

## Изучение модулей QComboBox

В продолжение предыдущей темы про выбор информации. Есть виджет позволяющий выбирать информацию из предложенных вариантов, но работающий не логически, а по значению.

QComboBox – виджет создающий выпадающий список, позволяет выбирать из ряда текстовых значений.

Основные функции QComboBox:

`addItem(String)` – добавляет элемент в QComboBox

`addItems(String[])` – добавляет несколько элементов в QComboBox, каждый элемент списка – добавляемый в QComboBox элемент

`currentText()` – функция возвращает текст выбранного элемента в String

`currentIndex()` – функция возвращает index выбранного элемента в int

`setMaxCount(int)` – устанавливает ограничение на количество элементов в списке.

`setMaxVisibleItems(int)` – устанавливает количество элементов отображаемых в окне без прокрутки.

`setMinimumContentsLength()` – устанавливает наименьшую возможную длину элементов списка.

`setFrame(bool)` – позволяет вывести фрейм в специальной окошке. True (по умолчанию) – вывод в окошке, False – вывод без окошка.

`setDuplicatesEnabled(bool)` – отвечает за возможность дублирования элементов в списке. По умолчанию стоит False – нельзя дублировать. Можно поставить True – можно дублировать.

## Изучение модуля QDialog

QDialog – модуль для создания диалоговых окон. Принцип работы модуля аналогичен вызову главного окна.

Диалоговое окно – всплывающее во время выполнения окно, предназначенное для общения с пользователем. В диалоговом окне может находиться как информация для оповещения пользователя, так и элементы интерфейса влияющие на выполнение программы. Возможно даже вынесение небольших программ в диалоговые окна.

Как создать диалоговое окно:

1) Импортируйте модуль диалогового окна в программу:

```
from PyQt6.QtWidgets import QDialog
```

2) Создайте класс наследующий свойство QDialog:

```
class CustomDlg(QDialog):
```

3) Внутри класса создайте функцию инициализации и заполните её также, как заполняете основную программу:

4) Создайте переменную внутри класса основной программы и присвойте свойства объекта класса диалогового окна:

```
dlg = CustomDlg()
```

5) Запустите диалоговое окно:

```
dlg.exec()
```

Быстрый способ создания диалогового окна:

1) Импортируйте модуль диалогового окна в программу:

```
from PyQt6.QtWidgets import QDialog
```

2) Создайте переменную с классом QDialog():

```
dlg = QDialog()
```

3) Установите название диалогового окна (setWindowTitle) и добавьте в нее виджеты и слои. Последняя команда dlg.setLayout()

4) Запустите диалоговое окно: dlg.exec()

Функция exec() и её статусы:

Функция exec возвращает статусы по закрытию программы. Внутри диалогового окна для управления статусами закрытия можно применять следующие функции.

accept() – закрывает диалоговое окно и вернет статус 1

reject() – закрывает диалоговое окно и вернет статус 0

finished(int) – закрывает диалоговое окно и возвращает статус int