

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний аерокосмічний університет
«Харківський авіаційний інститут»

Факультет систем управління літальних апаратів
Кафедра систем управління літальних апаратів

Лабораторна робота № 11

з дисципліни «Алгоритмізація та програмування»

Тема: "Розробка десктоп-застосунків в середовищі QtCreator"

ХАІ.301. 174. 319. 12 ЛР

Виконав студент гр. 319

(підпис, дата) Віталій НОВИКОВ
(П.І.Б.)

Перевірів

(підпис, дата) к.т.н., доц. Олена ГАВРИЛЕНКО
(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. Відповідно до завдання (дан діаметр окружності d . Знайти її довжину $L = \pi \cdot d$. Значення π вважати рівним 3.14), реалізувати:

А. Спроектувати і реалізувати в конструкторі форм графічний інтерфейс програми з віджетами QLabel, QLineEdit і QPushButton.

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.

Загалом побудувати графічний інтерфейс із використанням QLabel, QLineEdit, QPushButton, QGridLayout, QMenuBar.

ВИКОНАННЯ РОБОТИ

Інтерфейс програми:

Інтерфейс реалізований у QtDesigner та включає наступні елементи:

№	Віджет	Ім'я (objectName)	Призначення
1	QLabel	label_d	Підпис: "Введіть діаметр:"
2	QLineEdit	le_d	Введення значення діаметра
3	QPushButton	pushButton_calc	Кнопка для обчислення довжини
4	QLabel	label_result	Підпис: "Довжина кола:"
5	QLabel	l_result	Виведення результату
6	QMenuBar	menuFile	Меню "Меню"
7	QAction	actionLoad	Пункт меню "Load"
8	QAction	actionSave	Пункт меню "Save"

Опис алгоритму створення проекту в QtCreator:

1. Запуск QtCreator → File → New Project → Qt Widgets Application.
2. Вказання ім'я проекту (CircleLengthApp), директорію збереження.
3. Вибір компілятора (CMake).
4. У вікні mainwindow.ui побудува форми за допомогою QtDesigner.
5. Задавання objectName для кожного віджета.
6. Додавання обробника сигналу clicked() для кнопки та triggered() для елементів меню.
7. У mainwindow.cpp реалізація програмної логіки.
8. Запуск компіляції (Ctrl+B) та запуск (Ctrl+R).

Результати роботи:

- Програма дозволяє користувачу ввести діаметр окружності.
- Після натискання кнопки "Обчислити", програма обчислює довжину окружності та виводить результат з точністю до 2 знаків після коми.
- Передбачено повідомлення про помилки при неправильному введенні.
- Додано можливість зчитування даних з файлу та збереження результату у файл через меню Файл.

Лістинг коду наведено в додатку А (стор. 4).

Екран роботи програми показаний в додатку Б, рис. Б.1(стор.7).

Всі елементи меню показані в додатку Б, рис. Б.2(стор.8).

ВИСНОВКИ

У ході виконання лабораторної роботи була розроблена програма на мові C++ із використанням середовища QtCreator, що реалізує графічний інтерфейс для введення, обробки та виведення даних. У процесі виконання були закріплені знання щодо створення віджетів, обробки сигналів і слотів, а також роботи з файлами. Загалом, робота сприяла поглибленню знань з програмування, розвитку алгоритмічного мислення та формуванню навичок побудови зручного і функціонального графічного інтерфейсу користувача.

ДОДАТОК А

Лістинг коду програми

main.cpp

```
#include "mainwindow.h"

#include <QApplication>

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    MainWindow w;
    w.show();
    return a.exec();
}
```

mainwindow.cpp

```
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include <QMessageBox>    // Для показу повідомлень (помилки, попередження)
#include <QFileDialog>    // Для виклику діалогових вікон відкриття/збереження файлів
#include <QFile>          // Для роботи з файлами
#include <QTextStream>    // Для читання/запису тексту з/до файлу

MainWindow::MainWindow(QWidget *parent)
    : QMainWindow(parent)
    , ui(new Ui::MainWindow)
{
    // Ініціалізація інтерфейсу
    ui->setupUi(this);

    // Підключення пунктів меню до відповідних слотів
    connect(ui->actionLoad_from_file, &QAction::triggered,
            this, &MainWindow::on_actionLoad_from_file_triggered);

    connect(ui->actionSave_to_file, &QAction::triggered,
            this, &MainWindow::on_actionSave_to_file_triggered);
}

MainWindow::~MainWindow()
{
    delete ui; // Очищення виділеної пам'яті
}

// Слот для обробки натискання кнопки "Обчислити"
void MainWindow::on_pushButton_calc_clicked()
{
    float d, L;    // Змінні для діаметра та довжини кола
    bool ok;       // Прапорць успішного перетворення введення
13
    // Зчитування значення з поля вводу le_d
    d = ui->le_d->text().toFloat(&ok);

    if (ok && d > 0) {
        // Обчислення довжини кола ( $L = \pi * d$ ), де  $\pi \approx 3.14$ 
        L = 3.14 * d;

        // Виведення результату у мітку l_result
        ui->l_result->setText(QString::number(L));
    }
}
```

```

else {
    // Повідомлення про помилку у введенні
    QMessageBox::warning(this, "Помилка", "Некоректне значення діаметра!");

    // Очистити поле введення і результат
    ui->le_d->clear();
    ui->l_result->setText("0");

    // Повернути фокус на поле введення
    ui->le_d->setFocus();
}
}

// Слот для завантаження даних з файлу
void MainWindow::on_actionLoad_from_file_triggered()
{
    // Відкриття діалогового вікна для вибору файлу
    QString filePath = QFileDialog::getOpenFileName(this, "Відкрити файл",
                                                    QDir::currentPath(),
                                                    "Текстові файли (*.txt);;Усі файли (*)");

    if (!filePath.isEmpty()) {
        QFile file(filePath);

        // Перевірка чи відкрився файл
        if (!file.open(QIODevice::ReadOnly | QIODevice::Text)) {
            QMessageBox::warning(this, "Помилка", "Не вдалося відкрити файл для читання.");
            return;
        }

        // Створення текстового потоку для читання
        QTextStream in(&file);
        QString sd;
        in >> sd;

        // Запис зчитаного значення у поле введення
        ui->le_d->setText(sd);

        file.close(); // Закрити файл
    } else {
        // Якщо файл не було вибрано
        QMessageBox::information(this, "Увага", "Файл не вибрано.");
    }
}

// Слот для збереження результату у файл
void MainWindow::on_actionSave_to_file_triggered()
{
    // Відкриття діалогового вікна для збереження
    QString filePath = QFileDialog::getSaveFileName(this, "Зберегти файл",
                                                    QDir::currentPath(),
                                                    "Текстові файли (*.txt);;Усі файли (*)");

    if (!filePath.isEmpty()) {
        QFile file(filePath);

        // Перевірка чи відкрився файл для запису
        if (!file.open(QIODevice::WriteOnly | QIODevice::Text)) {
            QMessageBox::critical(this, "Помилка",
                                  "Не вдалося відкрити файл:\n" + file.errorString());
            return;
        }

        // Створення текстового потоку для запису
        QTextStream out(&file);
    }
}

```

```
// Запис результату у файл
out << "Діаметр: " << ui->le_d->text() << "\n";
out << "Довжина кола: " << ui->l_result->text() << "\n";

file.close(); // Закрити файл
} else {
    // Якщо користувач скасував вибір файлу
    QMessageBox::information(this, "Увага", "Файл не вибрано.");
}
}
```

ДОДАТОК Б

На рис.Б.1 показано екран роботи програми.

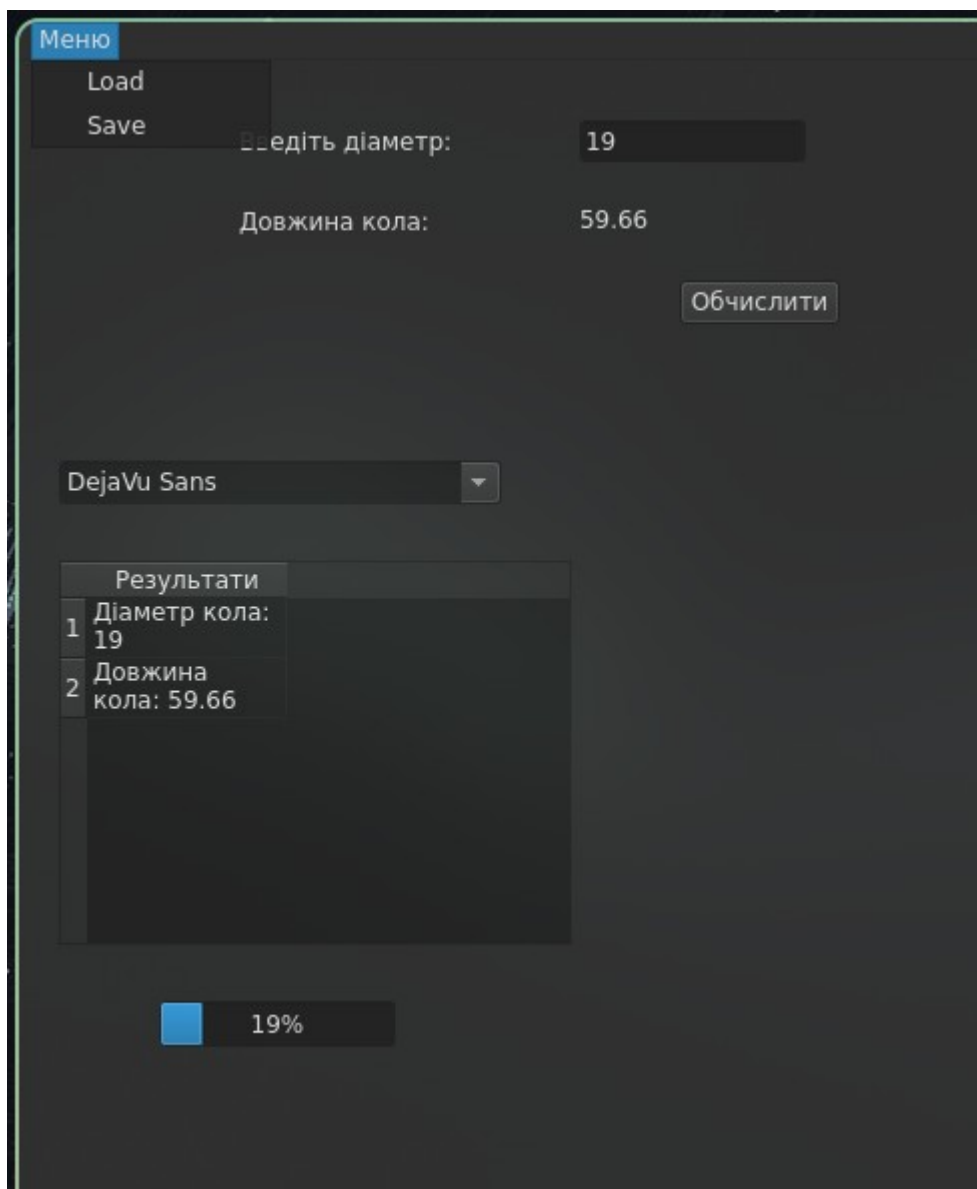


Рисунок Б.1 — Екран роботи програми.

На рис.Б.2 показано всі елементи програми.

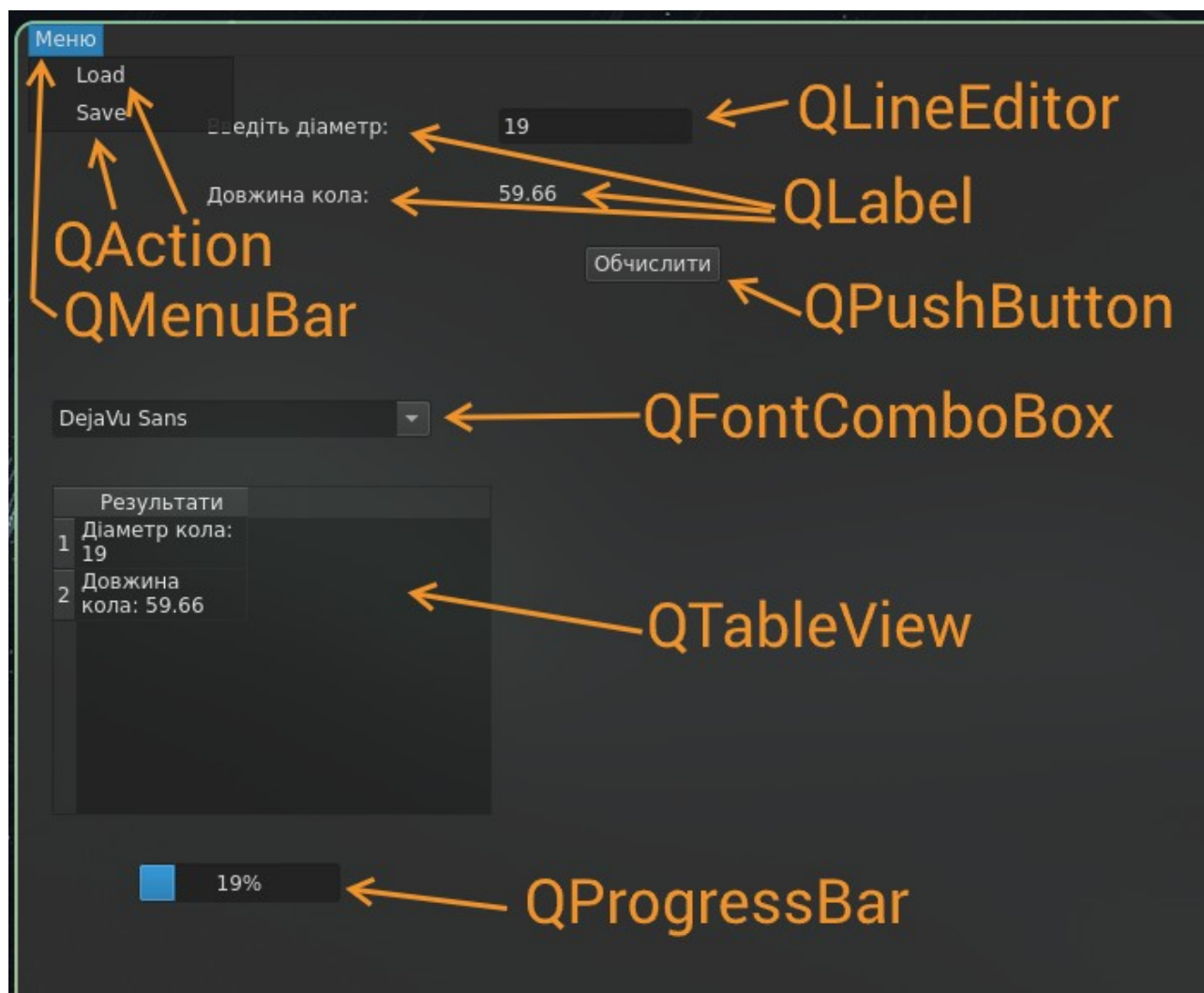


Рисунок Б.2 – Всі елементи програми.