 МІНІСТЕРСТВО ОСВІТИ I НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ   
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ

КАФЕДРА БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

**Практичне заняття №2**

з дисципліни «Об’єктно-орієнтоване програмування»

на тему: «Робота з об’єктами, використання покажчика this»

Варіант №11

**Виконав:**

студент гр. БС-52

Розюк П.В.

**Перевірив:**

ст. викладач каф. БМК

Радогуз А.В.

Зараховано від \_\_\_.\_\_\_.\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис викладача)

Київ-2017

**Завдання:**

1.Вивчити теоретичні теоретичні основи роботи з об’єктами та використання покажчика this.

2.Скористатися розробленим в попередній лабораторній класом для розширення його функціональності (нові методи класу мають працювати з покажчиком this).

3. Відповідно до свого варіанту завдання розробити діаграму класів в нотації UML та написати програму, яка містить клас для роботи із запитами користувача та інкапсулює в собі масив об’єктів розробленого в попередній лабораторній роботі класу.

**Створити масив об’єктів і вивести на екран:**

**-список файлів, дата створення яких належить заданому діапазону;**

**- список файлів, розмір яких перевищує заданий ;**

**- список файлів, ім’я яких починається з заданої букви;**

4.Скласти і захистити звіт.

**🞏 Практичне заняття без зауважень**

**🞏 Практичне заняття має зауваження:**

**🞏 несвоєчасний захист**

**🞏 присутні зауваження до UML діаграми:**

**🞏 діаграма класу не відповідає коду**

**🞏 виконані не за стандартом:**

**🞏 атрибути**

**🞏 відношення**

**🞏 багатостатність**

**🞏 інші зауваження:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**🞏 присутні зауваження до коду:**

**🞏 задача завдання вирішена хибно**

**🞏 код програми не компілюється**

**🞏 хибно задані специфікатори доступу**

**🞏 помилки у визначенні конструкторів / деструкторів**

**🞏 використано глобальні змінні**

**🞏 статичні змінні при роботі з масивами**

**🞏 оформлення коду**

**🞏 присутні зайві символи «{» та «}»**

**🞏 інші зауваження:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**🞏 невірні відповіді на запитання:**

**🞏 №1 🞏 №2 🞏 №3 🞏 №4 🞏 №5**

**🞏 №6 🞏 №7 🞏 №8 🞏 №9 🞏 №10**

**🞏 маються інші зауваження:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Лістинг програми:**

**File.h**

#pragma once

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <string.h>

using namespace std;

class File

{

private:

char \*name, \*date;

int number, size;

public:

void printdata();

// створення конструктора

File(char \*curname, char \*curdate, int cursize, int curnumber);

//set

void set\_name(char \*curname);

void set\_date(char \*curdate);

void set\_size(int cursize);

void set\_number(int curnumber);

//get

string get\_name();

string get\_date();

int get\_size();

int get\_number();

~File();

};

**File.cpp**

#include "File.h"

#include <string.h>

File::File(char \*curname, char \*curdate, int cursize, int curnumber)

{

int n = strlen(curname);

int z = strlen(curdate);

name = new char[n];

date = new char[z];

this->name = strcpy(name, curname);

this->date = strcpy(date, curdate);

this->size = cursize;

this->number = curnumber;

}

void File::set\_name(char \*curname)

{

name = new char[256];

this->name = strcpy(name, curname);

};

void File::set\_date(char \*curdate)

{

date = new char[256];

this->date = strcpy(date, curdate);

};

void File::set\_size(int cursize)

{

this->size = cursize;

};

void File::set\_number(int curnumber)

{

this->number = curnumber;

};

string File::get\_date() { return date; };

string File::get\_name() { return name; };

int File::get\_size() { return size; };

int File::get\_number() { return number; }

File :: ~File()

{

delete[] name;

delete[] date;

}

**arr.h**

#pragma once

#include "File.h"

class arr

{

public:

arr();

~arr();

File \*a[4];

void Size();

void Dateout( string g);

void Name();

void print();

};

**arr.cpp**

#include "arr.h"

#include "File.h"

#include <iostream>

#include <string>

#include <stdio.h>

#include <conio.h>

using namespace std;

arr::arr()

{

a[0] = new File("File\_1", "25.02.15", 135, 5);

a[1] = new File("File\_2", "17.03.15", 125, 54);

a[2] = new File("doc\_3", "28.05.16", 150, 55);

a[3] = new File("doc\_4", "27.02.17", 115, 4);

}

arr::()

{

delete a[0];

delete a[1];

delete a[2];

delete a[3];

}

void arr::Size()

{

int k, b;

cout << "Input size of the file:";

cin >> b;

for (int i = 0; i < 4; i++)

{

k = a[i]->get\_size();

if (b<k)

{

cout << "Data of the file " << i + 1 << ":\n";

printf("%s\n", a[i]->get\_name().c\_str());

printf("%s\n", a[i]->get\_date().c\_str());

cout << a[i]->get\_size() << " Mb" << endl;

cout << a[i]->get\_number() << "\n\n";

}

}

}

void arr::Name()

{

char \*b;

string c;

b = new char[256];

cout << "Input first character of the file:";

cin >> b;

for (int i = 0; i < 4; i++)

{

c = a[i]->get\_name();

if (c[0] == \*b)

{

printf("%s\n\n", a[i]->get\_name().c\_str());

}

}

delete[]b;

}

void arr::print()

{

for (int i = 0; i < 4; i++)

{

cout << "Data of the file " << i + 1 << ":\n";

printf("%s\n", a[i]->get\_name().c\_str());

printf("%s\n", a[i]->get\_date().c\_str());

cout << a[i]->get\_size() << " Mb" << endl;

cout << a[i]->get\_number() << "\n\n";

}

}

void arr::Dateout(string b)

{

for (int i = 0; i < 4; i++)

{

if (b == a[i]->get\_date())

{

cout << "Data of the file " << i + 1 << ":\n";

printf("%s\n", a[i]->get\_name().c\_str());

printf("%s\n", a[i]->get\_date().c\_str());

cout << a[i]->get\_size() << " Mb" << endl;

cout << a[i]->get\_number() << "\n\n";

}

}

}

**main.cpp**

#include "File.h"

#include "arr.h"

#include <iostream>

#include <conio.h>

#include <string>

using namespace std;

void main()

{

string c;

arr \*a1 = new arr();

a1->print();

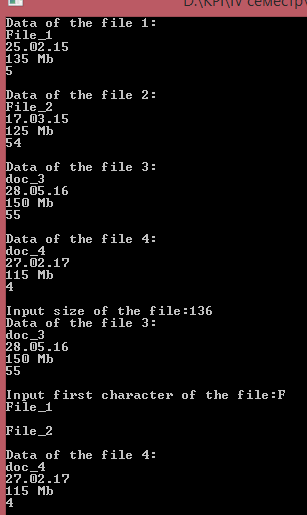
a1->Size();

a1->Name();

a1->Dateout("27.02.17");

\_getch();

}



**UML-діаграма:**



**Контрольні питання**

**1. Що таке покажчик this?**

Коли викликається будь-який метод, значення this стає адресою об’єкту, для якого метод

викликається. This інтерпретується як будь-який інший вказівник на об’єкт, його можна

використовувати для отримання доступу до даного об’єкту.

**2. Навести приклад, в якому не можна організувати код без використання покажчика this**

*class A*

*{*

*//поля*

*A\* method()*

*};*

*……*

*A\* method()*

*{*

*return this;*

*}*

Не потрібно створювати новий покажчик

**3. Два способи ініціалізації об’єктів у програмі**

Явне та неявне використання конструктора. При неявному використанні програма сама викликає конструктор, який виділяє пам’ять під об’єкт. В явному використанні конструктора в коді програми ми маємо змогу при створенні нового об’єкту одразу ініціалізувати його поля.

**4. Як ініціалізувати масив об’єктів?**

Замість елементів явно прописати конструктори нових об’єктів:

*Pc store [5] = { Pc( "Intel", "Asus", 32, 2, "Palit", false ),*

*Pc( "Intel", "Biostar", 1000, 4, "His", true ),*

*Pc( "Atom", "Asus", 500, 4, "Palit", true ),*

*Pc( "Atom", "Asus", 200, 4, "Gainward", false )*

*Pc( "Intel", "Biostar", 600, 2, "His", false ) };*

Інший варіант: об’явити об’єкти без конструктора (порожній конструктор для неявного виклику задати через перегрузку функцій). Потім запустити цикл, на кожній ітерації якого викликати всі сетери об’ктів даного класа в потрібному нам екземплярі.

Приклад: конструктор **station**(), метод **create**();

**5. Чому не можна ініціалізувати поле, що є масивом, під час визначення класу?**

При визначені класу ми лише прописуємо його структуру. Поняття ініціалізації стосується лише конкретних об’єктів, а не їх опису.

**6. Перелічити випадки, коли один клас викіористовує інший**

Об’єкти одного класу викликають методи іншого; об’єкт класу передається як параметр іншого класу; повертається як результат методу; створюється в методі як локальна змінна.

**7. Які є види відношень між класами?**

*Об’єднання* – об’єкти одного класу викликають методи іншого; об’єкт класу передається як

параметр іншого класу; повертається як результат методу; створюється в методі як локальна змінна;

*Узагальнення –* UML-наслідування, так як базовий клас є більш загальна форма нащадка

*Асоціація –* в загальному випадку коли об’єкт одного класу є атрибутом другого.

*Реалізація –* деякий клас надає сервіс, а інший реалізує конкретну поведінку цього сервісу

(інтерфейс та його реалізація).

**8. Які види відношень між класами визначені в мові моделювання UML**

Відношення залежності, асоціації, узагальнення, реалізації.

**9. Який тип відношень встановлено між класами в даній лабораторній роботі?**

Асоціація виду агрегації.

**10.Що означає оператор (::)?**

Оператор отримання доступу до класу.