Министерство науки и высшего образования Российской Федерации

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)

Институт прикладной математики и компьютерных наук

Предмет: Технология Разработки Программного Обеспечения

**Отчет по лабораторной работе №3**

Выполнила студентка группы 931920 Ткачук Полина Романовна

Студентка группы № 931920

П. Р. Ткачук

Томск – 2022

**1.Задача.**

Задача

Необходимо разработать приложение печати графиков в pdf формате. Графики различаются типом графика, стилем графика и данными.

При разработке архитектуры предусмотреть масштабирование  
системы  
1. Добавление новых графиков (Графики отличаются видом, данными)  
2. Изменение визуального стиля графиков (цветной, черно-белый)

Общие требования к GUI:

1. Загружаем данные, путем выбора нужного файла. Данные в ПО не отображаем, отображаем только график, построенный относительно считанных данных.
2. При печати в pdf выбираем место сохранения графика.

**2.Решение.**

Решение

Решение поставленной задачи разбивается на несколько пунктов.

1.Необходимо реализовать визуальный интерфейс программы. Окно приложения делится на 2 части и между этими частями плавающий разделитель QSplitter \*splitter

Левая часть-доступ и хранение данных, правая- отображение (график построенный на основе данных) и взаимодействие с пользователем при помощи виджетов, добавляемых на раскладку : выпадающий список с выбором типа графика ( создаем все типы в функции AddTypeCharts()), чекбокс задающий стиль(цветной/черно-белый), кнопку Печать графика

2.Необходимо организовать вывод графиков разных типов и стиля. Для создания графиков разных типов (6типов определенных в TypeThemeWidget -bar,area,pie, Line, Spline, Scatter.) были написаны методы вида

QChart \***createТипChar**t() const;

Эти методы задают имя типа, потом на основе исходных данных ( в цикле) считают количество данных, а точнее идем по данным таблицы возвращенной из функции **generateDataBase** (const QString& filePath) ключом является отдельная доля в графике, значение- доля занимаемая этой частью (высота столбца например), и по этим данным строят график. Внутри метода определяем цвет графика на основе информации- нажата ли кнопка чекбокс: черно-белый график. В функции updateUI(); на основе выбора пользователем типа графика строиться график того или иного вида, эта функция вызывается каждый раз при изменении пользователем выбора типа из выпадающего списка(&QComboBox::currentIndexChanged).

3.Необходимо проработать кнопку “Печать графика”. Функция openFileDialogWindow() реализует место, куда сохранить файл, также устанавливает фильтр файлов (только файлы бд и json-данные),

QPdfWriter writer\_(fileNames[0] + ".pdf");//сохраняем в формате пдф

А теперь добавляем сам контент в файл-> отрисовываем график.

|  |
| --- |
|  |
|  | QPainter painter(&writer\_); |
|  |  |
|  |  |
|  | chart->ReturnchartView()->render(&painter); |

Следует упомянуть о Концепции программирования MVC ("модель-вид-контроллер"). Разберем, из чего она состоит:

Модель – осуществляет соединение с источником данных, служит их логической моделью,  
предоставляет интерфейс другим компонентам архитектуры.

Вид (представление) – обеспечивает конечное (например, экранное) представление  
данных для пользователя.

Контроллер - отвечает за пользовательский интерфейс. Он обеспечивает связь между  
пользователем и системой. Так как контроллер в QT зашит в представлении и мы напрямую с ним не работаем, то применяется следующая разновидность архитектуры "модель-представление"

При реализации нашей задачи модели представляют собой

fileModel = new QFileSystemModel(this);

dirModel = new QFileSystemModel(this);

А представление- в виде таблицы (данные бд,json) и в виде дерева (папки для выбора сохранения графика при нажатии на кнопку печати)

tableView->setModel(fileModel);

treeView->setModel(dirModel);

С помощью указания model->index указывается  индекс, откуда начали работать с моделью

**3. Код программы.**

<https://github.com/PolinaTkachuk/LAB3_TRPO>

Разрабатывались такие файлы как, themewidget.cpp, themewidge.h, mainwindow.h, mainwindow.cpp, main.cpp

1. **Инструкция пользователя.**

При запуске программы пользователь будет наблюдать данные слева относительно окна приложения. Необходимо кликнуть мышью на определенный файл и тогда справа появится график относительно данных выбранного файла. Чтоб изменить тип графика в разделе “Выберете тип графика” кликаем на выпадающий список и нажимаем на нужный тип. Отмечаем галочкой чекбокс Черно-белый график, если нужно изменить стиль. При нажатии на кнопку печать графика откроется окно с выбором места хранения графика в pdf формате.

Для работы понадобится база данных типа SQLite. В папке проекта имеются несколько баз данных с расширением .sqllite ( например BLOOD\_SUGAR). На основе их будут строиться графики.

1. **Тестирование.**

Тестирование осуществляется при выборе разных баз данных, заранее загруженных в проект

BLOOD\_SUGAR

HUMIDITY\_MOSCOW

NORDPOOL\_PRICES

PRICES\_NATURAL\_GAS\_USD

TEMPERATURE\_NOVOSIB

В зависимости от значения ключей и значений для каждой конкретной бд строятся различные графики. Выбор базы данных осуществляется кликом по ней в оконном приложении.