МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

Інститут **КНІТ**
Кафедра **ПЗ**

ЗВІТ

До лабораторної роботи № 1-4
На тему: "*Середовище програмування*"
З дисципліни: "Об'єктно-орієнтоване програмування"

Лектор:
доцент кафедри ПЗ
Коротєєва Т.О.

Виконав:
студент групи ПЗ-16
Коваленко Д.М.

Прийняла:
доцент кафедри ПЗ
Яцишин С.І.

«_____» _____ 2022 р.
$\sum$ = _____.....

Львів — 2022

Тема. Середовище програмування. Створення проекту та налаштування його властивостей.

Мета. Засвоїти принцип візуального програмування шляхом створення та налаштування проекту.

Лабораторне завдання

1. Ознайомитись із середовищем QtCreator.
2. Створити новий проект. Зберегти його двома способами – через комбінації швидких клавіш та через меню.
3. Проглянути у вікні інспектора об'єктів властивості форми. Змінити назву форми та її розміри.
4. Запустити на виконання застосування.
5. Відкрити опції проекту, змінити налаштування на закладках Application, Compiler, Packages. Запустити на виконання застосування.
6. Реалізувати програму "Калькулятор";
7. Реалізувати програму "Редактор текстових файлів та переглядач фото"
8. Реалізувати програму "Гра"

Теоретичні відомості

Були використані такі компоненти:

Головне вікно (*MainWindow*):

- зміна назви вікна здійстнюється за допомогою властивості *windowTitle*;
- зміна розміру вікна здійснюється у вкладці *geometry* властивостями *Width* та *Height*;

Кнопка (*pushButton*):

- зміна тексту на кнопці здійстнюється за допомогою властивості *text*;
- натискання кнопки клавіатурою здійстнюється за допомогою властивості *shortcut*;

Напис (*label*):

- зміна тексту здійстнюється за допомогою властивості *text*;
- зміна розміщення тексту здійстнюється за допомогою властивості *alignment*;
- зробити напис невидимим можна за допомогою властивості *visible*;
- відобразити зображення на поверхні напису можна за допомогою властивості *pixmap*;

Поле для вводу (*lineEdit*):

- зміна тексту здійстнюється за допомогою властивості *text*;
- унеможливлює ввід тексту в поле з клавіатури властивість *readOnly=True*;
- дозволяє виділяти текст мишкою властивість *dragEnabled=True*;

Таблиця (*tableWidget*):

- приховати смуги прокрутки можна за допомогою властивості *verticalScrollbar=ScrollBarAlwaysOff* та *horizontalScrollbar=ScrollBarAlwaysOff*;
- змінити кількість рядків та стовпців можна за допомогою властивості *rowCount* та *columnCount*;
- дозволяє виділяти текст мишкою властивість *dragEnabled=True*;

Числове поле (*spinBox*):

- зазначтити числові межі можна за допомою властивостей *minimum* та *maximum*;

Меню (*MenuBar*):

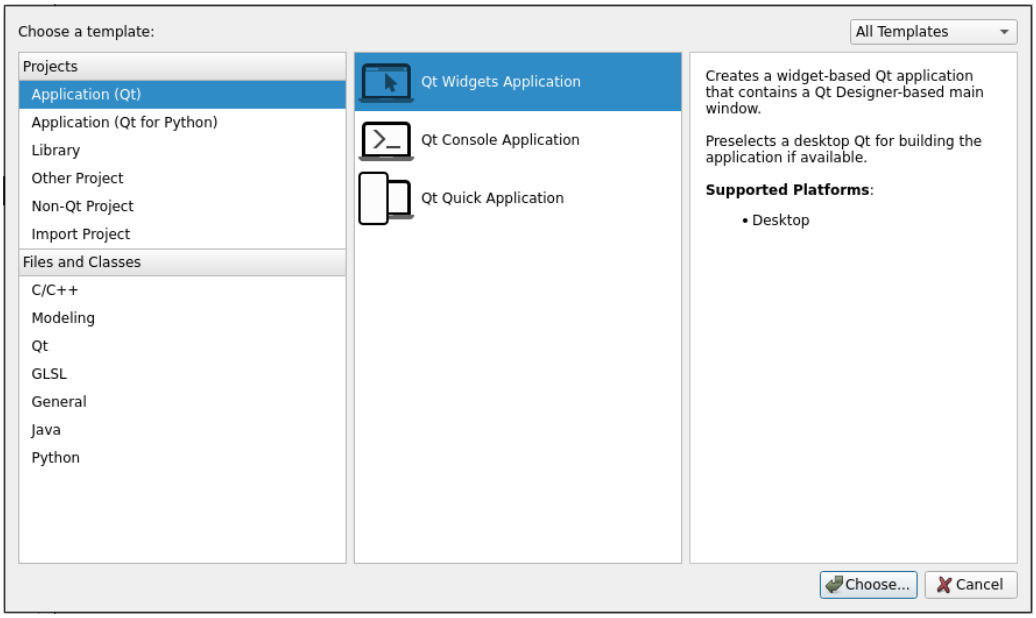
- відкрити пункт меню з клавіатури можна за допомою властивості *shortcut*;

Поле для вводу тексту (*textEdit*):

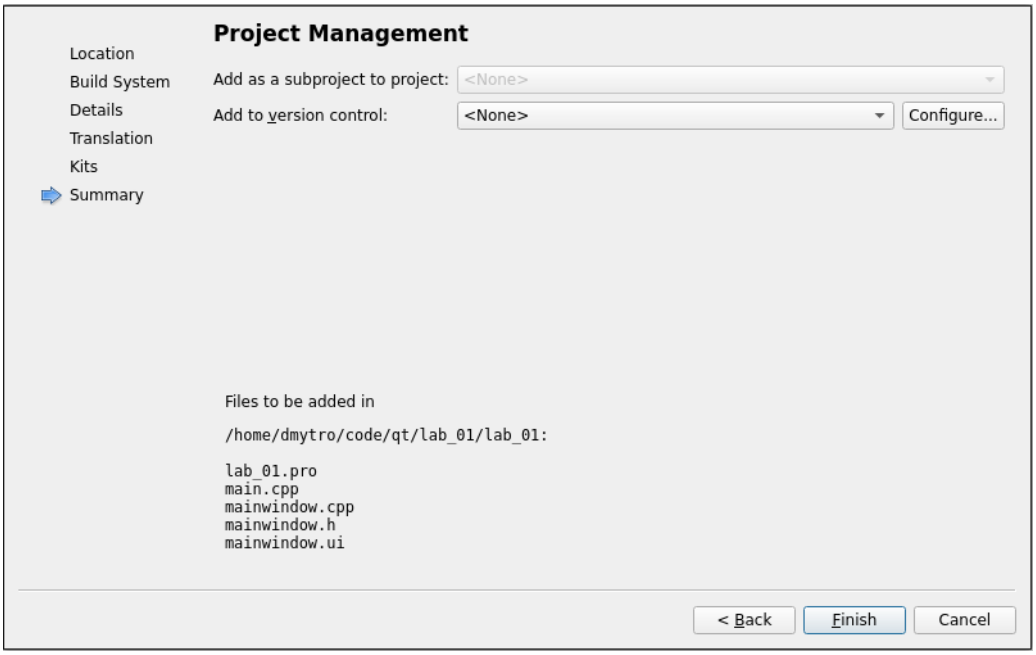
- зробити елемент недоступним можна за допомою властивості *enabled*;

Хід роботи

Для створення нового віконного проекту в середовищі *QtCreator* необхідно у головному меню обрати пункт *Projects > New* — вікриється вікно вибору шаблону (Рис. 1.а). Необхідно використати шаблон *Application (Qt)* та *Qt Widgets Application*. Наступне вікно дозволяє обрати шлях до майбутнього проекту, утиліту для збору проеку, назву головного класу програми, компілятор (Рис. 1.б).



(а)

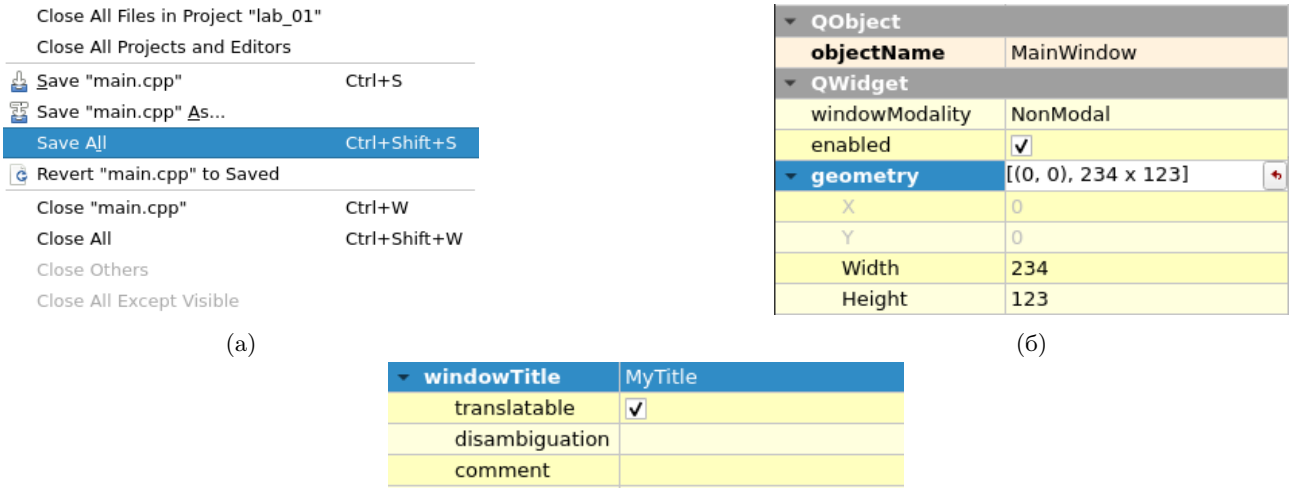


(б)

Рис. 1: Створення нового проекту.

Для збереження проекту можна скористатись комбінацією клавіш *Ctrl+Shift+S* або у меню *File* вибрати пункт *Save All* (Рис. 2.а).

Для зміни розміру та назви головного вікна необхідно вибрати властивості *geometry* та *windowTitle* об'єкту *MainWindow* (Рис. 2.б, 2.в).



(а) (б)

(в)

Рис. 2: Збереження нового проекту (а); Зміна розміру вікна (б); Зміна назви вікна (в).

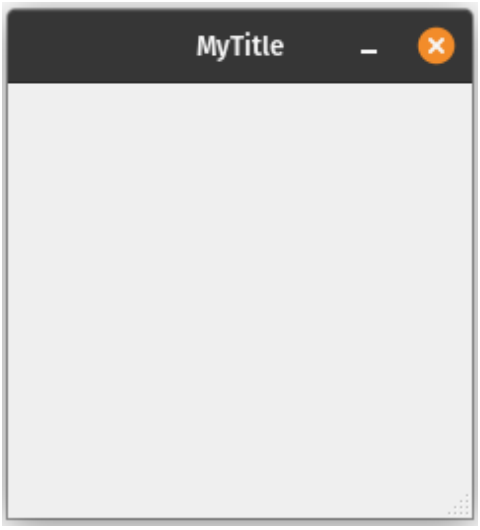


Рис. 3: Виконання застосунку.

Код програми "Калькулятор"

Назва файлу: *main.cpp*

```
#include "mainwindow.h"

#include <QApplication>

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

Назва файлу: *mainwindow.cpp*

```
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include "calculator.h"
```

```
#include "math.h"

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

MainWindow::~MainWindow() {
    delete ui;
}

Calculator calculator = Calculator();

void MainWindow::on_buttonClearAll_clicked() {
    ui->mainTextBox->setText("0");
    ui->debugLabel->setText("");
    calculator.reset();
}

void MainWindow::on_buttonClear_clicked() {
    if (
        ui->mainTextBox->text().length() == 1 ||
        ui->mainTextBox->text() == "Error"
    ) {
        ui->mainTextBox->setText("0");
    } else {
        QString text = ui->mainTextBox->text();
        text.chop(1);
        ui->mainTextBox->setText(text);
    }
}

void unaryOperation(
    Ui::MainWindow *ui,
    Calculator::operation op,
    std::string opName) {
    if (isnan(calculator.lValue))
        calculator.lValue = ui->mainTextBox->text().toDouble();
    calculator.op = op;
    calculator.calculate();
    if (isnan(calculator.result) || isinf(calculator.result)) {
        ui->mainTextBox->setText("Error");
        ui->debugLabel->setText("");
        return;
    }
    ui->mainTextBox->setText(QString::number(calculator.result));
    ui->debugLabel->setText(QString::fromStdString(opName) + "(" +
        QString::number(calculator.lValue) + ") = " +
        QString::number(calculator.result));
    calculator.reset();
}

void binaryOperation(Ui::MainWindow *ui,Calculator::operation op, std::
string opName)
{
    calculator.lValue = ui->mainTextBox->text().toDouble();
    calculator.op = op;
    ui->mainTextBox->setText("0");
    ui->debugLabel->setText(QString::number(calculator.lValue) +
        QString::fromStdString(opName));
```

```
}

void MainWindow::on_button1_clicked() {
    if (ui->mainTextBox->text() == "0") ui->mainTextBox->setText("1");
    else ui->mainTextBox->setText(ui->mainTextBox->text() + "1");
}

void MainWindow::on_button2_clicked() {
    if (ui->mainTextBox->text() == "0") ui->mainTextBox->setText("2");
    else ui->mainTextBox->setText(ui->mainTextBox->text() + "2");
}

void MainWindow::on_button3_clicked() {
    if (ui->mainTextBox->text() == "0") ui->mainTextBox->setText("3");
    else ui->mainTextBox->setText(ui->mainTextBox->text() + "3");
}

void MainWindow::on_button4_clicked() {
    if (ui->mainTextBox->text() == "0") ui->mainTextBox->setText("4");
    else ui->mainTextBox->setText(ui->mainTextBox->text() + "4");
}

void MainWindow::on_button5_clicked() {
    if (ui->mainTextBox->text() == "0") ui->mainTextBox->setText("5");
    else ui->mainTextBox->setText(ui->mainTextBox->text() + "5");
}

void MainWindow::on_button6_clicked() {
    if (ui->mainTextBox->text() == "0") ui->mainTextBox->setText("6");
    else ui->mainTextBox->setText(ui->mainTextBox->text() + "6");
}

void MainWindow::on_button7_clicked() {
    if (ui->mainTextBox->text() == "0") ui->mainTextBox->setText("7");
    else ui->mainTextBox->setText(ui->mainTextBox->text() + "7");
}

void MainWindow::on_button8_clicked() {
    if (ui->mainTextBox->text() == "0") ui->mainTextBox->setText("8");
    else ui->mainTextBox->setText(ui->mainTextBox->text() + "8");
}

void MainWindow::on_button9_clicked() {
    if (ui->mainTextBox->text() == "0") ui->mainTextBox->setText("9");
    else ui->mainTextBox->setText(ui->mainTextBox->text() + "9");
}

void MainWindow::on_button0_clicked() {
    if (ui->mainTextBox->text() == "0") ui->mainTextBox->setText("0");
    else ui->mainTextBox->setText(ui->mainTextBox->text() + "0");
}

void MainWindow::on_buttonDot_clicked() {
    if (!ui->mainTextBox->text().contains("."))
        ui->mainTextBox->setText(ui->mainTextBox->text() + ".");
}

void MainWindow::on_buttonEqual_clicked() {
    if (isnan(calculator.lValue)) return;
    calculator.rValue = ui->mainTextBox->text().toDouble();
    calculator.calculate();
```

```
    if (isnan(calculator.result) || isinf(calculator.result)) {
        ui->mainTextBox->setText("Error");
        ui->debugLabel->setText("");
        return;
    }
    ui->mainTextBox->setText(QString::number(calculator.result));
    ui->debugLabel->setText(ui->debugLabel->text() +
        QString::number(calculator.rValue)  + " = " +
        QString::number(calculator.result));
    calculator.reset();
}

void MainWindow::on_buttonDivide_clicked() {
    binaryOperation(ui, Calculator::operation::Divide, " / ");
}

void MainWindow::on_buttonMultiply_clicked() {
    binaryOperation(ui, Calculator::operation::Multiply, " * ");
}

void MainWindow::on_buttonSubstract_clicked() {
    if (ui->mainTextBox->text() == "0") {
        ui->mainTextBox->setText("-");
        return;
    }
    binaryOperation(ui, Calculator::operation::Substract, " - ");
}

void MainWindow::on_buttonAdd_clicked() {
    binaryOperation(ui, Calculator::operation::Add, " + ");
}

void MainWindow::on_buttonSqrt_clicked() {
    unaryOperation(ui, Calculator::operation::Sqrt, "sqrt");
}

void MainWindow::on_buttonMod_clicked() {
    binaryOperation(ui, Calculator::operation::Mod, " % ");
}

void MainWindow::on_buttonPow_clicked() {
    binaryOperation(ui, Calculator::operation::Pow, "^");
}

void MainWindow::on_buttonSin_clicked() {
    unaryOperation(ui, Calculator::operation::Sin, "sin");
}

void MainWindow::on_buttonCos_clicked() {
    unaryOperation(ui, Calculator::operation::Cos, "cos");
}

void MainWindow::on_buttonTan_clicked() {
    unaryOperation(ui, Calculator::operation::Tan, "tan");
}

void MainWindow::on_buttonPi_clicked() {
    ui->mainTextBox->setText(QString::number(calculator.PI));
}

void MainWindow::on_buttonE_clicked() {
```

```
    ui->mainTextBox->setText(QString::number(calculator.E));
}
```

Назва файлу: *calculator.cpp*

```
#include "calculator.h"

Calculator::Calculator() {}

void Calculator::calculate() {
    switch (this->op) {
        case Divide:
            this->result = this->lValue / this->rValue;
            break;
        case Multiply:
            this->result = this->lValue * this->rValue;
            break;
        case Substract:
            this->result = this->lValue - this->rValue;
            break;
        case Add:
            this->result = this->lValue + this->rValue;
            break;
        case Sqrt:
            this->result = sqrt(this->lValue);
            break;
        case Mod:
            this->result = fmod(this->lValue, this->rValue);
            break;
        case Pow:
            this->result = pow(this->lValue, this->rValue);
            break;
        case Sin:
            this->result = sin(this->lValue);
            break;
        case Cos:
            this->result = cos(this->lValue);
            break;
        case Tan:
            this->result = tan(this->lValue);
            break;
    };
}

void Calculator::reset() {
    this->lValue = NAN;
    this->rValue = NAN;
    this->result = NAN;
}
```

Назва файлу: *mainwindow.h*

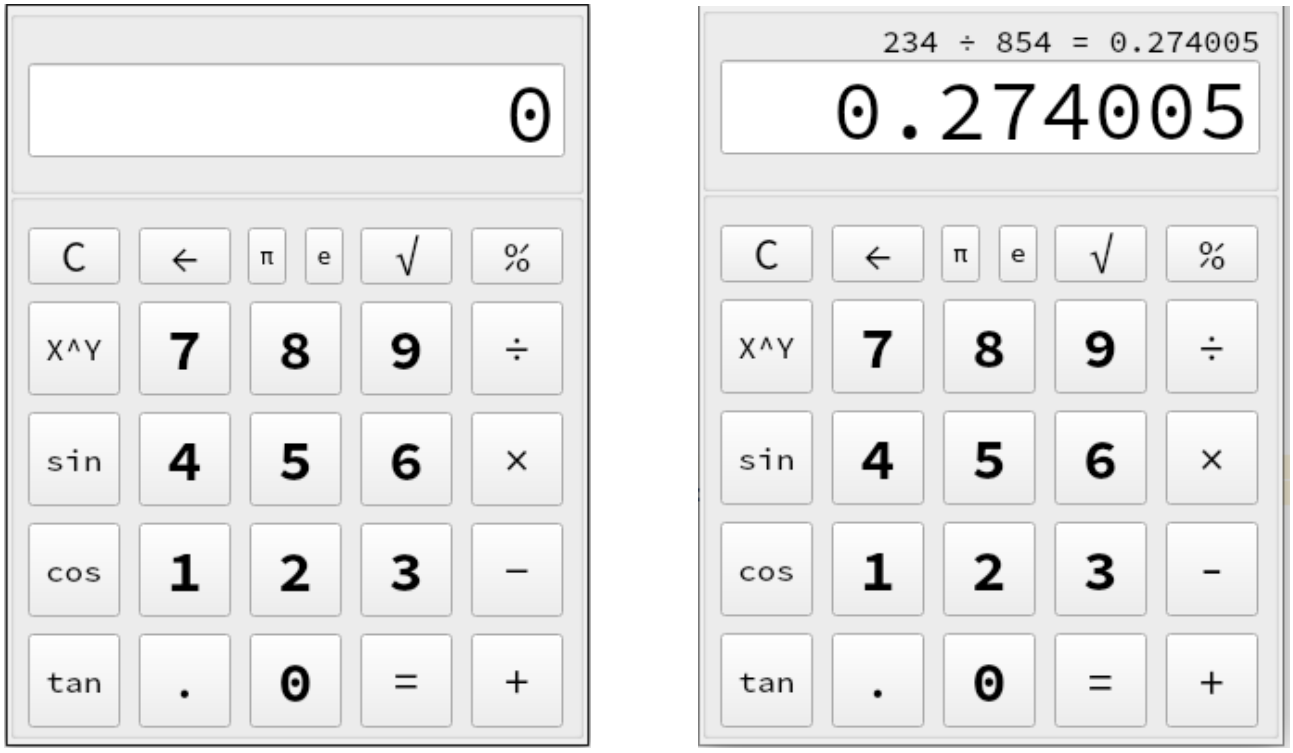
```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>

QT_BEGIN_NAMESPACE
namespace Ui { class MainWindow; }
QT_END_NAMESPACE
```

```
class MainWindow : public QMainWindow {
    Q_OBJECT

    public:
    MainWindow(QWidget *parent = nullptr);
    ~MainWindow();

    private slots:
        void on_buttonClearAll_clicked();
        void on_buttonClear_clicked();
        void on_buttonSubstract_clicked();
        void on_button1_clicked();
        void on_button2_clicked();
        void on_button3_clicked();
        void on_button4_clicked();
        void on_button5_clicked();
        void on_button6_clicked();
        void on_button7_clicked();
        void on_button8_clicked();
        void on_button9_clicked();
        void on_button0_clicked();
        void on_buttonDot_clicked();
        void on_buttonEqual_clicked();
        void on_buttonDivide_clicked();
        void on_buttonMultiply_clicked();
        void on_buttonAdd_clicked();
        void on_buttonSqrt_clicked();
        void on_buttonMod_clicked();
        void on_buttonPow_clicked();
        void on_buttonSin_clicked();
        void on_buttonCos_clicked();
        void on_buttonTan_clicked();
        void on_buttonPi_clicked();
        void on_buttonE_clicked();

    private:
        Ui::MainWindow *ui;
};
#endif // MAINWINDOW_H
```

Назва файлу: *calculator.h*

```
#ifndef CALCULATOR_H
#define CALCULATOR_H

#include <math.h>

class Calculator {
    public:
    enum operation {
        Divide,
        Multiply,
        Add,
        Substract,
        Sqrt,
        Mod,
        Pow,
        Sin,
        Cos,
        Tan,
```

```
    };
    Calculator();
    const double PI = M_PI;
    const double E = M_E;
    double lValue = NAN;
    double rValue = NAN;
    double result = NAN;
    operation op;
    void calculate();
    void reset();
};

#endif // CALCULATOR_H
```



(а) (б)

Рис. 4: Вигляд програми (а);
Робота програми (б).

Код програми "Редактор файлів та переглядач фото"

Назва файлу: *main.cpp*

```
#include "mainwindow.h"

#include <QApplication>

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

Назва файлу: *mainwindow.cpp*

```
#include "mainwindow.h"
#include "ui_mainwindow.h"

#include <QColorDialog>
#include <QFontDialog>
#include <QInputDialog>
#include <QPrintDialog>
#include <QSaveFile>
#include <QFile>
#include <QFileDialog>
#include <QImage>
#include <QImageReader>
#include <iostream>
#include <fstream>
#include <string>
using namespace std;

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

MainWindow::~MainWindow() {
    delete ui;
}

void MainWindow::on_actionNew_triggered() {
    ui->textEdit->setEnabled(true);
    ui->textEdit->setVisible(true);
    ui->imageLabel->setVisible(false);
    ui->imageLabel->setEnabled(false);
    setWindowTitle("New File");
}

void MainWindow::on_actionText_triggered() {
    ui->textEdit->setEnabled(true);
    ui->textEdit->setVisible(true);
    ui->imageLabel->setVisible(false);
    ui->imageLabel->setEnabled(false);
    ui->textEdit->clear();

    QFile file(QFileDialog::getOpenFileName(this, tr("Open File"), "/
home/dmytro/", tr("(*)")));

    if (!file.open(QIODevice::ReadOnly | QIODevice::Text))
    return;

    QTextStream in(&file);
    QString line = in.readLine();
    while (!line.isNull()) {
        ui->textEdit->setText(ui->textEdit->toPlainText() + line + "\n")
;

        line = in.readLine();
    }
    file.close();
```

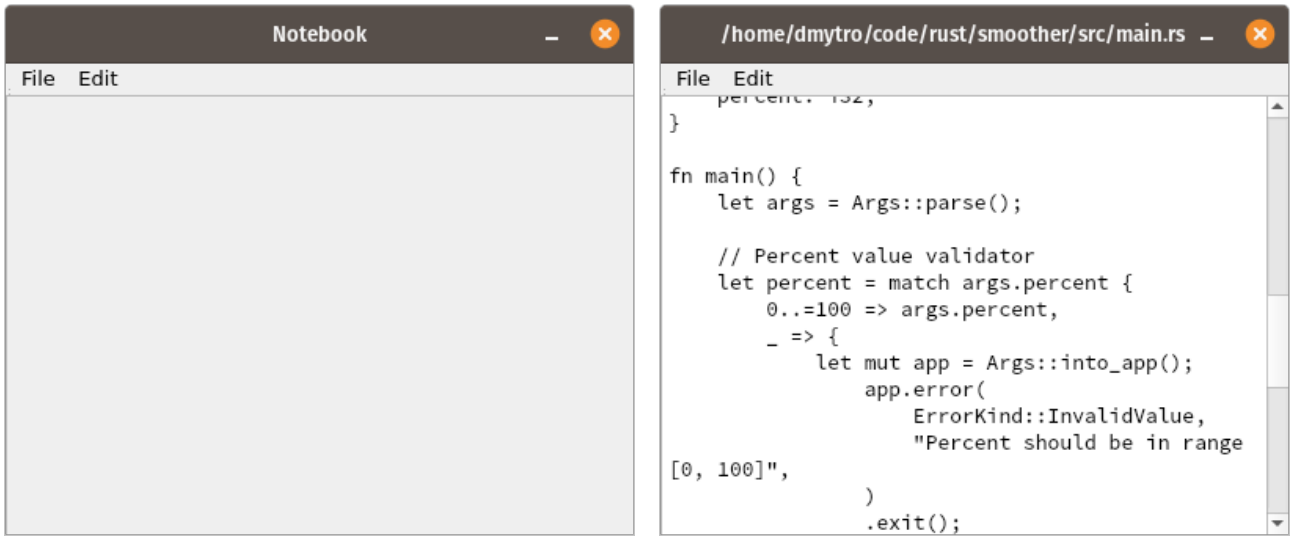
```
        setWindowTitle(file.fileName());
    }

    void MainWindow::on_actionImage_triggered() {
        ui->textEdit->clear();
        ui->textEdit->setEnabled(false);
        ui->textEdit->setVisible(false);
        ui->imageLabel->setEnabled(true);
        ui->imageLabel->setVisible(true);
        QString fileName = QFileDialog::getOpenFileName(this, tr("Open File"
), "/home/dmytro/", tr("(*)"));
        QImageReader reader(fileName);
        reader.setAutoTransform(true);
        const QImage newImage = reader.read();

        QImage image = newImage;
        ui->imageLabel->setPixmap(QPixmap::fromImage(image));
        ui->imageLabel->setScaledContents(true);

        setWindowTitle(fileName);
    }

    void MainWindow::on_actionSave_triggered() {
        if (!windowTitle().contains("*")) {
            return;
        }

        QString filename= QFileDialog::getSaveFileName(this, "Save As");

        if (filename.isEmpty())
        return;

        QFile file(filename);
        if (!file.open(QIODevice::WriteOnly | QIODevice::Text))
        return;

        QTextStream out(&file);

        out << ui->textEdit->toPlainText() << "\n";

        setWindowTitle(file.fileName());

        file.close();
    }

    void MainWindow::on_textEdit_textChanged() {
        if (!windowTitle().contains("*"))
        setWindowTitle(windowTitle() + "*");
    }

    void MainWindow::on_actionClose_triggered() {
        ui->textEdit->clear();
        ui->imageLabel->clear();
        setWindowTitle("Notebook");
        ui->textEdit->setEnabled(false);
        ui->textEdit->setVisible(false);
        ui->imageLabel->setEnabled(false);
        ui->imageLabel->setVisible(false);

    }
```

```
    void MainWindow::on_actionPring_triggered() {
        QPrintDialog dialog = QPrintDialog();
        dialog.open();
        dialog.exec();
    }

    void MainWindow::on_actionFind_2_triggered() {
        QInputDialog dialog = QInputDialog();
        bool ok;
        ui->textEdit->find(dialog.getText(this, tr("Find text"),
        tr("Text:"), QLineEdit::Normal,
        "", &ok),
        QTextDocument::FindBackward);
    }

    void MainWindow::on_actionFont_triggered() {
        ui->textEdit->setFont(QFontDialog::getFont(0, ui->textEdit->font()))
;

    }

    void MainWindow::on_actionColor_triggered() {
        ui->textEdit->setTextColor(QColorDialog().getColor());
    }

    void MainWindow::on_actionReplace_2_triggered() {
        QInputDialog to_replace_dialog = QInputDialog();
        QInputDialog replace_to_dialog = QInputDialog();
        QString before = to_replace_dialog.getText(this, tr("Replace"), tr("
Text:"), QLineEdit::Normal, "");
        QString after = replace_to_dialog.getText(this, tr("Replace"), tr("
Text:"), QLineEdit::Normal, "");
        QString text = ui->textEdit->toPlainText();
        text.replace(before, after, Qt::CaseSensitive);
        ui->textEdit->setText(text);
    }
```

Назва файлу: *mainwindow.h*

```
    #ifndef MAINWINDOW_H
    #define MAINWINDOW_H

    #include <QMainWindow>

    QT_BEGIN_NAMESPACE
    namespace Ui { class MainWindow; }
    QT_END_NAMESPACE

    class MainWindow : public QMainWindow
    {
        Q_OBJECT

        public:
        MainWindow(QWidget *parent = nullptr);
        ~MainWindow();

        private slots:
        void on_actionNew_triggered();
        void on_actionText_triggered();
        void on_actionImage_triggered();
```

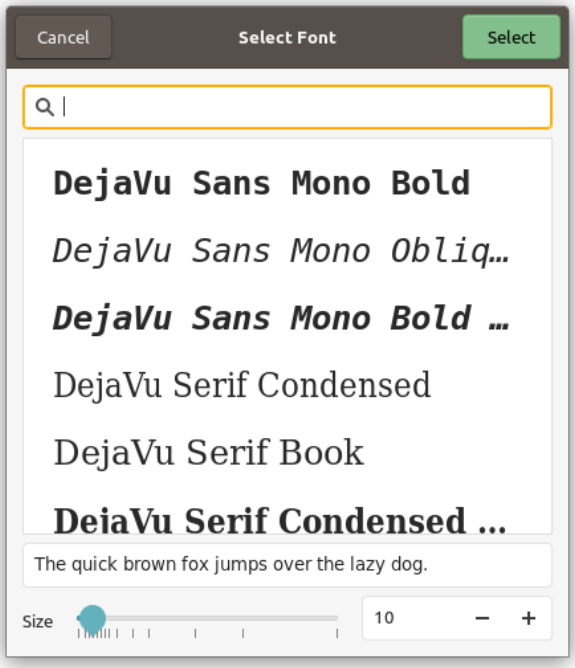
```
        void on_actionSave_triggered();
        void on_textEdit_textChanged();
        void on_actionClose_triggered();
        void on_actionPring_triggered();
        void on_actionFind_2_triggered();
        void on_actionFont_triggered();
        void on_actionColor_triggered();
        void on_actionReplace_2_triggered();

    private:
        Ui::MainWindow *ui;
};
#endif // MAINWINDOW_H
```



(а) (б)

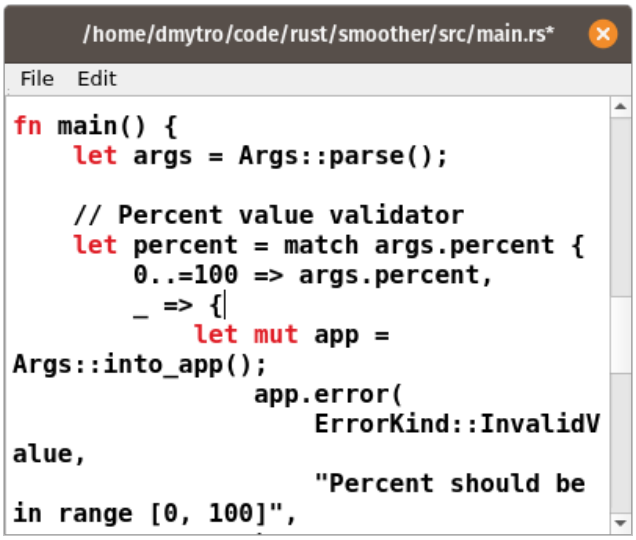
(в)

Рис. 5: Вигляд програми (а);
Робота програми з текстом(б);
Робота програми із зображеннями (в).



(а)

(б)



(в)

Рис. 6: Вибір шрифту (а);
Вибір кольору тексту(б);
Робота програми з різними шрифтами та кольорами (в).

Код програми "Гра"

Назва файлу: *main.cpp*

```
#include "mainwindow.h"

#include <QApplication>

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

```
}
```

Назва файлу: *mainwindow.cpp*

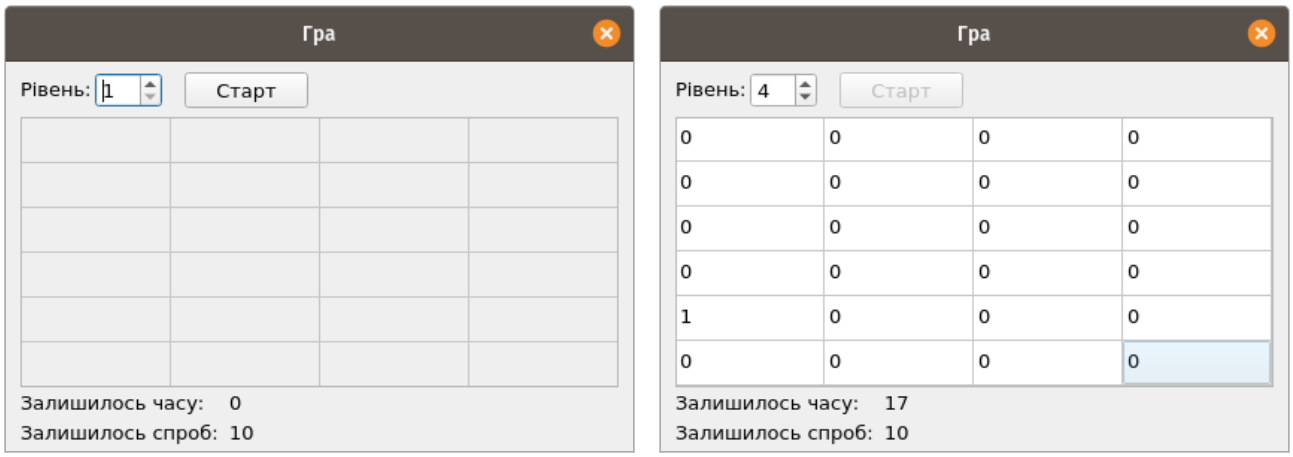
```
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include <QTimer>
#include <QMessageBox>

MainWindow::MainWindow(QWidget *parent)
: QMainWindow(parent)
, ui(new Ui::MainWindow) {
    this->gameTimer = new QTimer(this);
    this->labelTimer = new QTimer(this);
    this->loseTimer = new QTimer(this);
    connect(gameTimer, &QTimer::timeout, this, &MainWindow::spawnNumber)
;
    connect(labelTimer, &QTimer::timeout, this, &MainWindow::
updateTimeLabel);
    connect(loseTimer, &QTimer::timeout, this, &MainWindow::lose);

    ui->setupUi(this);
}

MainWindow::~MainWindow() {
    delete ui;
}

void MainWindow::on_startButton_clicked() {
    ui->tableWidget->setEnabled(true);
    ui->startButton->setEnabled(false);

    for (int i = 0; i < ui->tableWidget->rowCount(); i++) {
        for (int j = 0; j < ui->tableWidget->columnCount(); j++) {
            QTableWidgetItem *item = new QTableWidgetItem;
            item->setText("0");
            ui->tableWidget->setItem(i, j, item);
        }
    }

    int row = rand() % ( ui->tableWidget->rowCount() );
    int col = rand() % ( ui->tableWidget->columnCount() );
    ui->tableWidget->item(row, col)->setText("1");
    gameTimer->start((10 - ui->levelSpinBox->value()) * 500);
    labelTimer->start(100);
    loseTimer->start(20000);
}

void MainWindow::spawnNumber() {

    for (int i = 0; i < ui->tableWidget->rowCount(); i++)
    for (int j = 0; j < ui->tableWidget->columnCount(); j++)
    ui->tableWidget->item(i, j)->setText("0");

    int row = rand() % ( ui->tableWidget->rowCount() );
    int col = rand() % ( ui->tableWidget->columnCount() );
    ui->tableWidget->item(row, col)->setText("1");
    gameTimer->stop();
    gameTimer->start((10 - ui->levelSpinBox->value()) * 500);

}
```

```
    void MainWindow::updateTimeLabel() {
        ui->timeLabel->setText(QString::number(loseTimer->remainingTime() /
1000));
    }

    void MainWindow::on_tableWidget_cellClicked(int row, int column) {
        if (ui->tableWidget->item(row, column)->text() == "1") {
            ui->levelSpinBox->setValue(ui->levelSpinBox->value() + 1);
            ui->tableWidget->item(row, column)->setText("0");
            if (ui->levelSpinBox->value() == 9) win();
            else spawnNumber();
        } else {
            if (ui->attemptLabel->text() == "0" || ui->timeLabel->text() ==
"0") lose();
            else ui->attemptLabel->setText(
                QString::number(ui->attemptLabel->text().toInt() - 1)
            );
        }
    }

    void MainWindow::win() {
        labelTimer->stop();
        gameTimer->stop();
        loseTimer->stop();
        QMessageBox msgBox;
        msgBox.setText("You won.");
        msgBox.exec();
        ui->tableWidget->setEnabled(false);
        ui->startButton->setEnabled(true);
        ui->levelSpinBox->setValue(1);
        ui->timeLabel->setText("20");
        ui->attemptLabel->setText("10");
    }

    void MainWindow::lose() {
        labelTimer->stop();
        gameTimer->stop();
        loseTimer->stop();
        QMessageBox msgBox;
        msgBox.setText("You lost.");
        msgBox.exec();
        ui->tableWidget->setEnabled(false);
        ui->startButton->setEnabled(true);
        ui->levelSpinBox->setValue(1);
        ui->timeLabel->setText("20");
        ui->attemptLabel->setText("10");
    }
```

Назва файлу: *mainwindow.h*

```
    #ifndef MAINWINDOW_H
    #define MAINWINDOW_H

    #include <QMainWindow>
    #include <QTimer>

    QT_BEGIN_NAMESPACE
    namespace Ui { class MainWindow; }
    QT_END_NAMESPACE
```

```
class MainWindow : public QMainWindow
{
    Q_OBJECT

    public:
    MainWindow(QWidget *parent = nullptr);
    ~MainWindow();

    private slots:
    void spawnNumber();
    void updateTimeLabel();
    void win();
    void lose();
    void on_startButton_clicked();
    void on_tableWidget_cellClicked(int row, int column);

    private:
    Ui::MainWindow *ui;
    QTimer *gameTimer;
    QTimer *labelTimer;
    QTimer *loseTimer;

};
#endif // MAINWINDOW_H
```



(а) (б)

Рис. 7: Вигляд програми (а);
Робота програми (б).

Висновок

Я засвоїв принцип візуального програмування шляхом створення та налаштування проекту за допомогою середовища розробки QtCreator. Ознайомився з основними компонентами програм за допомогою розробки програм "Калькулятор", "Редактор текстових файлів та переглядач зображень", "Гра".