

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний аерокосмічний університет ім. М. Є. Жуковського
«Харківський авіаційний інститут»

Факультет систем управління літальних апаратів
Кафедра систем управління літальних апаратів

Лабораторна робота № 11

з дисципліни «Алгоритмізація та програмування»
на тему "Розробка десктоп-застосунків в середовищі QtCreator"

XAI.301. 174. 319. 2 ЛР

Виконав студент гр. 319

17.05.2025

(підпис, дата)

Дворнік І.П

(П.І.Б.)

Перевірів

_____ к.т.н., доц. Олена ГАВРИЛЕНКО

(підпис, дата)

(П.І.Б.)

МЕТА РОБОТИ

Ознайомитися з основами розробки програм з використанням QtDesigner і навчитися розробляти десктоп-застосунки із графічним користувацьким інтерфейсом для введення/виведення даних на мові програмування C++ в середовищі QtCreator.

ПОСТАНОВКА ЗАДАЧІ

Завдання 1. Вивчити алгоритм створення проекту Qt Widgets Application в середовищі QtCreator. Ознайомитись з налаштуваннями основних елементів для введення, виведення, компоновки форми і управління.

Завдання 2. Для вирішення завдання:

№17(табл.1). З початку доби минуло N секунд (N – ціле). Знайти кількість секунд, що пройшли з початку останньої години.

А. Спроекувати і реалізувати в конструкторі форм графічний інтерфейс програми з віджетами QLabel, QLineEdit і QPushButton. *Використати додаткові віджети.

В. Додати і відлагодити програмний код для введення вхідних даних з перевіркою на коректність (використати QMessageBox для виведення сповіщень), обчислень і виведення результатів.

С*. Додати пункти меню у QMenuBar для зчитування вхідних даних і збереження результатів в файл з використанням стандартних діалогів для вибору файла.

ВИКОНАННЯ РОБОТИ

Завдання 1

Дані для зачачі:

Віджети:

QCommandLinkButton - Кнопка з поясненням функцій

QTabWidget - Вкладки для структуризації інтерфейсу

QKeySequenceEdit - Вибір комбінації клавіш

QUndoView - Вивід історії дій

QGraphicsView - Відображення графічної сцени

QVideoWidget - Віджет для відтворення відео (підключається до

QMediaPlayer)

Вихідні дані:

Не знаю

Алгоритм вирішення:

Не знаю завдання не виконував

Завдання 2

Вирішення задачі №17(табл.1).

Вхідні дані:

Кількість секунд – додатні цілі числа, ліміт 86400.

Файл формату .txt – з додатнім цілим числом, ліміт 86400.

Вихідні дані:

Повідомлення про кількість секунд – додатні цілі числа, ліміт 3599.

Файл з повідомленням про кількість секунд – з додатнім цілим числом, ліміт 3599.

Алгоритм вирішення

1. Вивід вікна з проханням ввести кількість секунд, або завантажити з файлу формату .txt
2. Введення/завантаження.
3. Вивід результату
4. Результат можна зберегти у файл наприклад у заздалегіть заготовлений .txt файл

Лістинг коду вирішення задач №17(табл.1). наведено в дод. А(стор. 4-8).

Екран роботи програм показаний у дод. Б(стор. 9)

ВИСНОВКИ

Було вивчено введення/виведення даних на мові програмування C++ в середовищі QtCreator.

Отримано навички роботи з QtCreator.

Виникли труднощі з структуруванням та розумінням коду.

ДОДАТОК А

Лістинг коду програми

```
// file: main.cpp
#include <QApplication>
#include "mainwindow.h"

int main(int argc, char *argv[])
{
    QApplication app(argc, argv);

    MainWindow window;
    window.setWindowTitle("Seconds Calculator");
    window.show();

    return app.exec();
}

// file: mainwindow.h
#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>
#include <QLabel>
#include <QLineEdit>
#include <QPushButton>
#include <QMessageBox>
#include <QMenuBar>
#include <QMenu>
#include <QFileDialog>

class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    MainWindow(QWidget *parent = nullptr);

private slots:
    void calculateSeconds();           // Обчислює результат
    void loadFromFile();               // Зчитування з файлу
    void saveToFile();                 // Збереження результату

private:
    QLineEdit *inputEdit;
    QLabel *resultLabel;
```

```

};

#endif // MAINWINDOW_H

// file: mainwindow.cpp
#include "mainwindow.h"
#include <QVBoxLayout>
#include <QWidget>
#include <QSpinBox>
#include <QTextStream>
#include <QSlider>
#include <QCommandLinkButton>

// Конструктор основного вікна
MainWindow::MainWindow(QWidget *parent)
    : QMainWindow(parent)
{
    QWidget *central = new QWidget(this);
    QVBoxLayout *mainLayout = new QVBoxLayout(central);

    // Вкладки
    QTabWidget *tabs = new QTabWidget(this);
    QWidget *manualTab = new QWidget(this);
    QWidget *fileTab = new QWidget(this);

    QVBoxLayout *manualLayout = new QVBoxLayout(manualTab);

    // Поле для введення (QSpinBox)
    QSpinBox *secondsInput = new QSpinBox(this);
    secondsInput->setRange(0, 86400);
    secondsInput->setSuffix(" сек");
    manualLayout->addWidget(new QLabel("Введіть кількість секунд:", this));
    manualLayout->addWidget(secondsInput);

    // Альтернативний повзунок
    QSlider *slider = new QSlider(Qt::Horizontal, this);
    slider->setRange(0, 86400);
    manualLayout->addWidget(slider);

    // Синхронізація слайдера і spinbox
    connect(slider, &QSlider::valueChanged, secondsInput, &QSpinBox::setValue);
    connect(secondsInput, QOverload<int>::of(&QSpinBox::valueChanged), slider,
        &QSlider::setValue);

    // Кнопка обчислення
    QPushButton *calcButton = new QPushButton("Обчислити", this);
    manualLayout->addWidget(calcButton);

    // Результат
    resultLabel = new QLabel("Результат з'явиться тут.", this);

```

```

manualLayout->addWidget(resultLabel);

// Кнопка-пояснення
QCommandLinkButton *infoBtn = new QCommandLinkButton("Що робить програма?",
"Натисни для пояснення");
connect(infoBtn, &QCommandLinkButton::clicked, this, []() {
    QMessageBox::information(nullptr, "Інформація",
        "Програма обчислює, скільки секунд пройшло
        після останньої повної години, враховуючи
        кількість секунд від початку доби.");
});
manualLayout->addWidget(infoBtn);

// Таб «введення з файлу»
QVBoxLayout *fileLayout = new QVBoxLayout(fileTab);
QPushButton *loadBtn = new QPushButton("Зчитати з файлу формату .txt",
this);
fileLayout->addWidget(loadBtn);

// Додаємо вкладки
tabs->addTab(manualTab, "Ручне введення");
tabs->addTab(fileTab, "З файлу");

mainLayout->addWidget(tabs);
setCentralWidget(central);

// Підключення обробників
connect(calcButton, &QPushButton::clicked, this, [=]() {
    int N = secondsInput->value();
    int result = N % 3600;
    resultLabel->setText(QString("Після останньої години пройшло %1
секунд.").arg(result));
});

connect(loadBtn, &QPushButton::clicked, this, &MainWindow::loadFromFile);

// Меню
QMenu *fileMenu = menuBar()->addMenu("Файл");
fileMenu->addAction("Зберегти у файл", this, &MainWindow::saveToFile);
}

// Метод обчислення кількості секунд після останньої години
void MainWindow::calculateSeconds()
{
    bool ok;
    int N = inputEdit->text().toInt(&ok);

    if (!ok || N < 0) {
        QMessageBox::warning(this, "Помилка", "Будь ласка, введіть коректне
невід'ємне ціле число.");
    }
}

```

```

        return;
    }

    int result = N % 3600; // Кількість секунд після останньої години

    resultLabel->setText(QString("Після останньої години пройшло %1
    секунд.").arg(result));
}

// Метод зчитування значення з файлу
void MainWindow::loadFromFile()
{
    QString fileName = QFileDialog::getOpenFileName(this, "Виберіть вхідний
    файл");
    if (fileName.isEmpty()) return;

    QFile file(fileName);
    if (!file.open(QIODevice::ReadOnly | QIODevice::Text)) {
        QMessageBox::critical(this, "Помилка", "Не вдалося відкрити файл.");
        return;
    }

    QTextStream in(&file);
    QString line = in.readLine();
    file.close();

    // Спроба конвертувати рядок у число
    bool ok;
    int N = line.toInt(&ok);

    if (!ok || N < 0 || N > 86400) {
        QMessageBox::warning(this, "Помилка", "Файл містить некоректне
        значення.");
        return;
    }

    // Якщо використовуєш QSpinBox:
    if (auto spin = this->findChild<QSpinBox*>()) {
        spin->setValue(N); // Встановлюємо значення в полі
    }

    // Автоматично обчислюємо результат
    int result = N % 3600;
    resultLabel->setText(QString("Після останньої години пройшло %1
    секунд.").arg(result));
    // Повідомлення для користувача
    QMessageBox::information(this, "Зчитування завершено",
        "Значення успішно зчитано з файлу.\n"
        "Перейдіть на вкладку \"Ручне введення\", щоб
        побачити результат.");
}

```

```
}

// Метод збереження результату в файл
void MainWindow::saveToFile()
{
    QString fileName = QFileDialog::getSaveFileName(this, "Зберегти результат у файл");
    if (fileName.isEmpty()) return;

    QFile file(fileName);
    if (!file.open(QIODevice::WriteOnly | QIODevice::Text)) {
        QMessageBox::critical(this, "Помилка", "Не вдалося зберегти файл.");
        return;
    }

    QTextStream out(&file);
    out << resultLabel->text();
    file.close();
}
```


ДОДАТОК Б

Скрін-шоти вікна виконання програми

