|  |  |
| --- | --- |
| ДИСЦИПЛИНА | Технологии индустриального программирования |
| ИНСТИТУТ | ИПТИП |
| КАФЕДРА | Индустриального программирования |
| ВИД УЧЕБНОГО МАТЕРИАЛА | Методические указания к практическим занятиям по дисциплине |
| ПРЕПОДАВАТЕЛЬ | Александров Алексей Сергеевич |
| СЕМЕСТР | 3 семестр, 2024/2025 уч. год |

# Практическое занятие №7

## Текстовые потоки и файлы

Текстовый поток – это механизм, позволяющий взаимодействовать с устройством ввода-вывода в текстовом формате. Таким устройством может быть файл, терминал, буфер, сокет, программа и т.д. Текстовый поток используется для ввода или вывода информации в текстовом формате – символов алфавита и цифр.

В рамках данной практической работы будет рассмотрено использование текстовых потоков на примере работы с файлом.

Для работы с файлами в Qt существует класс QFile, позволяющий создавать, открывать, редактировать и сохранять файлы. Для его подключения необходимо использовать директиву #include <QFile>.

При создании объекта QFile ему необходимо передать путь к файлу в качестве параметра конструктора. Указать можно, как абсолютный, так и относительный путь.

Абсолютный (полный) путь — это путь, который указывает на одно и то же место в файловой системе, вне зависимости от текущего рабочего каталога или других обстоятельств. Полный путь всегда начинается с корневого каталога. Относительный путь представляет собой путь по отношению к текущему рабочему каталогу.

Для взаимодействия с файлом его необходимо открыть с одним из режимов доступа:

* ReadOnly – открытие устройства только для чтения данных;
* WriteOnly – открытие устройства только для записи данных, все записанные ранее данные удаляются, используется для создания файлов;
* ReadWrite — открытие устройства для чтения и записи данных   
  (то же, что ReadOnly | WriteOnly);
* Append — открытие устройства для добавления данных;

Таким образом открытие файла для записи происходит следующим образом:

#include <QFile>

QFile file("file.txt");

if(file.open(QIODevice::WriteOnly))

{

// файл успешно открыт для записи

}

Запись в файл производится с помощью метода write():

QFile file("file.txt");

if(file.open(QIODevice::WriteOnly))

{

file.write("Hello world!");

}

После записи файл необходимо закрыть, с помощью метода close():

QFile file("file.txt");

if(file.open(QIODevice::WriteOnly))

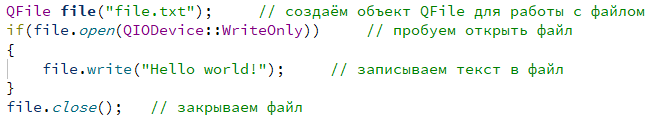
{

file.write("Hello world!");

}

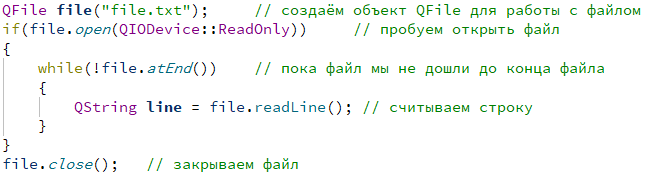
file.close();

Таким образом полный код будет выглядеть следующим образом:



Код записи текста в файл

Чтение файла выполняется аналогичным образом, но открыть файл необходимо с флагом ReadOnly и использовать метод readLine() для считывая файл построчно, пока файл не закончится. Нахождение каретки в конце файла проверяется вызовом метода atEnd():



Код чтения файла

Таким образом возможна простая работа с текстовыми файлами. Однако, если необходимо выполнять чтение и запись не только строк, но и числовых и других данных, необходимо использовать текстовые потоки.

В Qt текстовые потоки реализуются с помощью класса QTextStream, для подключения которого необходимо использовать директиву #include <QTextStream>.

Создание текстового потока производится следующим образом:

QFile file("file.txt"); // создаём объект QFile для работы с файлом

if(!file.open(QIODevice::WriteOnly))

{

// ошибка открытия файла

return;

}

QTextStream ts(&file); // создаём текстовый поток для файла

В данном случае, при открытии файла выполняется проверка, и, в случае если открыть файл не удалось, производится обработка ошибки. Далее, при успешном открытии файла создаётся объект класса QTextStream, в конструктор которого передаётся ссылка на объект файла.

Запись данных производится с помощью оператора << в который передаётся значение или переменная, которое нужно записать:

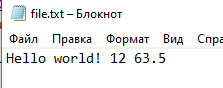
QTextStream ts(&file); // создаём текстовый поток для файла

ts << "Hello world!"; // запись в файл с помощью потока

ts << " " << 12 << " " << 63.5; // запись строк и чисел

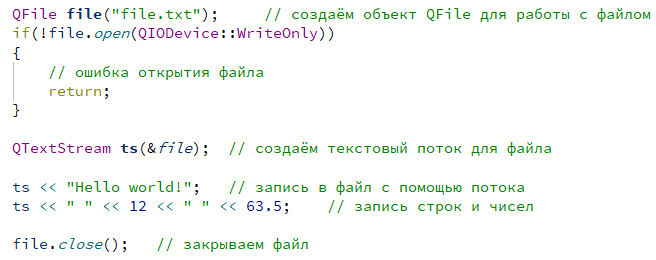
file.close(); // закрываем файл

Результат записи будет следующий:



Результат записи в текстовый файл

Таким образом полный код будет выглядеть следующим образом:



Код записи в текстовый файл с помощью текстового потока

Чтение данных производится оператором >>. В данном случае указывается переменная, в которую будет записан результат. Значения в данном случае считываются, разделённые пробелом, т.е. при чтении данных из файла за один раз будут считываться все символы, до того момента, пока на пути не будет пробела. Следующее значение будет считываться после этого пробела и до следующего. Таким образом Hello world будет считано как 2 отдельных значения:

QFile file("file.txt"); // создаём объект QFile для работы с файлом

if(!file.open(QIODevice::ReadOnly))

{

// ошибка открытия файла

return;

}

QTextStream ts(&file); // создаём текстовый поток для файла

QString h, w;

int v1;

double d1;

ts >> h >> w >> v1 >> d1;

// h = "Hello"

// w = "world!"

// v1 = 12

// d1 = 63.5

file.close(); // закрываем файл

Для удобства работы с программой пользователю необходимо предоставить возможность выбора куда сохранять файл и какой файл открыть. Для этого в Qt реализованы диалоговые окна выбора файлов, с помощью класса QFileDialog.

Для его подключения используется директива #include <QFileDialog>.

Вызвать диалог выбора пути сохранения файла можно с помощью статического метода QFileDialog::getSaveFileName():

QString fileName =

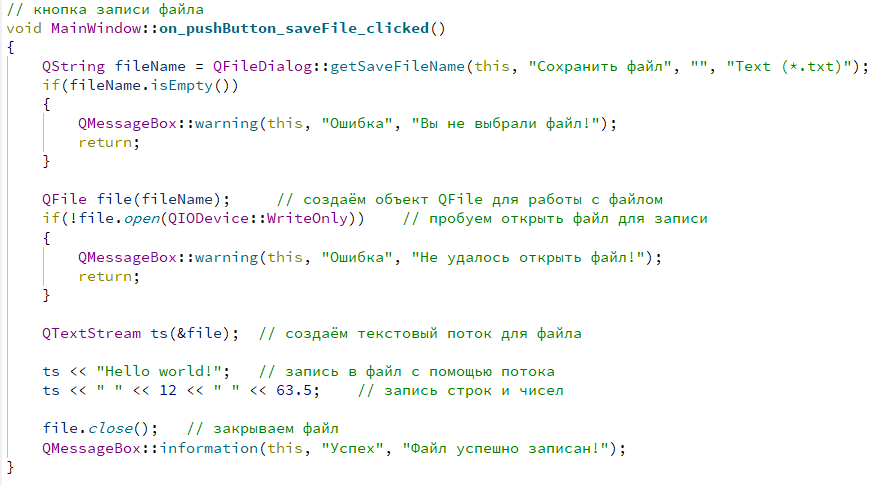
QFileDialog::getSaveFileName(this, "Сохранить файл", "", "Text (\*.txt)");

При вызове метода указывается заголовок («Сохранить файл»), стартовый путь, который можно оставить пустым (как в этом примере) – тогда диалог будет открыт в папке с программой, а также расширения файлов, которые необходимо отображать, или оставить пустым, тогда будут доступны для выбора все расширения файлов.

Если пользователь выбрал файл, его абсолютный путь будет записан в переменную fileName.

Однако, если файл не был выбран – пользователь закрыл окно или нажал «Отмена», в переменную filename будет записана пустая строка.

Таким образом полный код выбора файла и записи в него информации будет выглядеть следующим образом:



Код выбора и записи файла.

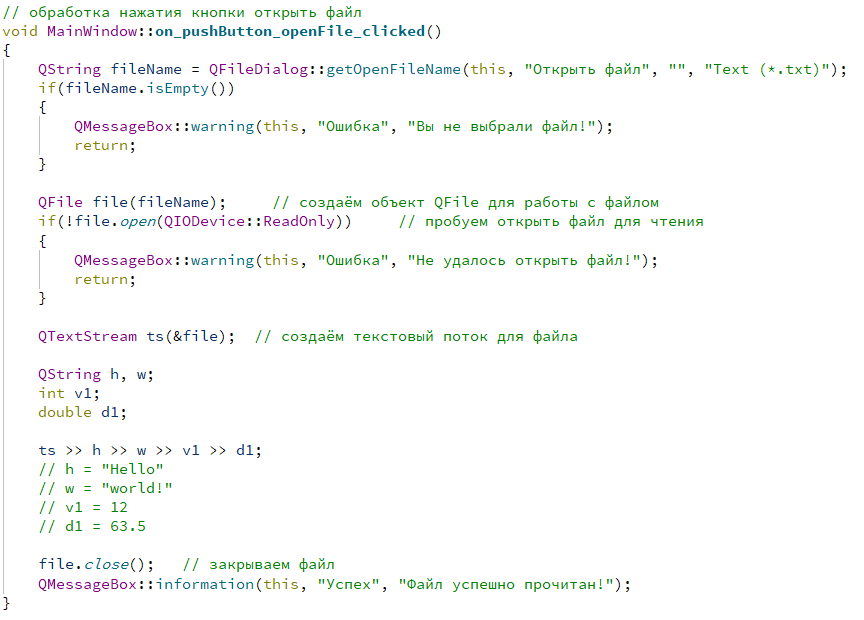
Диалоговое окно выбора файла для чтения выполняется аналогичным образом.

Для выбора файла, который необходимо открыть, используется статический метод QFileDialog::getOpenFileName():

QString fileName =

QFileDialog::getOpenFileName(this, "Открыть файл", "", "Text (\*.txt)");

Дальнейшие действия аналогичны:



Код открытия и чтения файла.

Таким образом возможно использование диалоговых окон для выбора файла, а также чтение и запись данных в текстовый файл с помощью текстовых потоков.