Класс Autorisation

Назначение:

Класс Autorisation представляет собой диалоговое окно авторизации, которое запрашивает у пользователя имя пользователя и пароль для доступа к основному окну приложения (MainWindow).

Устройство:

• Приватные переменные-члены:

• ui: Указатель на объект пользовательского интерфейса, созданный Qt Designer.

• window: Указатель на объект главного окна (MainWindow).

• sicretwindow: Указатель на объект окна с секретной информацией (sicret).

• Конструктор:

• Создает диалоговое окно авторизации и настраивает его пользовательский интерфейс.

• Метод on\_pushButton\_clicked():

• Обрабатывает нажатие кнопки "Войти".

• Проверяет правильность введенного имени пользователя и пароля.

• Если данные верны, создает окно с секретной информацией и отображает его.

• Деструктор:

• Удаляет объект диалогового окна авторизации.

Взаимодействие с другими классами:

• Класс Autorisation взаимодействует с классом MainWindow через указатель window. Когда пользователь успешно авторизуется, создается новый объект класса sicret и отображается в окне MainWindow.

Класс MainWindow

Назначение класса:

Класс MainWindow представляет главное окно приложения Qt, предназначенного для работы с табличными данными. Он предоставляет графический пользовательский интерфейс (GUI) и функциональность для управления и редактирования таблицы.

Устройство класса:

Наследование:

• Класс MainWindow наследует от QMainWindow. Это базовый класс для всех главных окон в Qt и предоставляет функции для управления строкой меню, панелью инструментов, строкой состояния и центральным виджетом.

• Класс MainWindow также наследует от Ui::MainWindow, который является автоматически сгенерированным классом Qt, который предоставляет доступ к элементам пользовательского интерфейса, определенным в файле .ui.

Свойства:

• Ui::MainWindow предоставляет доступ к различным элементам пользовательского интерфейса, таким как кнопки, меню и виджеты.

Методы:

• Конструктор: Конструктор создает экземпляр объекта MainWindow и инициализирует его графический пользовательский интерфейс.

• Public Slots:

\* updateActions(): Обновляет состояние действий в главном окне в зависимости от состояния таблицы.

• Private Slots:

• insertChild(): Вставляет дочерний элемент в родительский элемент в таблице.

• insertColumn(): Вставляет столбец в таблицу. Возвращает true, если вставка прошла успешно, в противном случае - false.

• insertRow(): Вставляет строку в таблицу.

• removeColumn(): Удаляет столбец из таблицы. Возвращает true, если удаление прошло успешно, в противном случае - false.

• removeRow(): Удаляет строку из таблицы.

Класс TreeItem

Назначение класса:

Класс TreeItem представляет элемент в древовидной структуре данных. Он используется для создания и управления иерархическими данными, такими как древовидные структуры или таблицы.

Устройство класса:

Конструктор:

• TreeItem(const QVector<QVariant> &data, TreeItem \*parent = 0):

Конструктор создает экземпляр объекта TreeItem с заданными данными и родительским элементом.

Методы:

• child(int number): Возвращает дочерний элемент с указанным номером.

• childCount() const:

Возвращает количество дочерних элементов.

• columnCount() const:

Возвращает количество столбцов в элементе.

• data(int column) const:

Возвращает данные в указанном столбце.

• insertChildren(int position, int count, int columns): Вставляет указанное количество дочерних элементов в указанную позицию с указанным количеством столбцов. Возвращает true, если вставка прошла успешно, в противном случае - false.

• insertColumns(int position, int columns): Вставляет указанное количество столбцов в указанную позицию. Возвращает true, если вставка прошла успешно, в противном случае - false.

• parent(): Возвращает родительский элемент.

• removeChildren(int position, int count): Удаляет указанное количество дочерних элементов из указанной позиции. Возвращает true, если удаление прошло успешно, в противном случае - false.

• removeColumns(int position, int columns): Удаляет указанное количество столбцов из указанной позиции. Возвращает true, если удаление прошло успешно, в противном случае - false.

• childNumber() const: Возвращает номер текущего элемента среди дочерних элементов его родителя.

• setData(int column, const QVariant &value): Устанавливает данные в указанном столбце. Возвращает true, если установка данных прошла успешно, в противном случае - false.

Описание назначения класса, похожее на официальный документ:

Класс TreeItem представляет элемент в древовидной структуре данных. Он предоставляет функции для доступа и управления данными элемента, а также для управления его дочерними элементами и столбцами. Класс TreeItem используется для создания иерархических структур данных, таких как древовидные структуры или таблицы.

Класс TreeModel

Назначение класса:

Класс TreeModel представляет древовидную модель данных, основанную на классе TreeItem. Он предоставляет интерфейс для управления и доступа к иерархически организованным данным в представлении типа "дерево".

Устройство класса:

Наследование:

• Класс TreeModel наследует от QAbstractItemModel. Это базовый класс для всех моделей, предоставляющих данные для представлений Qt, таких как QTreeView и QTableView.

Конструктор:

• TreeModel(const QStringList &headers, const QString &data, QObject \*parent = 0): Конструктор создает экземпляр объекта TreeModel с заданными заголовками столбцов, данными и родительским объектом.

Методы:

• data(const QModelIndex &index, int role) const:

Возвращает данные, связанные с указанным индексом модели, для указанной роли (например, отображаемый текст или данные модели).

• headerData(int section, Qt::Orientation orientation,int role = Qt::DisplayRole) const:

Возвращает данные заголовка для указанного раздела и ориентации.

• index(int row, int column, const QModelIndex &parent = QModelIndex()) const:

Возвращает индекс модели для указанной позиции в указанном родительском индексе.

• parent(const QModelIndex &index) const:

Возвращает родительский индекс для указанного индекса модели.

• rowCount(const QModelIndex &parent = QModelIndex()) const:

Возвращает количество строк в указанном родительском индексе.

• columnCount(const QModelIndex &parent = QModelIndex()) const:

Возвращает количество столбцов в указанном родительском индексе.

• flags(const QModelIndex &index) const:

Возвращает флаги для указанного индекса модели, которые определяют поведение редактирования и выбора.

• setData(const QModelIndex &index, const QVariant &value,int role = Qt::EditRole):

Устанавливает данные для указанного индекса модели и роли.

• setHeaderData(int section, Qt::Orientation orientation, const QVariant &value, int role = Qt::EditRole):

Устанавливает данные заголовка для указанного раздела, ориентации и роли.

• insertColumns(int position, int columns,const QModelIndex &parent = QModelIndex()):

Вставляет указанное количество столбцов в указанную позицию в указанном родительском индексе.

• removeColumns(int position, int columns,const QModelIndex &parent = QModelIndex()):

Удаляет указанное количество столбцов из указанной позиции в указанном родительском индексе.

• insertRows(int position, int rows,const QModelIndex &parent = QModelIndex()):

Вставляет указанное количество строк в указанную позицию в указанном родительском индексе.

• removeRows(int position, int rows, const QModelIndex &parent = QModelIndex()):

Удаляет указанное количество строк из указанной позиции в указанном родительском индексе.

Описание назначения класса

Класс TreeModel представляет древовидную модель данных, основанную на классе TreeItem. Он предоставляет универсальный интерфейс для управления иерархически организованными данными, которые обычно отображаются с помощью QTreeView. TreeModel поддерживает различные операции редактирования и манипуляции данными, такие как вставка, удаление и изменение строк и столбцов.

Класс sicret

Назначение класса:

Класс sicret представляет секретное окно в приложении. Оно предназначено для отображения списка всех возможных логинов и паролей, если пользователь вводит правильные учетные данные при авторизации в основном приложении.

Устройство класса:

Наследование:

• Класс sicret наследует от QDialog. Это базовый класс для всех диалоговых окон в Qt и предоставляет функции для управления окном, его размером, положением и взаимодействием с пользователем.

Члены данных:

• Ui::sicret ui: Указатель на автоматически сгенерированный класс пользовательского интерфейса Qt, который предоставляет доступ к элементам пользовательского интерфейса, определенным в файле .ui.

• QPoint p: Переменная, используемая для хранения положения окна.

Методы:

Конструктор:

Конструктор создает экземпляр объекта sicret и инициализирует его графический пользовательский интерфейс.

Деструктор:

Деструктор освобождает память, выделенную для объекта sicret.

Защищенные методы:

• Переопределенные защищенные методы предоставляют дополнительную функциональность для управления окном и обработкой событий.

Описание назначения класса, похожее на официальный документ:

Класс sicret представляет секретное диалоговое окно, которое отображает список всех возможных логинов и паролей. Он предназначен для использования в приложении, требующем авторизации. При вводе правильных учетных данных при авторизации открывается секретное окно, предоставляя администратору доступ к конфиденциальной информации.