|  |  |  |
| --- | --- | --- |
|  |  | УТВЕРЖДАЮ |
|  |  | Заведующий лаб. ВКС |
|  |  | \_\_\_\_\_\_\_\_\_\_\_\_\_ Фионов А.Н. |
|  |  | “\_\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_2013 |

**Разработка программного обеспечения**

**для тестирования лифтовых блоков**

**Пояснительная записка**

**Лист утверждения**

**А.В.00001-01 81 01**

Инв. № подл.

Подпись и дата

Взам. инв. №

Инв. № дубл.

Подпись и дата

**643.01180010.00001-01 81 02-лу**

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  | Ответственный исполнитель |
|  |  | Инженер лаб. ВКС |
|  |  | \_\_\_\_\_\_\_\_\_\_Медведева Ю.С. |
|  |  | “\_\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_2013 |

**УТВЕРЖДЕНО**

**643.01180010.00001-01 81 02**

**Разработка программного обеспечения**

**для тестирования лифтовых блоков**

**Инв. № подл.**

**Подпись и дата**

**Взам. инв. №**

**Инв. № дубл.**

**Подпись и дата**

**Пояснительная записка**

**643.01180010.00001-01 81 02**

# Содержание

[Содержание 2](#_Toc195534499)

[1. Описание шаблона для хранения информации и общей диаграммы модели данных 4](#_Toc195534500)

[2. Описание структуры данных 5](#_Toc195534501)

[3. Описание процесса работы с программой 7](#_Toc195534502)

[4. Источники, использованные при разработке 9](#_Toc195534503)

[Лист регистрации изменений 11](#_Toc195534504)

# Описание шаблона для хранения информации и общей диаграммы модели данных

Главной структурой каждого json-файла является проект. Эти проекты представляют из себя составной набор данных, каждый из которых обязательно включает в себя еще несколько вложенных структур, а именно:

* Структура зон пакетов – данная структура нужна для использования в визуальном интерфейсе программы. Она отвечает за цветовое выделение полей.
* Структура станций – данная структура отвечает за настройку каждой станции. Она включает в себя несколько структур: Структура виртуального порта – данная структура нужна для настроек аппаратной части порта. Эти данные передаются через COM-порт устройству, которое эмулирует лифтовой блок.
* Структура шаблона пакетов – набор данных, предназначенный для хранения информации о том, сколько места занимает та или иная зона в пределах одного пакета.
* Структура CRC – данная структура отвечает за информации о подсчете контрольной суммы для проверки целостности пакета.
* Структура пакетов данных – данная структура хранит информацию о пакетах данных для каждой станции.
* Структура тестов – в данной структуре пакеты собираются в единый тест. Ключевым полем здесь является поле действие, которое отвечает за то, необходим ли принять пакет или от эмулятора, или же отправить данные.

Стоит отметить, что практически все структуры могут храниться во множественных экземплярах в пределах одного json-файла. Поэтому было принято решение организовать программу таким образом, чтобы каждая структура представляла из себя отдельный класс со своими специфическими полями и уникальным идентификатором, который задается в конструкторе класса при его создании.

Над каждым классом модели должен стоять управляющий класс, который хранит в себе массив из подчиненных классов.

Ниже представлена диаграмма включений и отношений модели данных, в которой отражены все представленные структуры



Рисунок 1, Диаграмма включений и отношений модели данных

# Описание структуры данных

Для качественной реализации проекта было решено использовать схему разделения данных MVC(Model-View-Controller). Как видно из названия, данная схема состоит из трех компонентов а именно:

* Модель(model) – компонент, отвечающий за данные и определяющий структуру приложения. Это основные элементы программы, в которых хранятся данные.
* Представление(View) – компонент, отвечающий за взаимодействие с пользователем. К нему относятся классы, которые отвечают за перемещение данных от визуального интерфейса к классам-контроллерам.
* Контроллер(Controller) – компонент, отвечающий за связь между моделью и представлением. В них происходит обработка данных, которые, впоследствии сохраняются в моделях. Именно эти классы хранят в себе массивы с классами-моделями и отвечают за их добавление, удаление и изменение. Примером может послужить класс SectionManager, в котором хранится массив SectionModel.

Таким образом, классы были разделены на три компонента. Все, у которых в названии есть Model(SectionModel, PackageTemplateModel и д.р.) были отнесены к компоненту Model. Можно сказать, что данные классы являются единицами измерения соответствующих областей программы.

Все классы с суффиксом Manager(SectionManager, MainManager) были отнесены к компоненту Controller. Они служат связующим звеном между классами, относящимися к компоненту View и Model и занимаются управлением и хранением классов элемента Model. Через них осуществляется доступ к той или иной модели данных, происходит добавление, изменение или удаление той или иной модели.

Компоненты элемента View не отображен на данной схеме. Они работают в связке с QML-файлами и нужны для перемещения данных от контроллеров к визуальному интерфейсу. В них создаются и хранятся модели, которые передают данные во все списки графического интерфейса.

Также в реализации проекта было принято решение создать несколько классов-контроллеров, которые должны быть представленные в единственном экземпляре. Это классы PackageZoneManager, StationManager, PackageManager, TestManager и MainManager. Данные классы отвечают за управление целыми “секциями” в рамках одного проекта, поэтому создавать несколько таких контроллеров было бы просто неэффективно, потому что пришлось бы тратить много времени на поиск нужных данных.

Для того, чтобы гарантировать их уникальность, данные элементы были разработаны как Singleton классы, то есть доступ к ним можно получить только через один вход.

# Описание процесса работы с программой

При запуске программы у пользователя есть несколько доступных действий. Он может открыть уже существующий проект или создать новый. Во втором случае будет открыто дополнительное окно, в котором пользователь должен указать имя для нового проекта. При отсутствии проекта, остальные окна будут пустыми, а при выборе проекта, у которого отсутствует json-файл будет выведено сообщение об ошибке.

Во втором окне пользователь способен свободно добавлять новые или удалять уже существующие зоны пакетов.

В третьем окне пользователь может выполнить несколько действий:

Задать настройки для COM-порта – при раскрытии соответствующего порта становится доступным список параметров для редактирования.

Отредактировать или создать новый вид пакте – при создании нового пакета, открывается окно, в котором указывается размер зоны и ее название. После ввода данных размер будет округлен до ближайшего большего числа, кратного шестнадцати. Далее пользователь может добавить в таблицу зоны, созданные на предыдущем этапе. Для этого он должен нажать на пустую зону, и, после нажатия на соответствующую кнопку, появится окно, в котором необходимо указать позицию, с которой начинается зона, ее размер и выбрать нужную из списка доступных. Для редактирования уже существующих зон необходимо нажать на закрашенную область и, после нажатия на соответствующую кнопку ввести новые данные в всплывающее окно. Для удаление необходимо также выбрать закрашенную зону и, после нажатия на кнопку – подтвердить удаление, после чего вся зона сотрется.

Задать комментарий - универсальное поле, которое доступно практически во всех окнах. Оно нужно для оставления каких-либо заметок пользователем, и работает везде одинаково. В текстовое поле вводится комментарий, после чего он сохраняется в C++ класс. Поэтому в дальнейшем это поле не будет рассматриваться.

В четвертом окне пользователь заполняет или изменяет созданные ранее зоны в рамках одного пакета. При открытии окна ему доступен только просмотр уже созданных ранее пакетов. При нажатии на пакет внутри лифта, название станции и пакета переносятся в центральную часть окна, и происходит загрузка списка доступных зон пакета.

Для разблокировки функции редактирования существующих зон необходимо нажать на соответствующую кнопку. Все данные, которые уже были в зонах пакета, или которые пользователь изменил, переносятся во вкладку “Обзор всего пакета”, где распределяются по соответственным зонам, на основании данных из предыдущих окон. В этой вкладке уже нельзя изменить расположение зон, только их значение в рамках одного пакета.

Для быстрого ввода полных данных о пакетах предусмотрена отдельная вкладка, которая открывается при нажатии на кнопку “Вставить пакет”. При ее нажатии, открывается окно, в которое пользователь должен ввести данные в шестнадцатеричной системе счисления. Размер вводимой строки должен быть равен суммарному размеру всех зон пакета.

В пятом окне пользователь может настроить последовательность пакетов в рамках одного теста программы. При выборе нужного теста, становится доступным меню, где можно добавить новый пакет в общую очередь, которая будет использоваться при работе с эмулятором, отредактировать в существующей пакет, который будет использоваться и действие, которое будет выполнять программа с пакетом: отправлять эмулятору или ожидать это сообщение в ответе.

В шестом окне пользователь производит последнюю настройку перед работой с эмулятором. Вначале он выбирает COM-порт к которому подключено устройство и станцию которое оно эмулирует. Далее происходит пробное подключение к устройству и, в случае успеха, становится доступна следующая часть окна.

В ней пользователь выбирает тесты, которые нужно проверить, способ их передачи и время между отправкой. Также после успешного подключения в панели справа отображаются все доступные для данного лифта тесты. Если пользователь неправильно установил значения виртуального порта, то при нажатии на кнопку “Настройки порта” откроется вкладка сов семи настройками порта из третьего окна.

Пользователь может выбрать запуск одного теста, запуск всех тестов или запуск теста в цикле. Возможны комбинации методов. При запуске одного теста время между отправкой автоматически исправляется на нуль. Кроме того, можно настроить и способ передачи: синхронный или асинхронный.

После установки всех настроек, происходит обмен сообщениями с эмулятором, после чего тесты изменяют свои цвета, для отображения результатов. Также для дополнительной проверки пользователь может зайти во вкладку, которая доступна при нажатии на кнопку “Посмотреть историю обмена сообщениями”. Там более подробно расписана информация о прошедшем сеансе тестирования.

Для каждого окна предусмотрено несколько систем оповещений от случайных нажатий или ошибок при вводе. Как уже было сказано ранее практически везде недоступно изначальное редактирование готовых данных – для этого необходимо нажать соответствующую кнопку. Также для каждого окна при удалении предусмотрено всплывающее окно дя подтверждения, для предотвращения случайной потери данных. Также было реализовано специальное диалоговое окно с текстовым полем, в которое передаются сообщения об ошибках. В случае ошибки, действие не выполняется, а это поел становится видимым.

Для новых пользователей предусмотрено специальное меню “Помощь”, в котором расписан основной функционал приложения. Оно реализовано как несколько сменяющих друг друга окон.

# Источники, использованные при разработке

1. **Шлее Макс Qt 5.10. Профессиональное программирование на С++** М.:БХВ, 2018. 1072с
2. **Qt Core** [Электронный ресурс]. Режим доступа: https://doc.qt.io/qt-6/qtcore-index.html. – Загл. с экрана. – яз. англ
3. **Qt-test overview** [Электронный ресурс]. Режим доступа: https://doc.qt.io/qt-6/qtest-overview.html. – Загл. с экрана. – яз. англ.
4. **Qt Gui** [Электронный ресурс]. Режим доступа: https://doc.qt.io/qt-6/qtgui-index.html. – Загл. с экрана. – яз. англ
5. **Qt Qml** [Электронный ресурс]. Режим доступа: https://doc.qt.io/qt-6/qtqml-index.html. – Загл. с экрана. – яз. англ.
6. **Singleton** (Одиночка) или статический класс? [Электронный ресурс]. Режим доступа: https://habr.com/ru/articles/103681.– Загл. с экрана. – яз. рус.
7. **Что такое MVC: рассказываем простыми словами** [Электронный ресурс]. Режим доступа: https://ru.hexlet.io/blog/posts/chto-takoe-mvc-rasskazyvaem-prostymi-slovami.– Загл. с экрана. – яз. рус.
8. **Введение в CMake** [Электронный ресурс]. Режим доступа: https://habr.com/ru/articles/155467.– Загл. с экрана. – яз. рус.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Лист регистрации изменений | | | | | | | | | |
| Номера листов (страниц) | | | | | Всего  листов  (страниц)  в докум | №  документа | Входящий  № сопрово  дительного  документа  и дата | Подп. | Дата |
| Изм | изменен  ных | заме  ненных | новых | анулиро  ванных |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |