

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

КАФЕДРА ЕЛЕКТРОННИХ ОБЧИСЛЮВАЛЬНИХ МАШИН

Звіт  
з лабораторної роботи №1  
з дисципліни «Крос-платформне програмування»  
на тему: «Створення консольного додатка в QT CREATOR»

Виконав:  
ст. гр. КІУКІ-23-1  
Каплюх Д.О.

Прийняв:  
Михайліченко І.В.

Харків 2025

# **1 СТВОРЕННЯ КОНСОЛЬНОГО ДОДАТКА В QT CREATOR**

## **1.1 Мета роботи**

Оволодіти базовими навичками створення та налагодження консольних проєктів у середовищі Qt Creator, навчитися коректно працювати з текстовим введенням/виведенням і файлами засобами Qt (QFile, QTextStream), опанувати базові стратегії обробки аргументів командного рядка та організації даних за допомогою контейнерів Qt, познайомитися з інструментами діагностики (QDebug, qWarning, qCritical) і принципами обробки помилок.

## **1.2 Варіант 9**

Є два вхідні файли зі списками рядків. Необхідно знайти їхній перетин та різницю. Результат записати у три файли: спільні елементи, елементи лише з першого файла, елементи лише з другого файла. Передбачити можливість ігнорування регістру при порівнянні.

## **1.3 Хід роботи**

### **1.3.1 Вхідні дані**

Створюємо 2 вхідні файли (file1.txt та file2.txt) у папці проєкту.

Вміст файлу file1.txt:

Apple

Banana

cherry

Вміст файлу file2.txt:

banana

Cherry

Date

Дані файли потрібні для коректного виконання завдання.

### 1.3.2 Створення консольного проєкту

У Qt Creator створюємо новий консольний проєкт, що за замовчуванням має файли CMakeList.txt та main.cpp.

Лістинг 1 – вміст файлу CMakeList.txt:

```
cmake_minimum_required(VERSION 3.16)

project(LB1 LANGUAGES CXX)

set(CMAKE_AUTOUIC ON)
set(CMAKE_AUTOMOC ON)
set(CMAKE_AUTORCC ON)

set(CMAKE_CXX_STANDARD 17)
set(CMAKE_CXX_STANDARD_REQUIRED ON)

find_package(QT NAMES Qt6 Qt5 REQUIRED COMPONENTS Core)
find_package(Qt${QT_VERSION_MAJOR} REQUIRED COMPONENTS Core)

add_executable(LB1
    main.cpp
)
target_link_libraries(LB1 Qt${QT_VERSION_MAJOR}::Core)

include(GNUInstallDirs)
install(TARGETS LB1
    LIBRARY DESTINATION ${CMAKE_INSTALL_LIBDIR}
    RUNTIME DESTINATION ${CMAKE_INSTALL_BINDIR}
)
```

Лістинг 2 - вміст файлу main.cpp:

```
#include <QCoreApplication> // життєвий цикл застосунку
#include <QTextStream> // введення/виведення тексту
```

```

#include <QFile> // робота з файлами
#include <QHash> // хеш-таблиця
#include <QSet> // множина унікальних значень (на базі QHash)
#include <QDebug> // інтерфейс для форматування діагностичних повідомлень

int main(int argc, char *argv[])
{
    QApplication app(argc, argv); // Ініціалізація додатку Qt

    QStringList args = app.arguments(); //Отримання аргументів командного
рядка

    // Перевірка кількості аргументів
    if (args.size() < 3) {
        qWarning() << "Використання: " << args.at(0) << " <file1> <file2>
[-i]";

        qWarning() << "Де -i - опціональний флаг для ігнорування
регістру.";

        return 1;
    }

    // Визначення шляхів до файлів
    QString file1Path = args.at(1);
    QString file2Path = args.at(2);

    // Перевірка опції ігнорування регістру
    bool ignoreCase = false;
    if (args.size() > 3 && (args.at(3) == "-i" || args.at(3) == "--ignore-
case")) {
        ignoreCase = true;
        qInfo() << "Ігнорування регістру активовано.";
    }

    // Контейнери для зберігання рядків
    QSet<QString> set1; // Оригінальні рядки з file1
    QSet<QString> set2; // Оригінальні рядки з file2
    QHash<QString, QString> normalizedToOriginal1; // Для file1: нижній
регістр -> оригінал
    QHash<QString, QString> normalizedToOriginal2; // Для file2: нижній
регістр -> оригінал

    // Читання першого файлу
    QFile file1(file1Path);

```

```

        if (!file1.open(QIODevice::ReadOnly | QIODevice::Text)) {
            qCritical() << "Не вдалося відкрити файл №1 для читання:" <<
file1.errorString();
            return 1;
        }
        QTextStream in1(&file1);
        in1.setEncoding(QStringConverter::Utf8);
        while (!in1.atEnd()) {
            QString line = in1.readLine().trimmed();
            if (!line.isEmpty()) {
                set1.insert(line);
                if (ignoreCase) {
                    normalizedToOriginal1[line.toLowerCase()] = line;
                } else {
                    normalizedToOriginal1[line] = line;
                }
            }
        }
        file1.close();
        qInfo() << "Зчитано " << set1.size() << " унікальних рядків з файлу
№1.";

```

```

// Читання другого файлу
QFile file2(file2Path);
if (!file2.open(QIODevice::ReadOnly | QIODevice::Text)) {
    qCritical() << "Не вдалося відкрити файл №2 для читання:" <<
file2.errorString();
    return 1;
}
QTextStream in2(&file2);
in2.setEncoding(QStringConverter::Utf8);
while (!in2.atEnd()) {
    QString line = in2.readLine().trimmed();
    if (!line.isEmpty()) {
        set2.insert(line);
        if (ignoreCase) {
            normalizedToOriginal2[line.toLowerCase()] = line;
        } else {
            normalizedToOriginal2[line] = line;
        }
    }
}
file2.close();

```

```
qInfo() << "Зчитано " << set2.size() << " унікальних рядків з файлу  
№2.";
```

```
// Обчислення перетину і різниць
```

```
QSet<QString> common;
```

```
QSet<QString> onlyFirst;
```

```
QSet<QString> onlySecond;
```

```
// Перетин і різниці з урахуванням регістру
```

```
if (ignoreCase) {
```

```
    for (const QString &line : set1) {
```

```
        QString normLine = line.toLower();
```

```
        if (normalizedToOriginal2.contains(normLine)) {
```

```
            common.insert(normalizedToOriginal1[normLine]);
```

```
        } else {
```

```
            onlyFirst.insert(line);
```

```
        }
```

```
    }
```

```
    for (const QString &line : set2) {
```

```
        QString normLine = line.toLower();
```

```
        if (!normalizedToOriginal1.contains(normLine)) {
```

```
            onlySecond.insert(line);
```

```
        }
```

```
    }
```

```
} else {
```

```
    common = set1.intersect(set2);
```

```
    onlyFirst = set1.subtract(set2);
```

```
    onlySecond = set2.subtract(set1);
```

```
}
```

```
// Вивід статистики
```

```
qInfo() << "Спільні елементи: " << common.size();
```

```
qInfo() << "Тільки в першому файлі: " << onlyFirst.size();
```

```
qInfo() << "Тільки в другому файлі: " << onlySecond.size();
```

```
// Запис результатів у файли
```

```
// Спільні елементи
```

```
QFile commonFile("common.txt");
```

```
if (!commonFile.open(QIODevice::WriteOnly | QIODevice::Text)) {
```

```
    qCritical() << "Не вдалося відкрити common.txt:" <<
```

```
commonFile.errorString();
```

```
    return 1;
```

```
}
```

```

    QTextStream outCommon(&commonFile);
    outCommon.setEncoding(QStringConverter::Utf8);
    for (QSet<QString>::const_iterator it = common.constBegin(); it !=
common.constEnd(); ++it) {
        outCommon << *it << Qt::endl;
    }
    commonFile.close();

    // Тільки перший файл
    QFile onlyFirstFile("only_first.txt");
    if (!onlyFirstFile.open(QIODevice::WriteOnly | QIODevice::Text)) {
        qCritical() << "Не вдалося відкрити only_first.txt:" <<
onlyFirstFile.errorString();
        return 1;
    }
    QTextStream outOnlyFirst(&onlyFirstFile);
    outOnlyFirst.setEncoding(QStringConverter::Utf8);
    for (QSet<QString>::const_iterator it = onlyFirst.constBegin(); it !=
onlyFirst.constEnd(); ++it) {
        outOnlyFirst << *it << Qt::endl;
    }
    onlyFirstFile.close();

    // Тільки другий файл
    QFile onlySecondFile("only_second.txt");
    if (!onlySecondFile.open(QIODevice::WriteOnly | QIODevice::Text)) {
        qCritical() << "Не вдалося відкрити only_second.txt:" <<
onlySecondFile.errorString();
        return 1;
    }
    QTextStream outOnlySecond(&onlySecondFile);
    outOnlySecond.setEncoding(QStringConverter::Utf8);
    for (QSet<QString>::const_iterator it = onlySecond.constBegin(); it
!= onlySecond.constEnd(); ++it) {
        outOnlySecond << *it << Qt::endl;
    }
    onlySecondFile.close();

    // Повідомлення про завершення
    qInfo() << "Результати записано в common.txt, only_first.txt,
only_second.txt.";

    return 0;

```

```
}
```

### 1.3.3 Вихідні результати

Після виконання програми маємо відповідь у Application Output:

```
12:51:14: Starting
D:\NurePrograms\QT_Designs\LB1\build\Desktop_Qt_6_9_2_MinGW_64_bit-
Debug\LB1.exe
D:\NurePrograms\QT_Designs\LB1\build\Desktop_Qt_6_9_2_MinGW_64_bit-
Debug\file1.txt
D:\NurePrograms\QT_Designs\LB1\build\Desktop_Qt_6_9_2_MinGW_64_bit-
Debug\file2.txt -i...
    Ігнорування реєстру активовано.
    Зчитано 3 унікальних рядків з файлу №1.
    Зчитано 3 унікальних рядків з файлу №2.
    Спільні елементи: 2
    Тільки в першому файлі: 1
    Тільки в другому файлі: 1
    Результати записано в common.txt, only_first.txt, only_second.txt.
12:51:14: The command
"D:\NurePrograms\QT_Designs\LB1\build\Desktop_Qt_6_9_2_MinGW_64_bit-
Debug\LB1.exe
D:\NurePrograms\QT_Designs\LB1\build\Desktop_Qt_6_9_2_MinGW_64_bit-
Debug\file1.txt
D:\NurePrograms\QT_Designs\LB1\build\Desktop_Qt_6_9_2_MinGW_64_bit-
Debug\file2.txt -i" finished successfully.
```

Програма створила 3 (.txt) файли з вмістом, а саме:

Файл only\_first:

Apple

Файл only\_second:

Date

Файл Common:

cherry

Banana



### 1.3.4 Скріншоти роботи

```
12:51:14: Starting D:\NurePrograms\QT_Designs\LB1\build\Desktop_Qt_6_9_2_MinGW_64_bit-Debug\LB1.exe D:
\nurePrograms\QT_Designs\LB1\build\Desktop_Qt_6_9_2_MinGW_64_bit-Debug\file1.txt D:
\nurePrograms\QT_Designs\LB1\build\Desktop_Qt_6_9_2_MinGW_64_bit-Debug\file2.txt -i...
Ігнорування реєстру активовано.
Зчитано 3 унікальних рядків з файлу №1.
Зчитано 3 унікальних рядків з файлу №2.
Спільні елементи: 2
Тільки в першому файлі: 1
Тільки в другому файлі: 1
Результати записано в common.txt, only_first.txt, only_second.txt.
12:51:14: The command "D:\NurePrograms\QT_Designs\LB1\build\Desktop_Qt_6_9_2_MinGW_64_bit-Debug\LB1.exe D:
\nurePrograms\QT_Designs\LB1\build\Desktop_Qt_6_9_2_MinGW_64_bit-Debug\file1.txt D:
\nurePrograms\QT_Designs\LB1\build\Desktop_Qt_6_9_2_MinGW_64_bit-Debug\file2.txt -i" finished successfully.
```

Рисунок 1 – інформація з Application Output












	.cmake	15.10.2025 15:14	Папка с файлами	
	.qt	15.10.2025 15:14	Папка с файлами	
	.qtc	15.10.2025 15:14	Папка с файлами	
	.qtc_clangd	15.10.2025 15:14	Папка с файлами	
	CMakeFiles	15.10.2025 15:31	Папка с файлами	
	LB1_autogen	15.10.2025 15:31	Папка с файлами	
	Testing	15.10.2025 15:14	Папка с файлами	
	.ninja_deps	16.10.2025 12:51	Файл "NINJA_DEP..."	64 КБ
	.ninja_log	16.10.2025 12:51	Файл "NINJA_LOG"	2 КБ
	build.ninja	15.10.2025 15:14	Файл "NINJA"	42 КБ
	cmake_install.cmake	15.10.2025 15:14	Файл "CMAKE"	3 КБ
	CMakeCache	15.10.2025 15:14	Text Document	38 КБ
	common	16.10.2025 12:51	Text Document	1 КБ
	file1	15.10.2025 15:36	Text Document	1 КБ
	file2	15.10.2025 15:36	Text Document	1 КБ
	LB1	16.10.2025 12:51	Приложение	719 КБ
	only_first	16.10.2025 12:51	Text Document	1 КБ
	only_second	16.10.2025 12:51	Text Document	1 КБ
	qtcsettings.cmake	15.10.2025 15:14	Файл "CMAKE"	1 КБ

Рисунок 2 – Папка “build” проєкту

### 1.4 Висновки

У результаті виконання лабораторної роботи №1 було створено консольний додаток у середовищі Qt Creator, який реалізує завдання варіанту №9. Програма успішно зчитує два текстові файли (file1.txt і file2.txt), обчислює перетин і різницю множин рядків із підтримкою ігнорування регістру (опція -i), та записує результати у три файли: common.txt, only\_first.txt, only\_second.txt.

Використано класи Qt (QCoreApplication, QFile, QTextStream, QSet, QHash) для обробки аргументів командного рядка, роботи з файлами та ефективного виконання операцій над множинами. Усунуто попередження Clazy шляхом заміни діапазонних циклів на const-ітератори, що підвищило ефективність роботи з QSet. Кодування UTF-8 забезпечило коректну обробку Unicode-рядків.

Програма протестована з різними входами, включаючи режими з ігноруванням та без ігнорування регістру, що підтвердило її коректність. Отримано навички створення консольних застосунків, обробки файлів, використання контейнерів Qt і діагностики за допомогою qInfo, qWarning, qCritical. Робота відповідає поставленій меті та вимогам завдання.