**Проект. ООТРПО.**

**Участники проекта: гр. 5303 Допира В, Бочкарев И, Ильянов В., гр. 5304 Павлов Д**

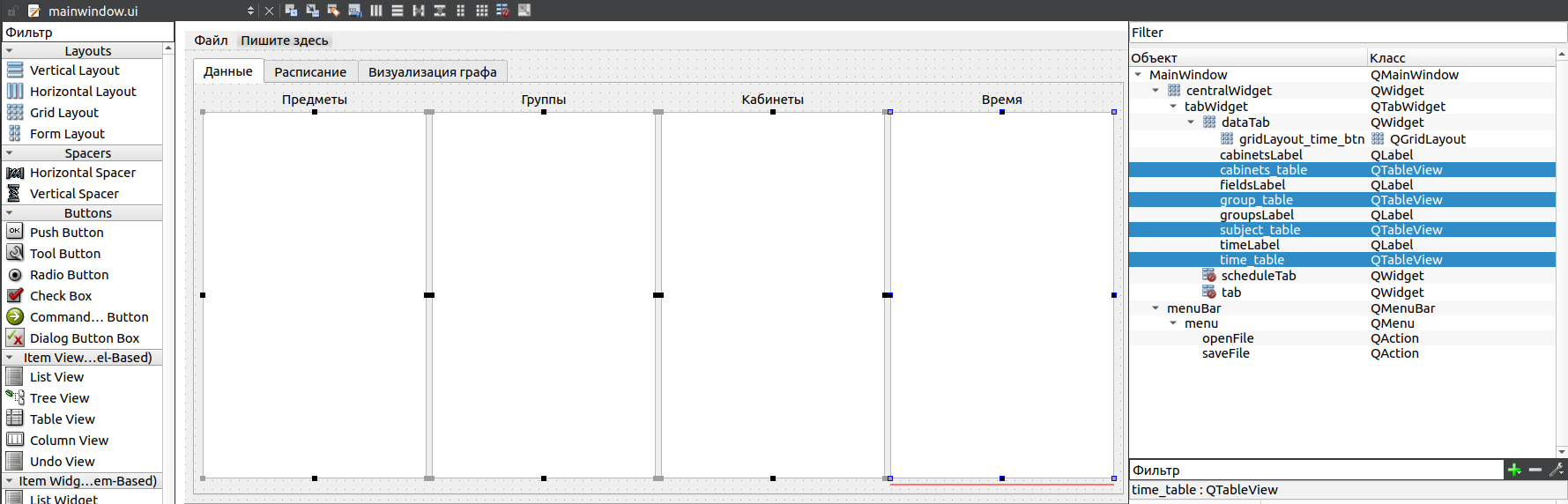
**Ссылка на видео:**

<https://drive.google.com/open?id=1de5bUZ0qsgwSQCq2fscvOsyGv7AqFh9V>

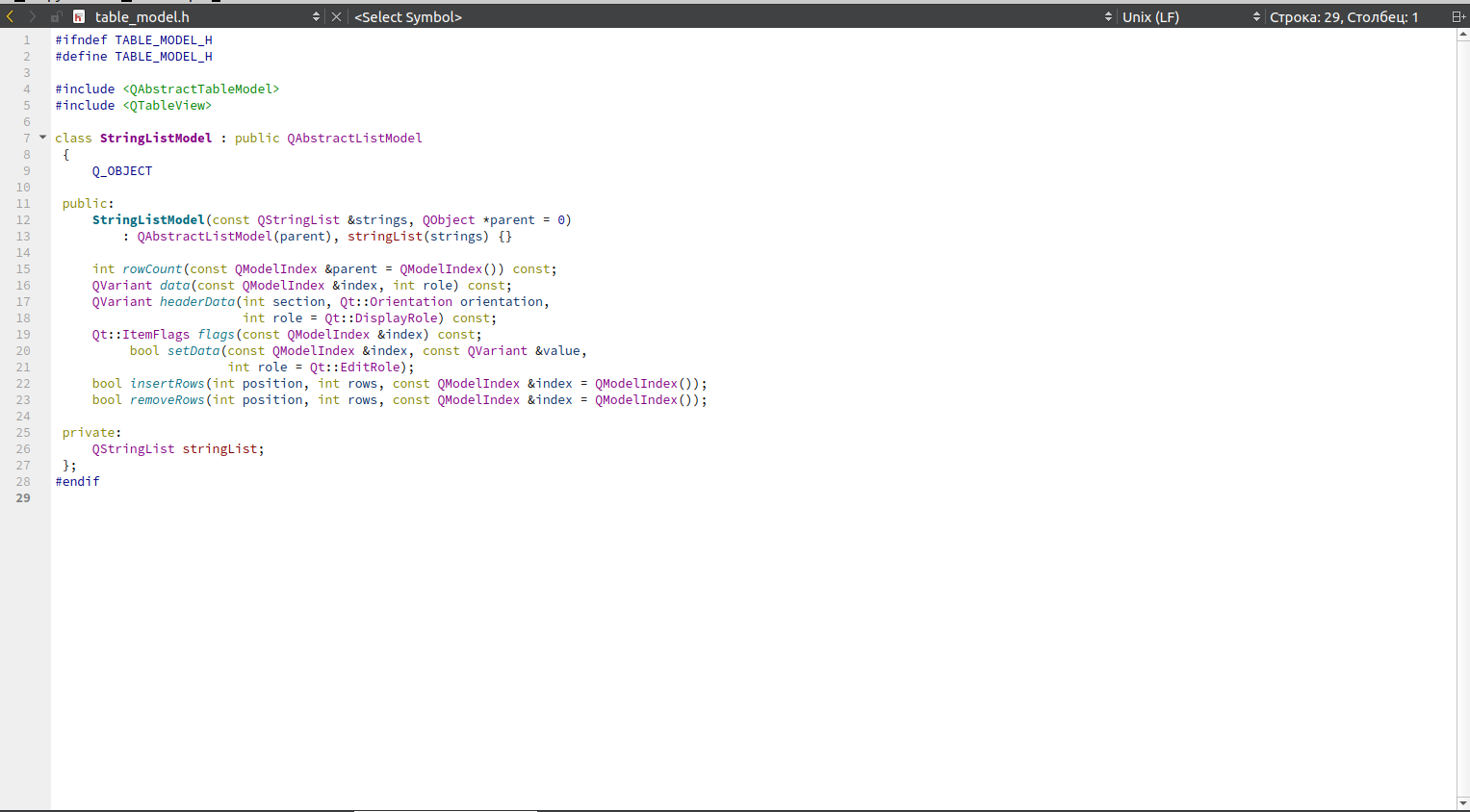
**Сценарий демонстрации работы с интерфейсом приложения**

Таблица subject\_table, потому что все остальные методы работы с таблицами аналогичные.

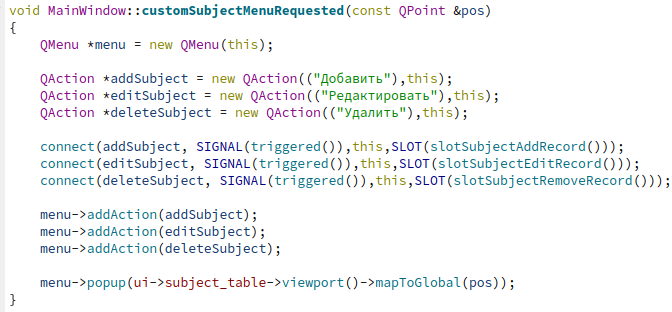
1. Таблицы для работы с записями реализованы с помощью класса QTableView, который отображает элементы из модели.



2. Модель для работы с QTableView была реализована в классе StringListModel, который был унаследован от класса QAbstractListModel, предоставляющий интерфейс используемый представлением QTableView для доступа к данным.



3. Контекстное меню с кнопками(Добавить, Редактировать, Удалить) для каждой таблицы было реализовано с помощью метода: void MainWindow::customSubjectMenuRequested(const QPoint &pos) пример метода:

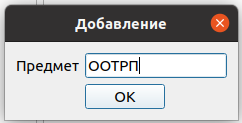
В данном методе создается объект класса QMenu, к которому добавляются объекты QAction, которые отвечают за события на нажатие.

Далее в методе connect происходит отправка сигнала с пункта меню (например: добавить, редактировать, удалить) в соответствующий slot, где будет выполняться соответствующий сигналу код.

Slots, соответствующие сигналам с контекстного меню (для таблицы subject\_table):



В слоте slotSubjectAddRecord открывается диалоговое окно для добавления данных:

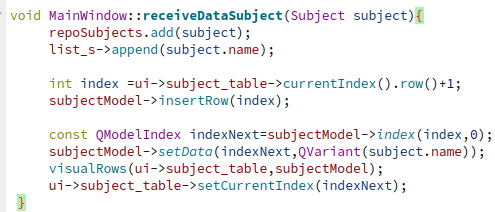


Диалоговое окно «Добавление» для таблицы subject\_table реализуется в классе dialogSubjectWindow.

При добавлении данных, посылается сигнал sendDataSubject(Subject) классу MainWindow с данными:

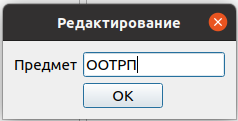


который вызывает slot receiveDataSubject :



В данном методе данные добавляются в объект класса repositoryremplate, который будет работать с графом и в объект класса QStringList на основе данных которого, строится модель для отображение данных в представлении QTableView.

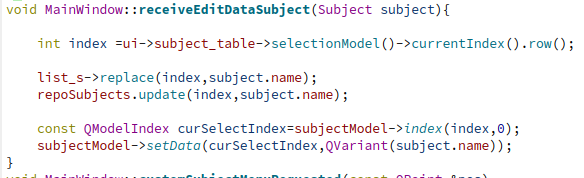
В слоте slotSubjectEditRecord открывается диалоговое окно для добавления данных:



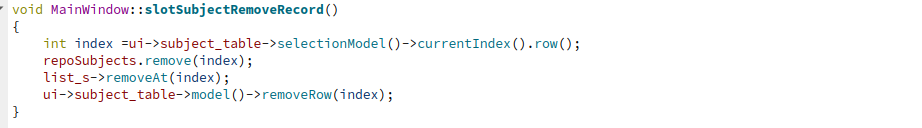
Диалоговое окно «Редактирование» для таблицы subject\_table реализуется в классе dialogSubjectWindow.

При редактировании данных, посылается сигнал sendDataSubject(Subject) классу MainWindow с данными.

который вызывает slot receiveEditDataSubject:

В данном методе данные обновляются в объекте класса repositoryremplate, который будет работать с графом и в объекте класса QStringList на основе данных которого, строится модель для отображение данных в представлении QTableView.

В слоте slotSubjectRemoveRecord реализуется удаление данных с объекта класса repositoryremplate и объекта класса QstringList. Удаление происходит по index модели.



**Визуализация графа**

Отображение графа происходит при помощи виджета QGraphicsView **GraphWidget** , в котором находится сцена для отрисовки объектов

Граф состоит из вершин и ребер.

За визуализацию вершин отвечает класс Node, а за визуализацию ребер класс Edge. Класс Node хранит в себе координаты вершины QPointF pos и список ребер QList<Edge \*> edgeList; , соединяющие смежные вершины. Класс Edge хранив в себе указатели Node \* двух вершин

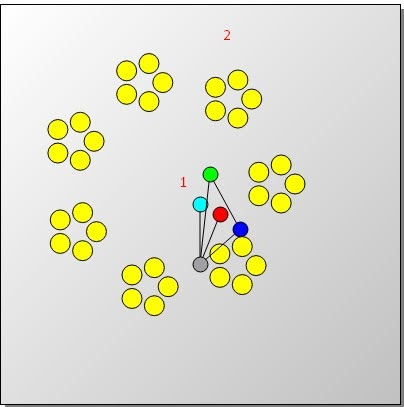
Модуль на данный момент демонстрирует следующие возможности визуализации графа: (см рисунок).

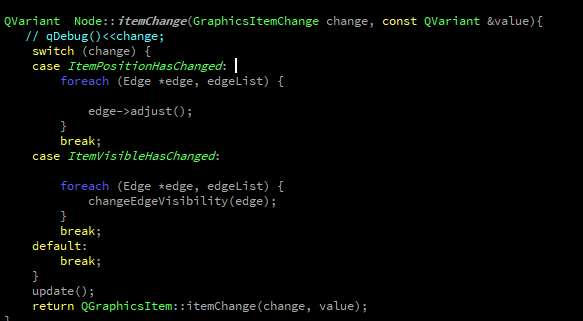
Представление

1. Визуализацию графа с ребрами

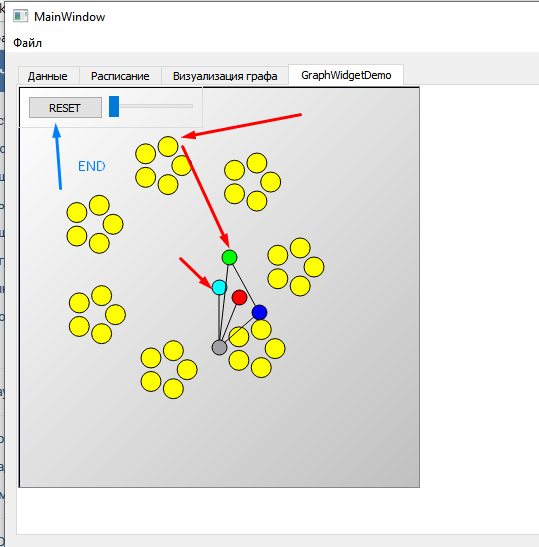
На рис. представлен граф, состоящий из пяти вершин.

При нажатии левой кнопкой можно спрятать выделенную вершину. Когда меняется состояние объекта Node , вызывается функция itemChnage. Если меняется состояние видимости (код ItemVisibleHasChange), то происходит проверка видимости ребер. Если хотя бы одна смежная вершина спрятана , то ребро ичезает с экрана.





При помощи кнопки Reset можно восстановить последнее состояние



Если вершина графа будет спрятана, то смежные ребра тоже будут спрятаны.

1. Визуализацию "цветка" вершин

Для упорядочивания вершин был придумал алгоритм “цветка вершин”

Цветок разделяется на множество листьев, в которых есть семена – вершины графа. Координаты лепестков и листьев определяются, как вершины правильных многоугольников. Код построения лепестков и семян представлена на рис. ниже. На вход подается координаты центра, радиус и число “семян” в лепестке

