


Програмування GUI


На основі мови C++ та фреймворку Qt

Лекція 10

Діалогові вікна



Діалогове вікно – це центральний елемент, що забезпечує взаємодію між користувачем та програмою.



Діалогові вікна завжди є віджетами верхнього рівня та мають свій заголовок.




Діалогові вікна можуть бути:


1. Модальні
2. Немодальні



Діалогові вікна




Модальні діалогові вікна використовуються для виведення важливих повідомлень, наприклад, для нагадування користувачеві про збереження документа перед закриттям програми. Модальні діалогові вікна переривають роботу програми та для продовження роботи вікно має бути закрите. У момент роботи вікна можливість роботи із самим додатком заблоковано.



Немодальні вікна у своїй появі не переривають роботу програми. Типовий приклад немодального вікна – вікно пошуку тексту у документі.




Діалогові вікна




Діалогові вікна можна розбити на наступні категорії:

1. Власні
2. Стандартні
3. Вікна повідомлень



Діалогове вікно можна створити практично з використанням будь-якого віджету, зробивши його віджетом верхнього рівня. Однак зручніше скористатися спеціально призначеним для цього класом QDialog.




Клас QDialog



Конструктор класу QDialog:

```
QDialog (QWidget *parent=0,  
Qt::WindowFlags f=0),
```



де `parent` – покажчик на батьківський віджет, `f` – прапорець налаштування вікна. Якщо покажчик на батьківський віджет `parent=0`, то діалогове вікно центруватиметься щодо центру екрана, інакше – щодо віджета-батька.



Методи класу QDialog



1. Метод `void show()` – відображає неmodalьне діалогове вікно.



2. Метод `void setModal (bool modal)` – робить діалогове вікно modalьним (`modal=true`) або неmodalьним (`modal=false`).



3. Метод `bool isModal ()` – перевіряє – є вікно modalьним чи неmodalьним.



Слоти класу QDialog



1. Слот `int exec()` – відображає модальне діалогове вікно та повертає одне із значень `enum DialogCode {Rejected, Accepted}`.



`QDialog::Accepted` – була натиснута кнопка ОК

`QDialog::Rejected` – була натиснута кнопка Cancel



2. Слоти `void accept()` та `virtual void reject()` – закривають вікно та встановлюють код повернення, що повертається методом `exec()`, рівним відповідно `QDialog::Accepted` або `QDialog::Rejected`.



Використання стандартних діалогових вікон



`QFileDialog` – діалогове вікно вибору файлів.

Клас `QFileDialog` містить статичні методи, що дозволяють створювати такі діалогові вікна:



```
QString QFileDialog::getOpenFileName (QWidget  
*parent=0, const QString &caption=QString(),  
const QString &dir=QString(), const QString  
&filter=QString(), QString selectedFilter=0,  
Options options=0) – створює та відображає діалогове  
вікно відкриття файлу, повертає повне ім'я файлу в об'єкті класу  
QString.
```



`parent` – покажчик на віджет-предок, `caption` – задає заголовок вікна діалогу, `dir` – шлях до робочої директорії, `filter` – фільтр розширень файлів, що відображаються, `selectedFilter` – вибраний фільтр, `options` – різні параметри вікна діалогу.



Використання стандартних діалогових вікон




2. `QString QFileDialog::getSaveFileName`
(`QWidget *parent=0`, `const QString`
&`caption=QString()`, `const QString`
&`dir=QString()`, `const QString`
&`filter=QString()`, `QString`
`selectedFilter=0`, `Options options=0`) –
створює та відображає діалогове вікно збереження
файлу, повертає повне ім'я файлу в об'єкті класу
`QString`.




Використання стандартних діалогових вікон



```
enum QFileDialog::Option
```

 `QFileDialog::ShowDirsOnly` – відображає лише каталоги (за замовчуванням – папки та файли)

 `QFileDialog::DontConfirmOverwrite` – не запитувати підтвердження у разі перезапису існуючого файлу (за замовчуванням – запитується підтвердження)

 `QFileDialog::DontUseNativeDialog` - не використовувати рідний каталог

```
QFileDialog::ReadOnly
```



Використання стандартних діалогових вікон



enum `QFileDialog::FileMode`

This enum is used to indicate what the user may select in the file dialog; i.e. what the dialog will return if the user clicks OK.

Constant	Value	Description
<code>QFileDialog::AnyFile</code>	0	The name of a file, whether it exists or not.
<code>QFileDialog::ExistingFile</code>	1	The name of a single existing file.
<code>QFileDialog::Directory</code>	2	The name of a directory. Both files and directories are displayed. However, the native Windows file dialog does not support displaying files in the directory chooser.
<code>QFileDialog::ExistingFiles</code>	3	The names of zero or more existing files.

This value is obsolete since Qt 4.5:

Constant	Value	Description
<code>QFileDialog::DirectoryOnly</code>	4	Use <code>Directory</code> and <code>setOption(ShowDirsOnly, true)</code> instead.

See also `setFileMode()`.



Використання стандартних діалогових вікон



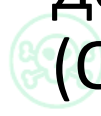
`QColorDialog` – діалогове вікно вибору кольору. Для відображення цього вікна необхідно викликати статичний метод класу `QColor`



```
QColorDialog::getColor (const QColor  
&initial=Qt::white, QWidget *parent=0),  
який повертає обраний колір класу QColor.
```



`initial` – задає колір для ініціалізації каталогу,
`parent` – покажчик на батьківський віджет.




Метод `bool QColor::isValid() const` – дозволяє дізнатися, якою кнопкою було закрито вікно (OK або Cancel). Повертає `false`, якщо було натиснуто кнопку Cancel (тобто об'єкт кольору був створений конструктором за умовчанням без завдання кольору).




Використання стандартних діалогових вікон


 QFontDialog – діалогове вікно вибору шрифту.

 Для відображення цього вікна необхідно викликати статичний метод класу QFont



 QFontDialog::getFont (bool *ok, const QFont &initial, QWidget *parent=0), який повертає вибраний шрифт в об'єкті класу QFont.

 ok – адресу змінної булевого типу, в яку метод запише true , якщо діалогове вікно було закрито кнопкою ОК, або false –якщо кнопкою Cancel, initial – задає тип шрифту для ініціалізації діалогу, parent – показчик на батьківський віджет.

Вікна повідомлень



Вікно повідомлення - простий елемент інтерфейсу користувача, який відображає текстове повідомлення і очікує реакції з боку користувача. Усі вікна класу `QMessageBox` – модальні, можуть містити кнопки (не більше трьох), заголовок та текст повідомлення. Клас `QMessageBox` містить низку статичних методів, які створюють та відображають вікна повідомлень інформаційного, попереджувального, запитального та критичного характеру.



Методи класу QMessageBox



1. Метод `int QMessageBox::information(QWidget *parent, const QString &caption, const QString &text, int button0, int button1=0, int button2=0)` – створює та відображає на екрані інформаційне вікно, що містить мінімум одну кнопку. Повертає ідентифікатор (`QMessageBox::OK`, `QMessageBox::Cancel` і т.д.) натиснутої кнопки.



`caption` – задає заголовок вікна, `text` – містить текст повідомлення, аргументи `button0`, `button1`, `button` – задають кнопки.



Методи класу QMessageBox



enum QMessageBox::StandardButton

QMessageBox::Ok

QMessageBox::Open



QMessageBox::Save

QMessageBox::Cancel

QMessageBox::Close

QMessageBox::Abort



QMessageBox::Retry

QMessageBox::Ignore

QMessageBox::NoButton

QMessageBox::YesToAll



QMessageBox::NoToAll



QMessageBox::Yes

QMessageBox::No






Методи класу QMessageBox



-  1. Метод `int QMessageBox::question (QWidget *parent, const QString &caption, const QString &text, int button0, int button1=0, int button2=0)` – створює та відображає на екрані вікно питання. Повертає ідентифікатор (`QMessageBox::OK`, `QMessageBox::Cancel` і т.д.) натиснутою кнопки.
-  2. Метод `int QMessageBox::warning (QWidget *parent, const QString &caption, const QString &text, int button0, int button1, int button2=0)` створює та відображає на екрані вікно попереджувального повідомлення. Повертає ідентифікатор кнопки.

Методи класу QMessageBox



-  4. `int QMessageBox::critical (QWidget *parent, const QString &caption, const QString &text, int button0, int button1, int button2=0)` – **вікно критичного повідомлення.**
-  5. `void QMessageBox::about (QWidget *parent, const QString &caption, const QString &text)` – **створює та відображає на екрані вікно з інформацією про програму.**
-  6. `void QMessageBox::aboutQt (QWidget *parent, const QString &caption=QString::null)` - **створює та відображає на екрані вікно з інформацією про бібліотеку Qt.**

