

## Лабораторная работа № 3

### РАЗРАБОТКА ОДНО- И МНОГОДОКУМЕНТНОГО ПРИЛОЖЕНИЯ

#### Фреймворк Interview

Разделение способов хранения данных и их представления позволяет использовать одинаковые виджеты для представления данных из различных источников, а также отображать одни и те же данные разными способами одновременно. Кроме того, этот подход позволяет изменять представление и модель данных в значительной степени независимо: необходимо лишь сохранять интерфейс взаимодействия.

Для разделения модели и представления данных библиотекой Qt предоставляется Interview Framework. Эта часть библиотеки реализует измененный шаблон проектирования MVC (модель-представление-контролёр).

Этот фреймворк (каркаса) включает в себя три основных разновидности **классов**:

- Классы модели наследуются от класса `QAbstractItemModel`, служат для предоставления остальным частям фреймворка интерфейса доступа к данным из некоторого источника.
- Классы представления наследуются от класса `QAbstractItemView`, служат для отображения данных пользователю. Данные получают от модели. Для указания на конкретные данные используются индексы модели.
- Классы делегаты наследуются от `QAbstractItemDelegate`, используются стандартными представлениями для отображения и редактирования элементов данных.

Библиотека Qt предоставляет ряд классов моделей и представления.

#### Модели:

- `QStandardItemModel` – общая модель хранения произвольных данных;
- `QFileSystemModel` – модель файловой системы; она позволяет получать атрибуты файлов и каталогов, а также содержимое каталогов;
- `QStringListModel` – модель предоставляющая представлению строки;
- `QSortFilterProxyModel` – модель которая предоставляет возможность сортировки и отбора данных из другой модели; – а также классы для работы с базами данных.
- Абстрактные классы для разных типов моделей.
- `QAbstractTableModel` – абстрактный класс, позволяющий реализовать табличную модель данных;
- `QAbstractListModel` – абстрактный класс, позволяющий реализовать модель данных в виде списка.
- Представления:
- `QTableView` – табличное представление данных;
- `QTreeView` – древовидное представление данных;
- `QListView` – представление данных в виде списка;
- `QComboBox` может использовать модели для получения списка элементов.

Виджеты `QListWidget`, `QTreeWidget` и `QTableWidget` являются вариантами соответствующих представлений, содержащими внутри себя модели и источники данных. Эти виджеты предоставляют доступ к внутренним моделям.

## Задание № 1 «Однодокументный интерфейс»

Будем называть словом последовательность букв и цифр. Будем называть путем последовательность слов, разделенных пробелами.

Разработайте приложение с реализованным на Qt графическим пользовательским интерфейсом, которое предоставляет пользователю следующие возможности:

Загружать список путей из файла. Каждый путь списка записывается в файле в отдельной строке. Пустые строки следует игнорировать.

- Отображать его сжато в виде дерева. Каждому узлу дерева соответствует некоторый путь. Корень дерева соответствует пустому пути и не отображается. Путь, соответствующий не корневому узлу, равен пути предка, к которому через пробел дописали справа название узла. Названия дочерних элементов одного предка должны быть уникальными.

- Добавлять к дереву путей новые узлы.
- В строке состояния должно быть отображено:
  - постоянно количество узлов дерева без потомков (такие узлы называются листьями)
  - при выборе элемента дерева временно полный путь до него.
- Доступ к действиям должен быть предоставлен через пункты меню и панель инструментов.

На рисунке 1 показан пример внешнего вида приложения после открытия файла следующего содержания:

```
1 2 3
2 3 4
1 3 4
1 2 4 4
1 2 3 1
```

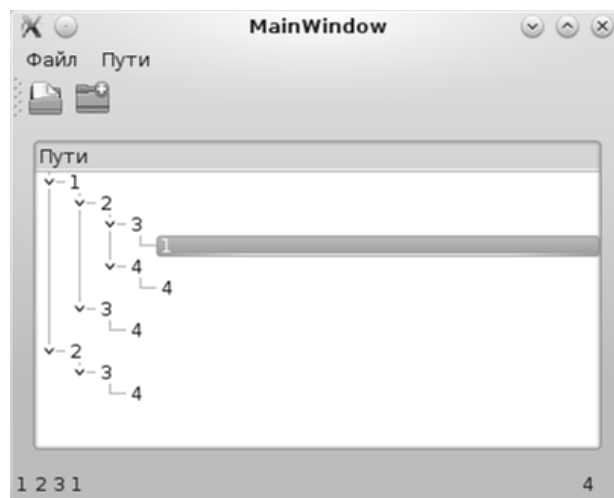


Рисунок 1 - Вид окна приложения

## Задание № 2 « Многодокументный интерфейс»

Ознакомьтесь с документацией по классам QMdiArea, QMdiSubWuidow. QSignalMapper и с примером «MDI Example» (см. соответствующий раздел документации библиотеки Qt и проект Examples\Qt-5.14.1\widgets\mainwindow\mainwindow\mdi в каталоге установки библиотеки Qt).

Измените разработанное в предыдущей лабораторной работе приложение так, чтобы оно предоставляло многодокументный интерфейс. У внутренних окон нет меню, панели инструментов и строки состояния, все действия осуществляются через соответствующие элементы главного окна. Приложение должно позволять работать с несколькими окнами.