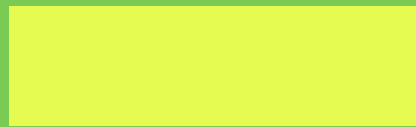


```
Build options:
Mode ..... release
Optimize release build for size ..... yes
Building shared libraries ..... no
Using C++ standard ..... C++1z
Using ccache ..... no
Using gold linker ..... no
Using precompiled headers ..... no
Using LTCG ..... no
Target compiler options:
SSE ..... SSE4.1 SSE4.2
AVX ..... AVX512
AVX512 ..... AVX512
Other x86 ..... Other x86
Build parts .....
App store com .....
Qt modules and .....
Qt Concurrent .....
Qt D-Bus .....
Qt D-Bus dire .....
Qt Gui .....
Qt Network ..... yes
Qt Sql ..... yes
Qt Testlib ..... yes
Qt Widgets ..... yes
Qt Xml ..... yes
Support enabled for:
```



QT: Input/Output and Networking

Кравець Ольга
ПМО-31





QIODevice

ЦЕ АБСТРАКТНИЙ КЛАС, УЗАГАЛЬНЮЮЧИЙ
ПРИСТРІЙ ВВЕДЕННЯ / ВИВЕДЕННЯ, ЯКИЙ
МІСТИТЬ ВІРТУАЛЬНІ МЕТОДИ ДЛЯ ВІДКРИТТЯ І
ЗАКРИТТЯ ПРИСТРОЇВ ВВЕДЕННЯ / ВИВЕДЕННЯ, А
ТАКОЖ ДЛЯ ЧИТАННЯ І ЗАПИСУ БЛОКІВ ДАНИХ
АБО ОКРЕМИХ СИМВОЛІВ.



Класи, які наслідують клас QIODevice:



QFILE

клас для
проведення
операцій з
файлами

QBUFFER

дозволяє записувати
і зчитувати дані в
масив QByteArray,
начебто це пристрій
або файл

QABSTRACTSOCKET

базовий клас для
мережевої комунікації
за допомогою сокетів

QPROCESS

клас надає можливість
запуску процесів з
додатковими аргументами і
дозволяє обмінюватися
інформацією з цими
процесами за допомогою
методів, визначених у
QIODevice

Обробка файлів

- **QFILE**
- **QDIR**
- **QDIRITERATOR**
- **QFILEINFO**



QFILE

EXISTS() - МЕТОД ДЛЯ ПЕРЕВІРКИ ІСНУВАННЯ

REMOVE() - МЕТОД ДЛЯ ВИДАЛЕННЯ ФАЙЛА

OPEN() - МЕТОД ДЛЯ ЧИТАННЯ / ЗАПISУ ФАЙЛА
(ПОВЕРТАЄ TRUE / FALSE)

ISOPEN() - МЕТОД ПЕРЕВІРКИ ЧИ ВІДКРИТИЙ ФАЙЛ

CLOSE() - МЕТОД, ЩОБ ЗАКРИТИ ФАЙЛ

ФАЙЛ МОЖНА ВІДКРИТИ В ТАКИХ РЕЖИМАХ:

IO_READONLY

IO_WRITEONLY

IO_READWRITE

IO_APPEND



ПОТОКИ

QTEXTSTREAM

Типи даних:

char
QChar
char *
QString
QByteArray
short
int
long
float
double

QDATASTREAM

Типи даних:

QByteArray
QFont
QImage
QMap
QPixmap
QString
QValueList
Variant

ШЛЯХИ

ОСНОВНІ МЕТОДИ:

`current()`

`root()`

`home()`

`exists()`

`cd(dirName)`

`cdUp()`

`isRoot()`

`mkdir(const QString & dirName)`

`rmdir(const QString & dirName)`

`rename()`

`setSorting(flags)` (flags можуть бути такими: `QDir::Name`, `QDir::Time`, `QDir::Size` та `Unsorted`)

`setFilter(filters)` (для `entryList()` та `entryInfoList()`)

Мережа

QTCP SOCKET

забезпечує асинхронне
буферизоване TCP з'єднання

QABSTRACT SOCKET

платформо незалежна
оболонка для оригінальних
сокетів API

QNETWORK PROXY

здійснює підтримку проксі
сервера

QNETWORK ACCESS MANAGER I

забезпечує менеджмент
інфраструктуру для мережевих
операцій

QNETWORK INTERFACE

забезпечую інформацію про
мереживний інтерфейс
комп'ютера

| Практика!

ДОКУМЕНТАЦІЯ:

[HTTPS://DOC.QT.IO/QT-5/IO.HTML](https://doc.qt.io/qt-5/io.html)