Звіт

По лабораторній роботі №2

Виконав:

Студент ФІТ 2-11

Забродський Олексій Русланович

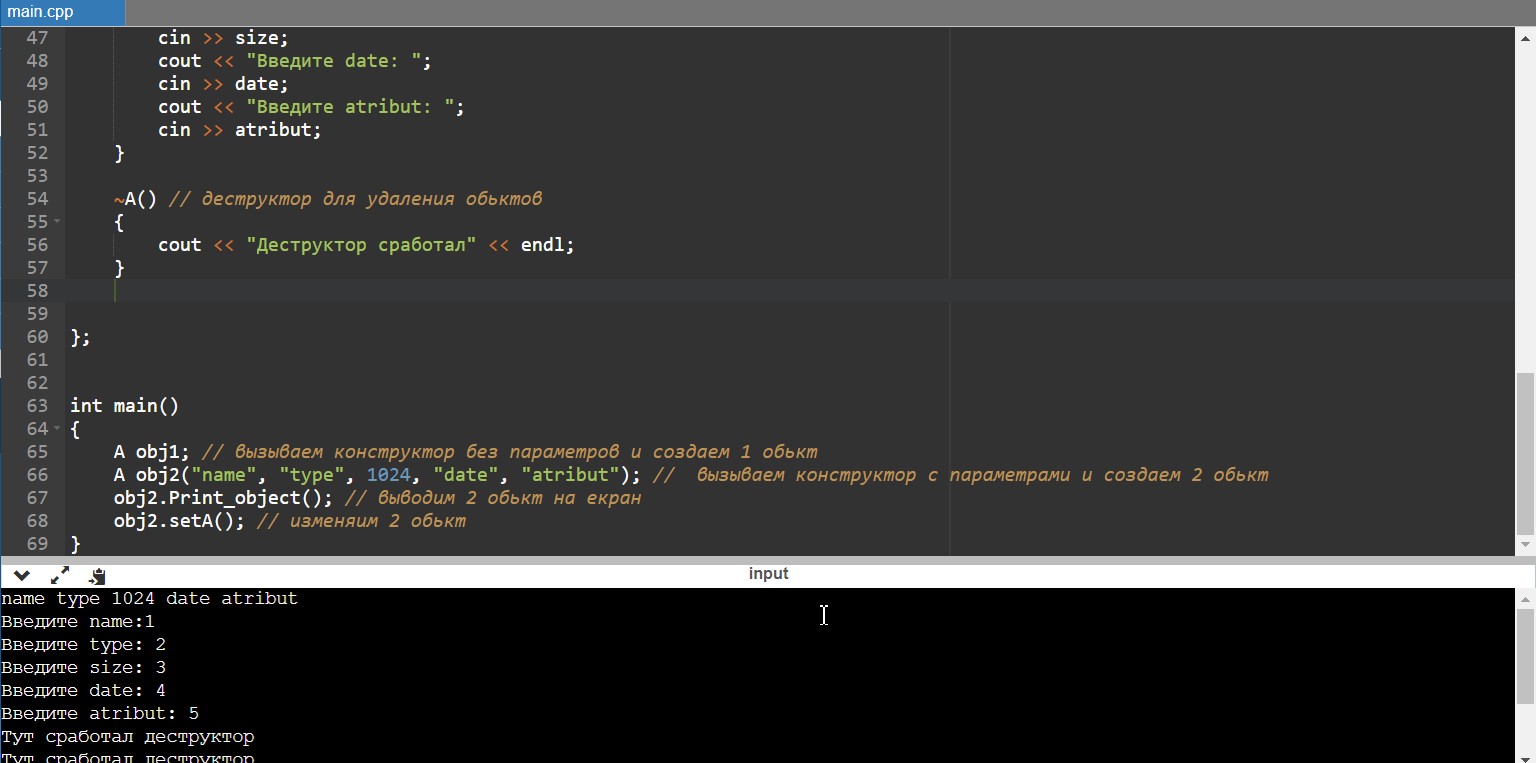
**Завдання 1**

Створити клас для обробки записів бази даних у відповід­ності з вказаною предметною областю. Розробити програму, дотримуючись таких вимог:

1. Реалізувати окремими функціями класу:
   * конструктори без параметрів та з параметрами;
   * деструктор;
   * виведення інформації на екран;
   * редагування записів.
2. При розробці програми слід здійснити захищення даних (опис з модифікатором private) для ізоляції елементів-даних класу від підпрограм, в яких цей клас використовується.
3. В головній функції передбачити створення масиву об’єктів класу та виконання пошуку згідно вказаного критерію.

**Варіант 8:**

|  |  |  |  |
| --- | --- | --- | --- |
| 8. | Список файлів | ім'я файла, розширення, розмір, дата створення, атрибути. | Дата створення |



**Лістинг програми:**

#include <iostream>

#include <string>

using namespace std;

class A

{

public:

string name;

string type;

int size;

string date;

string atribut;

public:

A() // конструктор без параметров

{

name = " ";

type = " ";

size = 0;

date = " ";

atribut = " ";

}

A(string nn, string tt, int ss, string dd, string aa) // конструктор с параметрами

{

name = nn;

type = tt;

size = ss;

date = dd;

atribut = aa;

}

void Print\_object() // вывод на экран

{

cout << name <<" "<< type <<" "<< size << " " << date<<" "<< atribut << endl;

}

void setA() // с помощью этого метода изменим начальные значения

{

cout << "Введите name:";

cin >> name;

cout << "Введите type: ";

cin >> type;

cout << "Введите size: ";

cin >> size;

cout << "Введите date: ";

cin >> date;

cout << "Введите atribut: ";

cin >> atribut;

}

~A() // деструктор для удаления обьктов

{

cout << "Деструктор сработал" << endl;

}

};

int main()

{

A obj1; // вызываем конструктор без параметров и создаем 1 обькт

A obj2("name", "type", 1024, "date", "atribut"); // вызываем конструктор с параметрами и создаем 2 обькт

obj2.Print\_object(); // выводим 2 обькт на екран

obj2.setA(); // изменяим 2 обькт

obj2.Print\_object();

}

Вхідні дані програми: A obj2("name", "type", 1024, "date", "atribut").

Вихідні дані: 1 2 3 4 5.

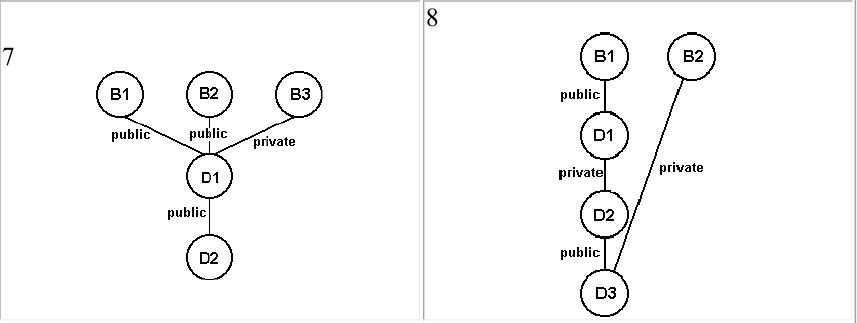
**Завдання 2**

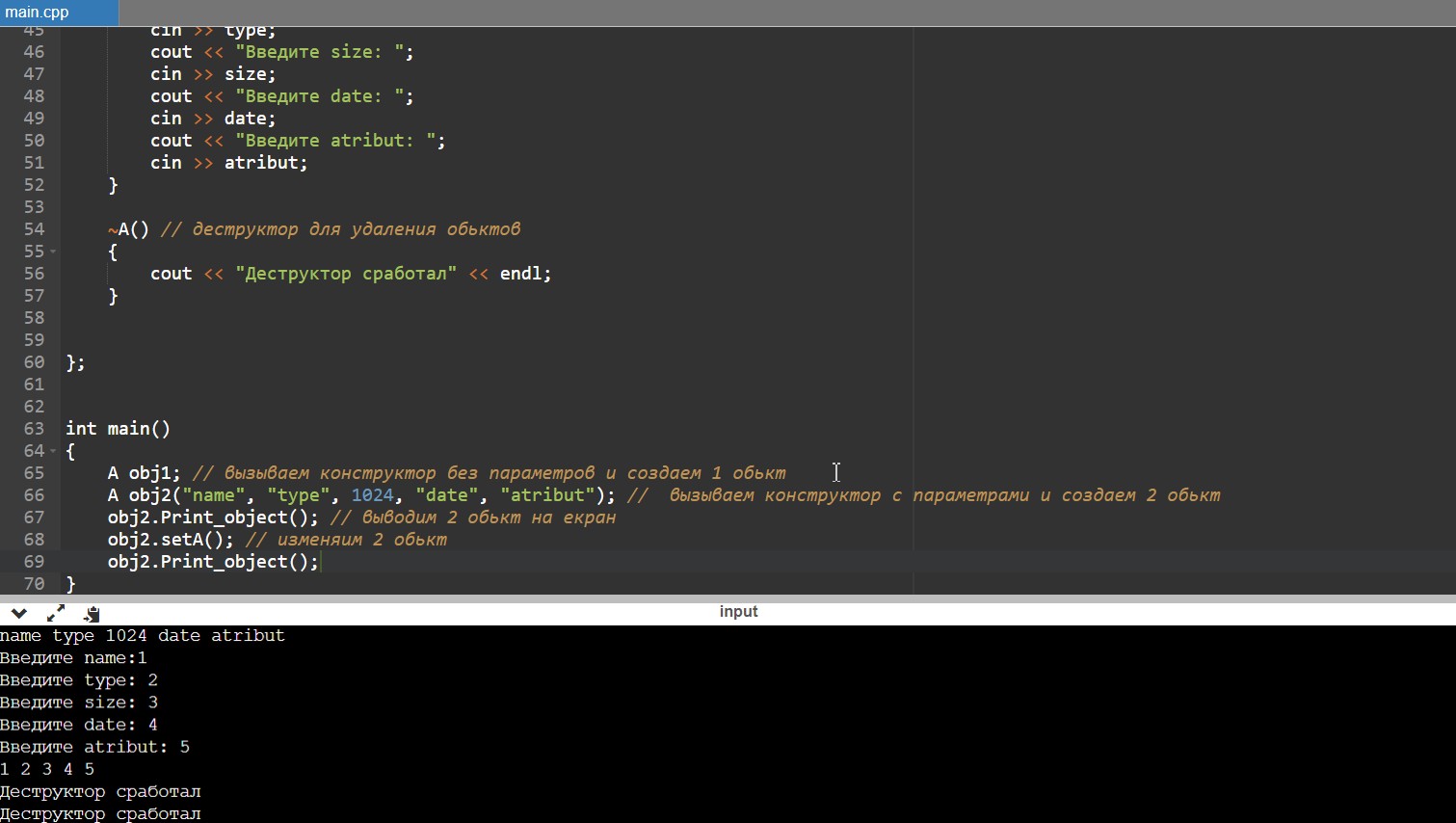
Необхідно побудувати ієрархію класів відповідно схемі успадкування, наведеній у варіанті завдання.

Кожен клас повинен містити конструктор ініціалізації, і функцію show для виведення значень.

Функція main повинна ілюструвати ієрархію успадкування.

**Варіант 8:**





**Лістинг програми:**

#include <iostream>

#include <stdlib.h>

using namespace std;

class B1{

int a;

public:

B1() { };

B1(int x) { a=x; } ;

void show\_B1() { cout <<"B1= "<< a << "\n"; }

};

class B2{

int e;

public:

B2() {};

B2(int k) { e = k; };

void show\_B2() { cout <<"B2= "<< e << "\n"; }

};

class D1: public B1 {

int b;

public:

D1(int x, int y) : B1(y) { b=x;};

void show\_D1() { cout <<"D1= "<< b << "\n"; show\_B1();}

};

class D2: private D1{

int c;

public:

D2(int x, int y, int q) : D1(y,q) { c=x;};

void show\_D2() { cout <<"D2= "<< c << "\n"; show\_D1();}

};

class D3: private B2, public D2 {

int d;

public:

D3(int x, int y, int z, int i, int j) : D2(y,z,i), B2(j) { d=x;}

void show\_D3() { cout << "D3= "<<d << "\n"; show\_B2(); show\_D2();}

};

main()

{

setlocale(LC\_ALL, "Russian");

D3 temp(100,200,300,400,500);

cout << "D3 temp(100,200,300,400,500);\n";

cout<< "\n Відповідно до ієрархії класу D3: \n";

temp.show\_D3();

return 0;

}

Вхідні дані програми: D3 temp(100,200,300,400,500).

Вихідні дані відповідно до ієрархії класу D3:

D3= 100

B2= 500

D2= 200

D1= 300

B1= 400

**Висновок**

Виконуючи лабораторну роботу я набув навичок розробки програм з використанням базового принципу об’єктно-орієнтованого програмування – успадкування. Навчився використовувати конструктори та деструктори при наслідуванні.