестирование программы в целом. В результате тестирования ошибок не обнаружено. В данных пунктах описана проделанная в каждом случае работа и ее результаты.

Привести направления дальнейшей разработки. Что в разработке может быть расширено, улучшено, оптимизировано?

# ЛИТЕРАТУРА

1. Панюкова Т.А., Панюков А.В. Языки и методы программирования: Путеводитель по языку С++. Учебное пособие. – М.: Книжный дом «Либроком», 2013. – 216 с.
2. Макаровских Т.А., Панюков А.В. Языки и методы программирования: Создание простых GUI-приложений с помощью Visual C++: Учебное пособие. Изд. 2-е. – М.: «Ленанд», 2018. – 144 с.
3. Панюкова Т.А. Документирование программного обеспечения: В помощь техническому писателю: Учебное пособие. – М.: Книжный дом «Либроком», 2012. – 264 с.
4. Рязанова, Н. Ю. Программирование на языке С++ в среде Visual Studio CLR Windows Forms : учебное пособие / Н. Ю. Рязанова, К. Л. Тассов, М. В. Филиппов. — Москва : МГТУ им. Н.Э. Баумана, 2017. — 64 с. — ISBN 978-5-7038-4563-9. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/103509 (дата обращения: 28.05.2021). — Режим доступа: для авториз. пользователей.
5. Пахомов Б. И. C/C++ и MS Visual C++ 2012 для начинающих. — СПб.: БХВ-Петербург, 2013. – 512 с.   
   <https://karsu.uz/wp-content/uploads/2019/07/7.pdf>
6. Допускается и приветствуется добавление своих источников информации.