

**Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки**

Лабораторна робота №1

з дисципліни
«Об'єктно-орієнтоване програмування»

Виконав:
студент групи ІМ-42
Туров Андрій Володимирович
номер у списку групи: 26

Перевірів:
Порєв В. М.

Київ 2025

Варіант 26

Завдання 2 + 3:

2. Два вікна діалогу. Спочатку з'являється перше, яке має дві кнопки: [Далі >] і [Відміна]. Якщо натиснути кнопку [Далі >], то воно закриється і з'явиться друге діалогове вікно, яке має кнопки: [< Назад], [Так] і [Відміна]. Якщо натиснути кнопку [<Назад], вікно закриється і перехід до першого вікна.
3. Вікно діалогу з елементом списку (List Box) та двома кнопками: [Так] і [Відміна]. У список автоматично записуються назви груп нашого факультету. Якщо вибрати потрібний рядок списку і натиснути [Так], то у головному вікні повинен відображатися текст вибраного рядка списку.

Текст програм

```
#include "action1.h"
#include "action2.h"
#include <qaction.h>
#include <qapplication.h>
#include <qlabel.h>
#include <qmainwindow.h>
#include <qmenubar.h>
#include <qnamespace.h>
#include <qobject.h>
#include <qstatusbar.h>
#include <qtmetamacros.h>

#define WINWIDTH 360
#define WINHEIGHT 120

class MainWindow final : public QMainWindow {
    Q_OBJECT
private:
    QLabel *label = new QLabel("Лабораторна робота №1");

public:
    MainWindow() {
        auto *menu = menuBar()->addMenu("&Меню");
        auto *action1 = menu->addAction("Робота &1");
        auto *action2 = menu->addAction("Робота &2");

        connect(action1, &QAction::triggered, this,
            ↳ action1_callback);
        connect(action2, &QAction::triggered, this, [this] {
            QString chosen = action2_callback();
            label->setText(chosen);
        });
    }
};
```

```

        label->setAlignment(Qt::AlignCenter);
        auto font = label->font();
        font.setPointSize(20);
        font.setBold(true);
        label->setFont(font);

        setCentralWidget(label);
    }
};

auto main(int argc, char *argv[]) -> int {
    QApplication app(argc, argv);
    MainWindow win;
    win.resize(WINWIDTH, WINHEIGHT);
    win.show();

    return QApplication::exec();
}

#include "main.moc"

```

Файл 1: src/main.cpp

```

#pragma once

#include <qdialog.h>
#include <qwidget.h>

class FirstDialog final : public QDialog {
    Q_OBJECT
public:
    FirstDialog(QWidget *parent = nullptr);
};

```

Файл 2: src/action1_diag1.h

```

#include "action1_diag1.h"
#include <qboxlayout.h>
#include <qlabel.h>
#include <qpushbutton.h>

FirstDialog::FirstDialog(QWidget *parent) : QDialog(parent) {
    setWindowTitle("Крок 1");

    auto *vbox = new QVBoxLayout(this);
    auto *label = new QLabel("Перше діалогове вікно", this);
    vbox->addWidget(label);

    auto *hbox = new QHBoxLayout();
    auto *btnNext = new QPushButton("Далі >", this);

```

```

auto *btnCancel = new QPushButton("Відміна", this);
hbox->addStretch();
hbox->addWidget(btnNext);
hbox->addWidget(btnCancel);
vbox->addLayout(hbox);

connect(btnNext, &QPushButton::clicked, this,
    ↳ &QDialog::accept);
connect(btnCancel, &QPushButton::clicked, this,
    ↳ &QDialog::reject);
}

```

Файл 3: src/action1_diag1.cpp

```

#pragma once

#include <qdialog.h>
#include <qwidget.h>

class SecondDialog final : public QDialog {
    Q_OBJECT
public:
    SecondDialog(QWidget *parent = nullptr);
    const static int kBackCode = 1001;
};

```

Файл 4: src/action1_diag2.h

```

#include "action1_diag2.h"
#include <qboxlayout.h>
#include <qdialog.h>
#include <qlabel.h>
#include <qpushbutton.h>
#include <qwidget.h>

SecondDialog::SecondDialog(QWidget *parent) : QDialog(parent) {
    setWindowTitle("Крок 2");

    auto *vbox = new QVBoxLayout(this);
    auto *label = new QLabel("Друге діалогове вікно", this);
    vbox->addWidget(label);

    auto *hbox = new QHBoxLayout();
    auto *btnBack = new QPushButton("< Назад", this);
    auto *btnYes = new QPushButton("Так", this);
    auto *btnCancel = new QPushButton("Відміна", this);
    hbox->addStretch();
    hbox->addWidget(btnBack);
    hbox->addWidget(btnYes);
}

```

```

hbox->addWidget(btnCancel);
vbox->addLayout(hbox);

connect(btnBack, &QPushButton::clicked, this, [this]() {
    ↪ done(kBackCode); });
connect(btnYes, &QPushButton::clicked, this, &QDialog::accept);
connect(btnCancel, &QPushButton::clicked, this,
    ↪ &QDialog::reject);
}

```

Файл 5: src/action1_diag2.cpp

```

#pragma once

extern void action1_callback(bool checked = false);

```

Файл 6: src/action1.h

```

#include "action1.h"
#include "action1_diag1.h"
#include "action1_diag2.h"

#include <QHBoxLayout>
#include <QVBoxLayout>
#include <qapplication.h>
#include <qdialog.h>
#include <qlabel.h>
#include <qpointer.h>
#include <qpushbutton.h>
#include <qstring.h>

static auto run(QWidget *parent) -> bool {
    for (;;) {
        FirstDialog first(parent);
        if (first.exec() != QDialog::Accepted) {
            return false;
        }

        SecondDialog second(parent);
        const int result = second.exec();
        if (result == QDialog::Accepted) {
            return true;
        }
        if (result == SecondDialog::kBackCode) {
            continue;
        }
        return false;
    }
}

```

```
void action1_callback(bool checked) {
    QPointer<QWidget> parent = QApplication::activeWindow();
    (void)run(parent.data());
}
```

Файл 7: src/action1.cpp

```
#pragma once

#include <qobject.h>
extern auto action2_callback(bool checked = false) -> QString;
```

Файл 8: src/action2.h

```
#include "action2.h"

#include <QHBoxLayout>
#include <QVBoxLayout>
#include <qapplication.h>
#include <qdialog.h>
#include <qkeysequence.h>
#include <qlabel.h>
#include <qlistwidget.h>
#include <qmainwindow.h>
#include <qpointer.h>
#include <qpushbutton.h>
#include <qstatusbar.h>
#include <qstringlist.h>

class GroupListDialog final : public QDialog {
    Q_OBJECT
public:
    explicit GroupListDialog(QWidget *parent = nullptr) :
        QDialog(parent) {
        setWindowTitle("Вибір групи");

        auto *vbox = new QVBoxLayout(this);
        auto *label = new QLabel("Оберіть групу факультету:", this);
        vbox->addWidget(label);

        list = new QListWidget(this);
        list->setSelectionMode(QAbstractItemView::SingleSelection);
        vbox->addWidget(list);

        const QStringList groups = {"IM-41", "IM-42", "IM-43",
            "IM-44", "IC-43"};
        list->addItem(groups);
    }
};
```

```

    auto *hbox = new QHBoxLayout();
    btn_yes = new QPushButton("Так", this);
    auto *btn_cancel = new QPushButton("Відміна", this);
    btn_yes->setEnabled(false);

    hbox->addStretch();
    hbox->addWidget(btn_yes);
    hbox->addWidget(btn_cancel);
    vbox->addLayout(hbox);

    connect(list, &QListWidget::currentRowChanged, this,
            [this](int row) { btn_yes->setEnabled(row >= 0); });
    connect(btn_yes, &QPushButton::clicked, this,
            ↪ &QDialog::accept);
    connect(btn_cancel, &QPushButton::clicked, this,
            ↪ &QDialog::reject);
}

QString selectedGroup() const {
    if (auto *item = list->currentItem())
        return item->text();
    return {};
}

private:
    QListWidget *list = nullptr;
    QPushButton *btn_yes = nullptr;
};

auto action2_callback(bool) -> QString {
    QPointer<QWidget> mainWin = QApplication::activeWindow();

    GroupListDialog dlg(mainWin.data());
    if (dlg.exec() == QDialog::Accepted) {
        const QString chosen = dlg.selectedGroup();
        return "Вибрано групу: " + chosen;
    }
    return "Лабораторна робота №1";
}

#include "action2.moc"

```

Файл 9: src/action2.cpp

```

cmake_minimum_required(VERSION 3.16)
project(lab1 LANGUAGES CXX)

set(CMAKE_CXX_STANDARD 17)
set(CMAKE_CXX_STANDARD_REQUIRED ON)

```

```

find_package(Qt6 REQUIRED COMPONENTS Core Widgets)

qt_standard_project_setup()
qt_add_executable(lab1
  src/main.cpp
  src/action1.cpp
  src/action1_diag1.cpp
  src/action1_diag1.h
  src/action1_diag2.cpp
  src/action1_diag2.h
  src/action2.cpp
)

target_link_libraries(lab1 PRIVATE Qt6::Core Qt6::Widgets)

add_custom_target(run
  COMMAND $<TARGET_FILE:lab1>
  DEPENDS lab1
  WORKING_DIRECTORY $<TARGET_FILE_DIR:lab1>
)

```

Файл 10: CMakeLists.txt

Зображення

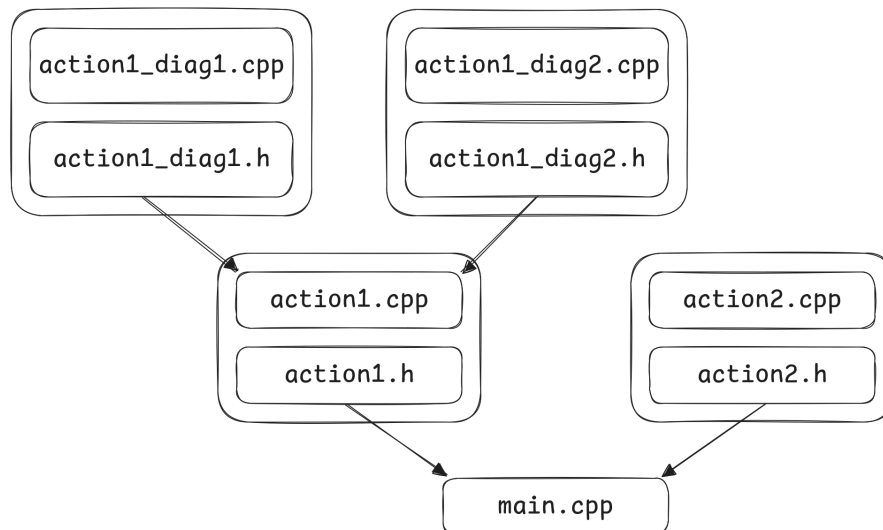


Рис. 1: #include-ієрархія

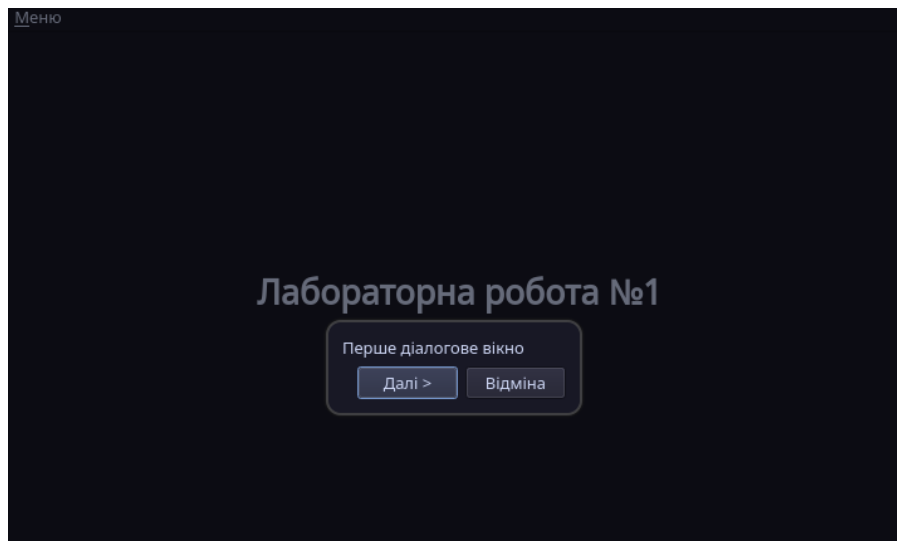


Рис. 2: Робота 1: Перший діалог

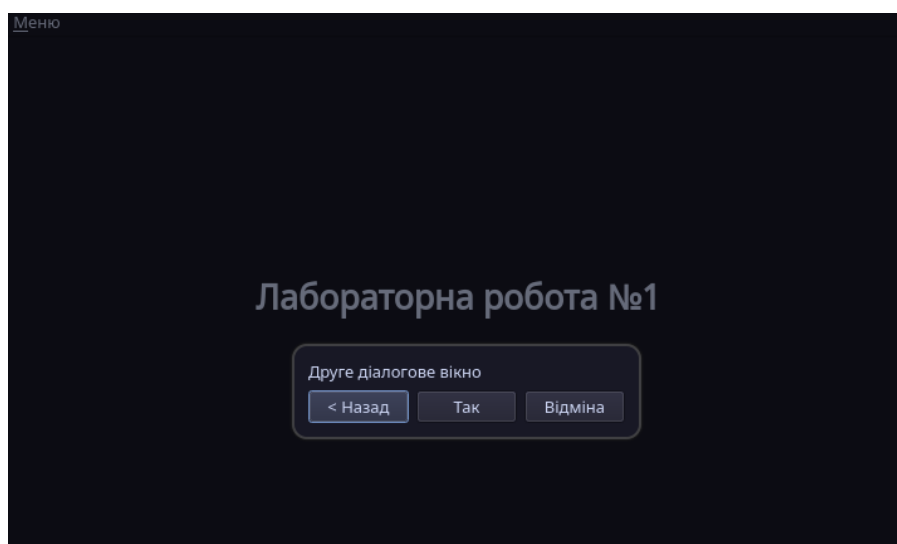


Рис. 3: Робота 1: Другий діалог

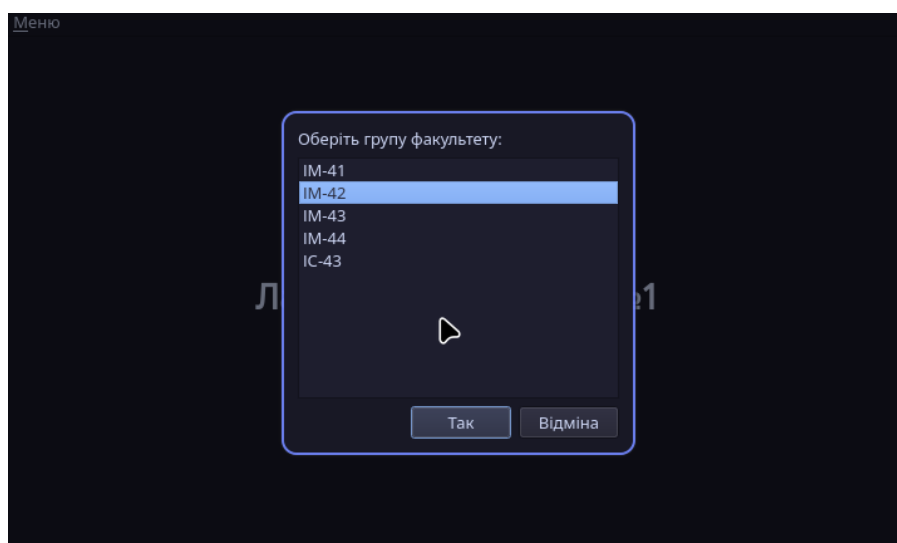


Рис. 4: Робота 2: Діалог

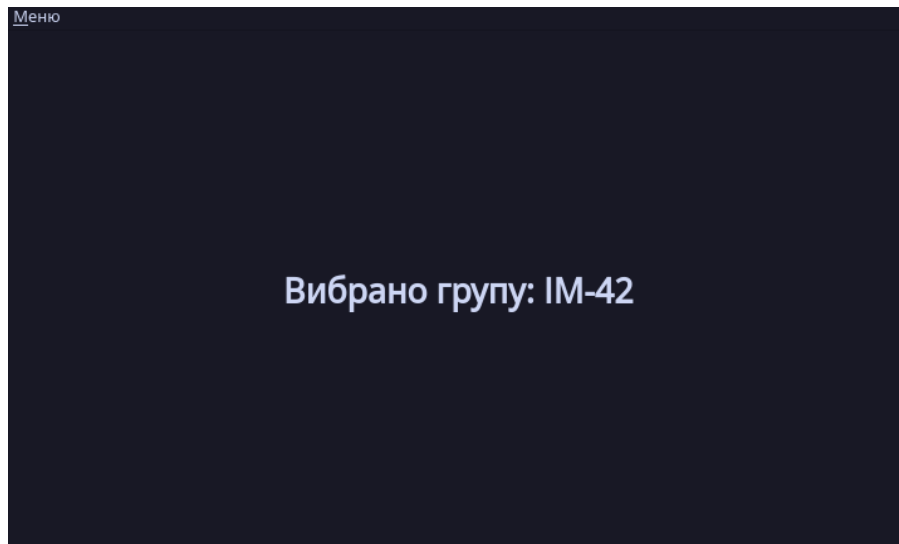


Рис. 5: Робота 2: Результат

Висновок

Написав графічний застосунок за допомогою фреймворку Qt6. Розділив логіку виконання роботи на модулі так, що кожен модуль містить певну логіку відображення одного діалогового вікна (або групи вікон). Водночас, переконався, щоб дерево залежностей було направлене строго у напрямку до головного виконувального файлу.