

ООП
Семестр 2. Qt
Меню. Многооконные приложения.
Черновик

Кафедра ИВТ и ПМ

2019

План

Меню

QTabWidget

Многооконные приложения

Диалоговые окна

Создание новых окон

QMdiArea

QStackedWidget

Ссылки и литература

Outline

Меню

QTabWidget

Многооконные приложения

Диалоговые окна

Создание новых окон

QMdiArea

QStackedWidget

Ссылки и литература

Создание меню

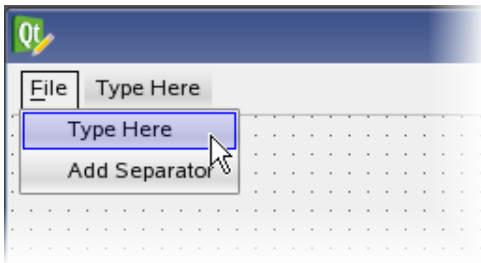
- ▶ Когда создано продолжение из шаблона Qt Widgets Application в главном окне уже присутствует объект menuBar типа QMenuBar.

▼ MainWindow	QMainWindow
▼ centralWidget	QWidget
textEdit	QTextEdit
statusBar	QStatusBar
▼ menuBar	QMenuBar
▼ menuMain	QMenu
▼ menufile	QMenu
actionopen	QAction
actionsave	QAction
▼ menuedit	QMenu
actionclear	QAction
actionread_only	QAction
menuhelp	QMenu

- ▶ Это панель меню окна. По умолчанию меню пустое.

Создание меню

- ▶ Чтобы добавить элемент меню достаточно задать текст пункта меню в дизайнера форм.

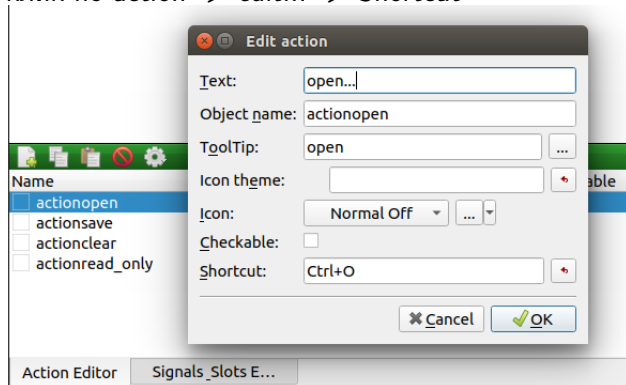


- ▶ Автоматически будет создан новый объект - элемент меню, если добавлено новое меню на панель меню.
- ▶ Для каждого элемента меню, которое не включает в себя другие будет создан объект класса QAction

QAction

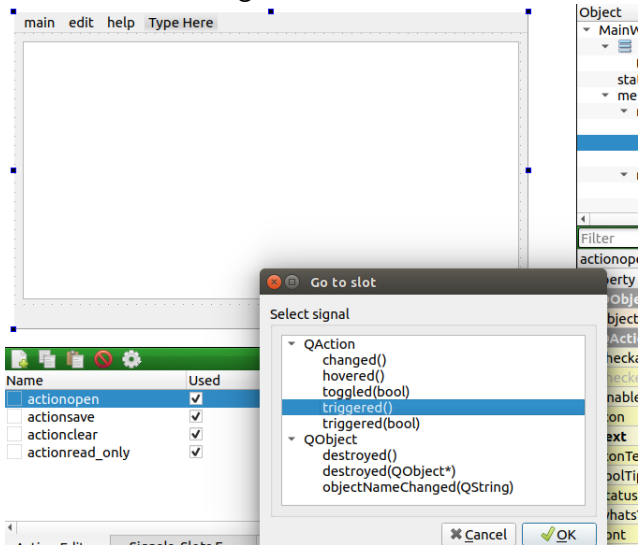
- ▶ QAction будет ожидать действие пользователя, и вызывать сигнал triggered()
- ▶ Объект типа QAction может вызывать сигнал не только по клику на привязанный к нему элемент меню, но и по горячей клавише или нажатию на кнопку на панели инструментов QToolBar.

клик по action -> edit... -> Shortcut



QAction

- ▶ Создать обработчик для срабатывающей action можно также как и для виджета
клик по action -> go to slot...



Обработчик QAction

- Обработчик QAction будет создан как метод в классе окна

```
void MainWindow::on_actionopen_triggered(){
    QString home = QDir::homePath();
    QString fileName = QFileDialog::getOpenFileName(this,
        "Open Image", home, "Text Files (*.txt *.scv)");
    QFile f(fileName);
    if ( f.open(QFile::ReadOnly) ){
        QTextStream s(&f);
        ui->textEdit->setText( s.readAll() ); }
}
```

```
void MainWindow::on_actionsave_triggered(){
    QString home = QDir::homePath();
    QString fileName = QFileDialog::getSaveFileName(this,
        "Open Image", home);
    QFile f(fileName);
    if ( f.open(QFile::WriteOnly) ){
        QTextStream s(&f);
        s << ui->textEdit->toPlainText();
        f.close();    }
}
```


Панель инструментов

- ▶ На панель инструментов `ToolBar` можно помещать `QAction`, которые превратятся в кнопки.
- ▶ Для этого достаточно перетащить соответствующий `QAction` из главного меню приложения на панель
- ▶ Чтобы `QAction` остался в меню при перетаскивании следует зажать `Ctrl`

Обработка горячих клавиш

В конструкторе главного окна:

```
new QShortcut(QKeySequence(Qt::CTRL + Qt::Key_Q),  
             this, SLOT(close()));
```

stackoverflow.com/questions/17631601/how-to-program-custom-keyboard-shortcuts

Outline

Меню

QTabWidget

Многооконные приложения

Диалоговые окна

Создание новых окон

QMdiArea

QStackedWidget

Ссылки и литература

Outline

Меню

QTabWidget

Многооконные приложения

Диалоговые окна

Создание новых окон

QMdiArea

QStackedWidget

Ссылки и литература

Outline

Меню

QTabWidget

Многооконные приложения

Диалоговые окна

Создание новых окон

QMdiArea

QStackedWidget

Ссылки и литература

QFileDialog

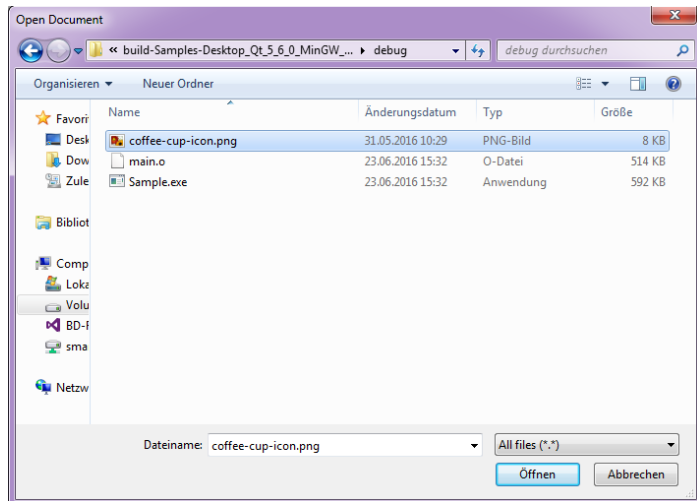
- ▶ Окно с выбором файла.
- ▶ Окно появляется после вызова статического метода `getOpenFileName`
- ▶ Метод возвращает строку - имя файла

```
QString fileName = QFileDialog::getOpenFileName(this,  
    "Open File",      // название окна  
    "/home",          // начальная папка  
    "Images (*.png *.xpm *.jpg)" // фильтр по расширениям  
);
```

- ▶ Для выбора имени нового файла используется метод `getSaveFileName`

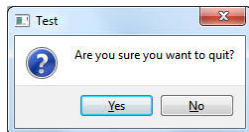
<https://doc.qt.io/qt-5/qfiledialog.html>

QFileDialog



QMessageBox

```
QMessageBox::StandardButton reply;  
reply = QMessageBox::question(this,  
    "Test", "Are you sure you want to quit?",  
    QMessageBox::Yes|QMessageBox::No);  
  
if (reply == QMessageBox::Yes) {  
    qDebug() << "Yes was clicked";  
    QApplication::quit();  
} else {  
    qDebug() << "Yes was *not* clicked";  
}  
  
QString text = QDialog::getText(parent, "Title", "text");
```



Замечание о диалоговых окнах

- ▶ Подобные диалоговые окна стоит показывать для подтверждения действия пользователя, которое нельзя быстро отменить или отменить вообще
- ▶ Например для подтверждения удаления файла
- ▶ Сообщения, которые не несут критически важной информации и не требуют подтверждения сами по себе стоит показывать другим способом, не требующего от пользователя лишних действий. Например вывести сообщение в лейбл, строку статуса или показать исчезающее сообщение средствами ОС.
- ▶ Например диалоговое сообщение о том, что искомая строка не найдена в тексте документа не уместно.
- ▶ Пользователь не должен вступать с информационными сообщениями в диалог.

Outline

Меню

QTabWidget

Многооконные приложения

Диалоговые окна

Создание новых окон

QMdiArea

QStackedWidget

Ссылки и литература

Многооконные приложения

- ▶ Создание многооконных приложений достаточно просто, за счёт того что каждое окно - экземпляр класса
- ▶ Поэтому для создания нового окна достаточно:
 - ▶ Создать класс и файл формы для нового окна
В QtCreator: ПКМ на проекте -> добавить новый -> Qt -> Класс формы QtDesigner.
Будет добавлено три файла: заголовочный и сpp файл для класса и ui файл с дизайном окна
 - ▶ Подключение заголовочного файла в файл класса главного окна
 - ▶ Создание экземпляра класса нового окна (в конструкторе или в отдельном методе)
 - ▶ Вызов метода `show()` для нового окна по событию (например по нажатию на кнопку)

Многооконные приложения

Демонстрация

[youtube.com/watch?v=VigUMAfE2q4](https://www.youtube.com/watch?v=VigUMAfE2q4)

<https://www.youtube.com/watch?v=6e/Y8O20I8>

Создание шаблона окна

stackoverflow.com/questions/48465405/qt-multiple-window-design

Outline

Меню

QTabWidget

Многооконные приложения

Диалоговые окна

Создание новых окон

QMdiArea

QStackedWidget

Ссылки и литература

Outline

Меню

QTabWidget

Многооконные приложения

Диалоговые окна

Создание новых окон

QMdiArea

QStackedWidget

Ссылки и литература

Outline

Меню

QTabWidget

Многооконные приложения

- Диалоговые окна

- Создание новых окон

- QMdiArea

- QStackedWidget

Ссылки и литература

Ссылки и литература

1. youtube: Qt Tutorials For Beginners 18 - QAction, QMenu, QToolBar
2. flaticon.com - свободные иконки
3. <https://www.iconfinder.com>

Ссылки и литература

- ▶ [Qt Википедия](#)
- ▶ [OpenSource версия](#)
- ▶ [Qt wiki](#)

Книги:

- ▶ Qt 5.X. Профессиональное программирование на C++. Макс Шлее. 2015 г. 928 с. Книга периодически обновляется с выходом новых версий фреймворка Qt.
- ▶ Qt. Профессиональное программирование. Разработка кроссплатформенных приложений на C++. Марк Саммерфилд

Материалы курса

Слайды, вопросы к экзамену, задания, примеры

github.com/VetrovSV/OOP