

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення
Дисципліна: Об'єктно-орієнтоване програмування

Лабораторна робота №2

**Тема: «ТИПИ ЗВ'ЯЗКІВ МІЖ КЛАСАМИ: АСОЦІАЦІЯ,
АГРЕГАЦІЯ, КОМПОЗИЦІЯ, РЕАЛІЗАЦІЯ ТА ЗАЛЕЖНІСТЬ»**

Виконав: ст. гр. КН-24

Куріщенко П. В.

Перевірив: асистент

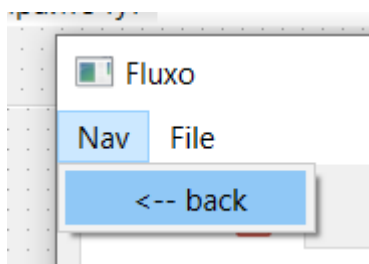
Козірова Н. Л.

Мета: ознайомитись з основними поняттями асоціація, агрегація, композиція, реалізація та залежність в ООП та навчитись їх програмно реалізовувати мовою C++.

Варіант – 14 - 10 = 4

Завдання 1: Додати кнопку для повернення до попередньої сторінки

Зміни fluxo.ui:



Зміни fluxo.h:

```
24 private slots:
25     void createTab();
26     void closeTab(int index);
27     void goBack(); // метод для повернення на попередню сторінку
```

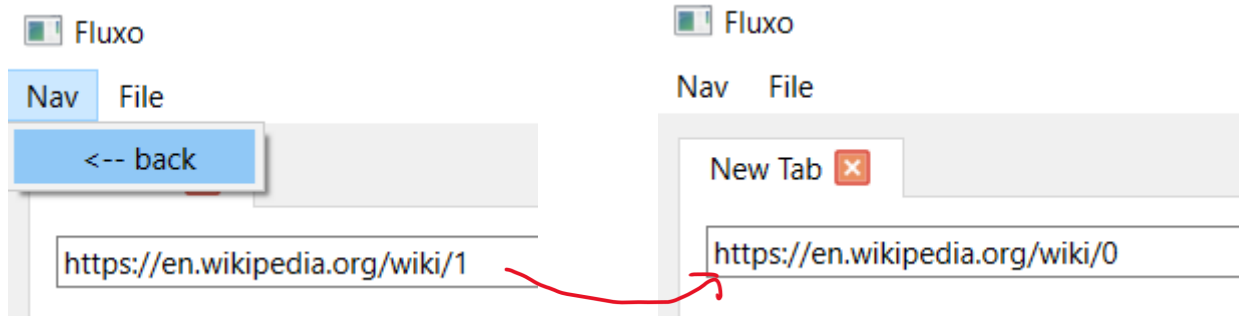
Зміни tab.h:

```
14 public:
15     explicit Tab(QObject *parent = nullptr);
16     ~Tab();
17
18     QLayout* getTabView() { return tabLayout; }
19     QWebEngineView* getView() { return view; } // геттер для звертання до QWebEngineView
```

Зміни fluxo.cpp:

```
26     connect (m_addTab, &QAction::triggered, this, &Fluxo::createTab);
27     connect (ui->tabWidget, &QTabWidget::tabCloseRequested, this, &Fluxo::closeTab);
28     connect (ui->action_back, &QAction::triggered, this, &Fluxo::goBack);
29 }
30
31
32
33
34
35
36 }
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58 void Fluxo::goBack()
59 {
60     QWidget *tab = ui->tabWidget->currentWidget();
61     m_tabs[tab]->getView()->back();
62 }
```

Результат виконання:



Завдання 2: Варіант 14-10 = 4. Створіть систему класів для моделювання інформації про музичний гурт.

Клас MusicalBand

Лістинг .h:

```
#ifndef MUSICALBAND_H
#define MUSICALBAND_H

#include "musician.h"
#include "bandleader.h"
#include <iostream>
#include <QList>

using namespace std;

class MusicalBand
{
public:
    MusicalBand();
    MusicalBand(const QString& n,
               const QString& s,
               const BandLeader& l)
        : name(n), style(s), leader(l) {}

    void addMusician(Musician*);
    void delMusician(Musician*);

    QString getName();
    QString getStyle();
    QList<Musician*> getList();
    int getLen();

private:
    QString name;
    QString style;
    QList<Musician*> musicians;
    BandLeader leader;
};
```

```
#endif // MUSICALBAND_H
```

Лістинг .cpp:

```
#include "musicalband.h"

MusicalBand::MusicalBand() {}

QString MusicalBand::getName() { return name; }

QString MusicalBand::getStyle() { return style; }

void MusicalBand::addMusician(Musician* m) { musicians.append(m); }

void MusicalBand::delMusician(Musician* m) {
    musicians.removeOne(m);
    delete m;
}

QList<Musician*> MusicalBand::getList() { return musicians; }

int MusicalBand::getLen() { return musicians.length(); }
```

Клас Musician

Лістинг .h:

```
#ifndef MUSICIAN_H
#define MUSICIAN_H

#include "instrument.h"
#include "manager.h"

class Musician
{
public:
    Musician();
    Musician(const QString& n,
             const Instrument& i,
             Manager* m)
        : name(n), instrument(i), manager(m) {}

    QString getName();

private:
    QString name;
    Instrument instrument;
    Manager* manager;
};

#endif // MUSICIAN_H
```

Лістинг .cpp:

```
#include "musician.h"

Musician::Musician() {}
```

```
QString Musician::getName() { return name; }
```

Клас Manager

Лістинг .h:

```
#ifndef MANAGER_H
#define MANAGER_H

#include <QString>

class Manager
{
public:
    Manager();
    Manager(const QString& n)
        : name(n) {}

    QString getName();

private:
    QString name;
};

#endif // MANAGER_H
```

Лістинг .cpp:

```
#include "manager.h"

Manager::Manager() {}

QString Manager::getName() { return name; }
```

Клас Instrument

Лістинг .h:

```
#ifndef INSTRUMENT_H
#define INSTRUMENT_H

#include <QString>

class Instrument
{
public:
    Instrument();
    Instrument(const QString& n, unsigned short y)
        : name(n), prodYear(y) {}

    QString getName();
    unsigned short getprodYear();

private:
    QString name;
    unsigned short prodYear;
};
```

```
};  
  
#endif // INSTRUMENT_H
```

Лістинг .cpp:

```
#include "instrument.h"  
  
Instrument::Instrument() {}  
  
QString Instrument::getName() { return name; }  
  
unsigned short Instrument::getprodYear() { return prodYear; }
```

Клас BandLeader

Лістинг .h:

```
#ifndef BANDLEADER_H  
#define BANDLEADER_H  
  
#include <QString>  
  
class BandLeader  
{  
public:  
    BandLeader();  
    BandLeader(const QString& n)  
        : name(n) {}  
  
    QString getName();  
  
private:  
    QString name;  
};  
  
#endif // BANDLEADER_H
```

Лістинг .cpp:

```
#include "bandleader.h"  
  
BandLeader::BandLeader() {}  
  
QString BandLeader::getName() { return name; }
```

Результат виконання:

```
Enter band name: KN24
Enter band style: rock
Enter band leader name: Pasha
Enter number of musicians: 2

--- Musician 1 ---
Enter musician name: Vova
Enter instrument name: guitar
Enter instrument production year: 2023
Enter manager name: Artem

--- Musician 2 ---
Enter musician name: Yarik
Enter instrument name: drums
Enter instrument production year: 2019
Enter manager name: Dima

--- Band Info ---
Band: KN24 (rock)
Leader: Pasha
Musician 1: Vova
Musician 2: Yarik
```