МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра ЕОМ



Звіт до лабораторних робіт

з ООП

Виконав:

ст. групи КІ-15

Ковальський П.А.

Прийняв:

викл. Козак Н.Б.

Львів-2020

ЛАБОРАТОРНА РОБОТА № 4.

КЛАСИ ТА ОБ’ЄКТИ

**Мета:** познайомитися із класами та об‘єктами.

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Визначення класу

Основна відмінність будь-якої об‘єктно-орієнтованої мови програмування від інших не об‘єктно-орієнтованих мов програмування є можливість створення нових визначених користувачем типів, що називаються класами. Клас – це визначений користувачем тип з даними-елементами (властивостями) та функціями-елементами (методами), що являються членами класу. Він зазвичай описує певну абстракцію якоїсь сутності реального світу з її властивостями і можливими діями. Змінна типу клас називається об‘єктом. Об‘єкт – це вже не абстракція, а елемент реального світу, який може володіти певними характеристиками, які задаються властивостями в класі, та діяти згідно операцій заданих методами у класі. Оголошення класу в мові С++ має наступний синтаксис (не обов‘язково щоб клас мав всі секції чи спадкував базовий клас): class ім'я\_класу : список\_базових\_класів , publіc: // Оголошення загальнодоступних (відкритих) членів класу, що можуть бути доступні звідусіль protected: // Оголошення захищених членів класу, що доступні тільки для похідних класів prіvate: // Оголошення закритих членів класу, що доступні лише членам цього класу, та не можуть бути змінені чи викликані з-за меж класу напряму, а лише за посередництвом методів з інших секцій };

Члени класу (властивості і методи), оголошені після ключового слова publіc стають відкритими членами. Це означає, що вони доступні: • усім іншим членам класу; • дружнім конструкціям (класам, методам, функціям); • членам похідних класів; • з-під об‘єктів класу після створення об‘єктів; після створення об‘єкту класу до його членів можна звертатися лише якщо вони є загальнодоступними. Члени класу, оголошені після ключового слова prіvate, стають закритими членами. Вони доступні: • для інших членів того ж класу; • друзям класу. Якщо в класі не оголошено секцій, але визначено методи чи властивості, то вони вважатимуться такими, що оголошені у секції prіvate. Для модифікації властивостей, що оголошені в секції private створюються спеціальні методи, які здійснюють цю модифікацію. Назви таких методів прийнято починати зі слова set. На жаргоні такі методи інколи називають сетерами. Для читання властивостей, що оголошені в секції private створюються спеціальні методи, які безпечно повертають значення цих властивостей. Назви таких методів прийнято починати зі слова get. На жаргоні такі методи інколи називають ґетерами. Члени класу, оголошені після ключового слова protected, стають захищеними членами. Вони доступні: • для інших членів того ж класу; • членам похідних класів; • друзям класу.

**Варіант-12**

Клас CBrowser (Інтернет-браузер). Клас зберігає адресу поточної відкритої сторінки, розмір сторінки, швидкість під‘єднання до інтернету (завантаження і відвантаження). Конструктор має ініціалізувати ці властивості. Методи дозволяють модифікувати і читати властивості, завантажувати і відвантажувати файли з і на сторінку повертаючи при цьому час обміну інформації, виводити на екран поточний стан об‘єкта.

**Код програми:**

#include <iostream>

#include <string>

using namespace std;

class CBrowser

{

public:

CBrowser();

CBrowser(std::string adress, double height, double width, double inSpeed, double outSpeed)

{

set\_PageAdress(adress);

set\_PageHeight(height);

set\_PageWidth(width);

set\_downloadSpeed(inSpeed);

set\_outloadSpeed(outSpeed);

}

~CBrowser();

void set\_PageAdress(std::string adress)

{

currentPageAdress = adress;

}

void set\_PageHeight(double height)

{

PageHeight = height;

}

void set\_PageWidth(double width)

{

PageWidth = width;

}

void set\_downloadSpeed(double inSpeed)

{

downloadSpeed = inSpeed;

}

void set\_outloadSpeed(double outSpeed)

{

outloadSpeed = outSpeed;

}

std::string get\_PageAdress()

{

return currentPageAdress;

}

double get\_PageHeight()

{

return PageHeight;

}

double get\_PageWidth()

{

return PageWidth;

}

double get\_downloadSpeed()

{

return downloadSpeed;

}

double get\_outloadSpeed()

{

return outloadSpeed;

}

double GetSizeOfSpeed(double sizeofFile)

{

double time = 0.0;

double speed;

speed = get\_downloadSpeed();

time = sizeofFile / speed;

return time;

}

double Download(double sizeofFile)

{

return GetSizeOfSpeed(sizeofFile);

}

double Outload(double sizeofFile)

{

double time = 0;

double speed;

speed = get\_outloadSpeed();

time = sizeofFile / speed;

return time;

}

void Show\_status()

{

cout << "Page adress: " << get\_PageAdress() << endl;

cout << "Page height: " << get\_PageHeight() << endl;

cout << "Page width: " << get\_PageWidth() << endl;

cout << "Page download speed: " << get\_downloadSpeed() << endl;

cout << "Page outload speed: " << get\_outloadSpeed() << endl;

}

private:

string currentPageAdress;

double PageHeight;

double PageWidth;

double downloadSpeed;

double outloadSpeed;

};

void Start()

{

CBrowser \*browser1;

CBrowser \*browser2;

double time[2];

string pagePath;

double sizeY;

double sizeX;

double speed\_in;

double speed\_out;

std::string pagePath2;

double sizeY2;

double sizeX2;

double speed\_in2;

double speed\_out2;

cout << "Page 1 adress: ";

getline(cin, pagePath);

cout << "Page 1 height: ";

cin >> sizeX;

cout << "Page 1 width: ";

cin >> sizeY;

cout << "Page 1 download speed: ";

cin >> speed\_in;

cout << "Page 1 outload speed: ";

cin >> speed\_out;

browser1 = new CBrowser(pagePath, sizeX, sizeY, speed\_in, speed\_out);

getchar();

cout << endl;

cout << "Page 2 adress: ";

getline(cin, pagePath2);

cout << "Page 2 height: ";

cin >> sizeX2;

cout << "Page 2 width: ";

cin >> sizeY2;

cout << "Page 2 download speed: ";

cin >> speed\_in2;

cout << "Page 2 outload speed: ";

cin >> speed\_out2;

browser2 = new CBrowser(pagePath2, sizeX2, sizeY2, speed\_in2, speed\_out2);

cout << endl;

time[0] = browser1->Download(500.8);

cout << "Page 1 time to download 500.8 Mb file : " << time[0] << endl;

time[1] = browser1->Download(345.6);

cout << "Page 2 time to download 345.6 Mb file : " << time[1] << endl;

cout << endl;

time[0] = browser1->Outload(500.8);

cout << "Page 1 time to outload 500.8 Mb file : " << time[0] << endl;

time[1] = browser1->Outload(345.6);

cout << "Page 2 time to outload 345.6 Mb file : " << time[1] << endl;

cout << endl;

cout << "Page 1 status : " << endl;

browser1->Show\_status();

cout << endl;

cout << "Page 2 status : " << endl;

browser2->Show\_status();

}

int main()

{

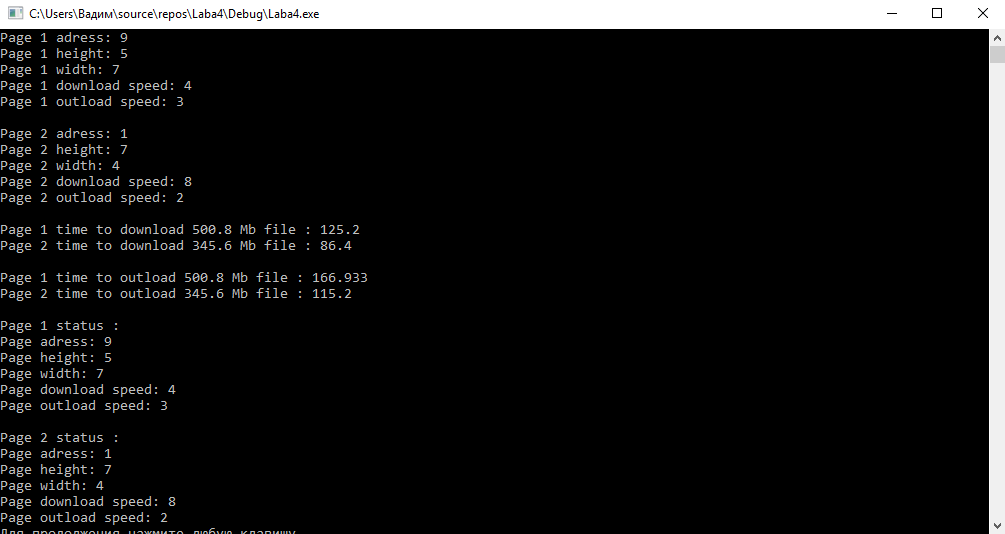
Start();

system("pause");

return 0;

}

**Результат:**



**Висновок:** Роблячи дану лабораторну роботу я познайомився із класами та об‘єктами.