 МІНІСТЕРСТВО ОСВІТИ I НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ   
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ

КАФЕДРА БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

**Практичне заняття №1**

з дисципліни «Об’єктно-орієнтоване програмування»

на тему: «Реалізація класів, конструктори та деструктори»

Варіант №11

**Виконав:**

студент гр. БС-52

Розюк П.В.

**Перевірив:**

ст. викладач каф. БМК

Алхімова С.М

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

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

(підпис викладача)

Київ-2017

**Завдання:**

Вивчити теоретичні відомості щодо створення класів та об’єктів мовою програмування C++.

1. Відповідно до свого варіанту розробити клас та відповідну діаграму класів в нотації UML для описаних в завданні характерних даних, визначити конструктор користувача, за допомогою якого можна ініціалізувати всі поля визначеного класу, а також методи для зміни значень кожного із полів:

**Створити клас що характеризує файл: ім’я файлу, розмір, дата створення, кількість звернень.**

1. Розробити програмний додаток, в якому необхідно створити об’єкт, що є екземпляром визначеного в завданні класу, та ініціалізувати його поля за допомогою створеного конструктора користувача, вивести інформацію про створений об’єкт на екран, змінити значення полів об’єкта за допомогою викликів відповідних методів класу, після чого знову вивести інформацію про модифікований об’єкт на екран, видалити створений об’єкт.
2. 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:**

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

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

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

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

Діаграма класів UML:



Лістинг програми:

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <conio.h>

#include <stdio.h>

#include <string.h>

using namespace std;

class File

{

private:

char \*name, \*date, \*size;

int number;

public:

void printdata()

{

cout << "Description of the file:\n";

cout << name << endl;

cout << date << endl;

cout << size << endl;

cout << number << endl;

}

// створення конструктора

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

name(curname), date(curdate), size(curdate), number(curnumber)

{

int n = strlen(curname);

int z = strlen(curdate);

int g = strlen(cursize);

name = new char[n];

date = new char[z];

size = new char[g];

strcpy(name, curname);

strcpy(date, curdate);

strcpy(size, cursize);

number = curnumber;

}

//set

void set\_name(char \*curname)

{

strcpy(name, curname);

};

void set\_date(char \*curdate)

{

strcpy(date, curdate);

};

void set\_size(char \*cursize)

{

strcpy(size, cursize);

};

void set\_number(int curnumber)

{

number = curnumber;

};

//get

char \*get\_name() { return name; };

char \*get\_date() { return date; };

char \*get\_size() { return size; };

int get\_number() { return number; }

~File()

{

delete name; delete date; delete size;

}

};

void main()

{

int a, b, x, g;

char \*q, \*w, \*e, \*k, \*d, \*z;

q = new char[20];

w = new char[20];

e = new char[20];

setlocale(LC\_CTYPE, "rus");

File objc1("File\_1", "25.02.15", "135 Mb", 5);

objc1.printdata();

cout << "If you want to change data of the file press 1, no - 0:\n";

cin >> a;

if (a == 1)

{

cout << "Input what do you want to change in file: name-1, date-2, size-3, number of calls-4:";

cin >> g;

switch (g)

{

case 1:

cout << "\nInput new name:\n";

cin >> q;

objc1.set\_name(q); break;

case 2:

cout << "\nInput new date:\n";

cin >> w;

objc1.set\_date(w); break;

case 3:

cout << "\nInput new size:\n";

cin >> e;

objc1.set\_size(e); break;

case 4:

cout << "\nInput new number of calls:\n";

cin >> b;

objc1.set\_number(b); break;

default: cout << "Sorry,but you have entered the number which is not in this list";

}

}

cout << "New data:\n";

k = objc1.get\_name();

z = objc1.get\_date();

d = objc1.get\_size();

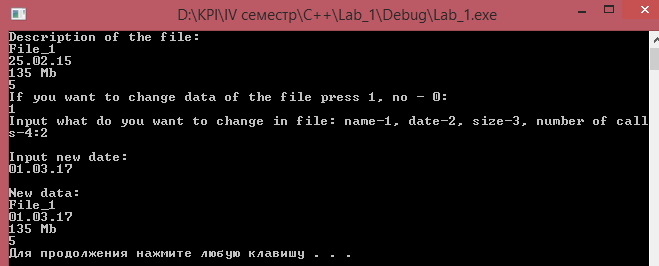
x = objc1.get\_number();

cout << k << endl; cout << z << endl; cout << d << endl; cout << x << endl;

system("pause");

\_getch();

}



Контрольні запитання:

1. Що таке клас, навіщо він потрібен?

Клас- це сутність що задає типи даних користувача, в яких дані характеризують одне ціле і відображають поведінку об’єкту.

1. Що таке об’єкт?

Це зміна певного класу .

1. Як пов’язані між собою класи і об’кти в програмі?

Клас – це шаблон для об’єкта, що описує його структуру(визначають поля) і поведінку(визначають методи ,опис яких знаходиться в класі).

1. Що таке інкапсуляція?

Інкапсуляція – у класах об’єднуються дані і методи для роботи з ними, лише через методи можливий доступ до прихованих даних класу.

1. За рахунок чого реалізується захист від несанкціонованого доступу до даних?

За допомогою специфікаторів: public, private, protected.

1. Чим відрізняються поля від змінних?

Тим, що вони визначають структуру об’єкта,це дані які характеризують об’єкт.

1. Що визначають методи класу?

Методи(функції) визначають поведінку об’єкта і описуються всередині класу.

1. Перелічити чим можуть бути поля класу?
2. Що таке конструктор, навіщо він потрібен?

Це функція класу, яка викликається автоматично при створенні статичного чи динамічного об’єкту класу. Він потрібен для ініціалізації об’єкта і змінних класу.

1. Що таке деструктор, навіщо він потрібен?

Це функція, яка автоматично викликається програмою, вона не повертає значень і не має параметрів, слугує для звільнення пам’яті що виділив конструктор.