

## Лабораторна робота 4

**Тема.** Створення настільного додатку на основі технології Microsoft .NET з використанням конструкторів та деструкторів на мові C++

**Мета.** Ознайомитись із можливостями створення конструкторів та деструкторів у Visual Studio та їх використання у Windows Form додатках.

### Приклад виконання завдання

**Завдання.** Створити настільний додаток на основі технології Microsoft .NET з використанням мови CLR (Common Language Runtime) для розв'язку наступного завдання: *нехай дано базовий клас школа та похідний клас учень. Створити конструктори та деструктори базового та похідного класів. В похідному класі реалізувати методи пошуку за прізвищем учня та його середнім балом. Вхідні дані завантажуються із файлу та вихідні дані зберігаються у файл..*

1 Створити пустий проект CLR на основі технології Windows Forms.

2 Додати новий елемент до проекту – Windows Form.

3 Додати наступний програмний код у файл форми з розширенням .cpp (має таке ж ім'я, як і файл заголовків форми), який обирається в Оглядачі рішень (Solution Explorer).

```
1      #include "MyForm.h"
2
3      using namespace lab1; //ім'я вашого проекту
4      //[STAThreadAttribute]
5      int main(array<System::String ^> ^args)
6      {
7          Application::EnableVisualStyles();
8          Application::SetCompatibleTextRenderingDefault(false);
9          Application::Run(gcnew MyForm());
10         return 0;
11     }
```

4 Задати налаштування для проекту у його властивостях.

6 Створити базовий та похідні класи для розв'язку завдання. Реалізувати конструктори та деструктори в базовому по похідному класах. Оголосити масив об'єктів похідних класів у файлі заголовків (Header.h):

– базовий клас школа;

```

class School//базовий клас Школа
{
    char* NameSchool;//назва школи
    char* AdressSchool;//адреса школи
    char* TelepfoneSchool;//телефон школи
    char* EmailSchool;//email-адреса школи
public:
    School() {};//конструктор за замовчуванням
    School(char* na, char* a, char* t, char* e);//конструктор базового класу
    ~School() {};//деструктор класу
    //методи повернення значень закритих змінних класу
    char* GetNameSchool() { return NameSchool; }
    char* GetAdressSchool() { return AdressSchool; }
    char* GetTelepfoneSchool() { return TelepfoneSchool; }
    char* GetEmailSchool() { return EmailSchool; }
};

```

— похідний клас учень;

```

class Pupil:public School//похідний клас Учень
{
    char* NamePupil;//прізвище, ім'я, по батькові учня
    char* Class;//назва класу
    char* ClassTeacher;//класний керівник
    double GPA;//середній бал
public:
    Pupil() {};//конструктор за замовчуванням
    Pupil(char* np, char* cl, char* ct, double g, char* na, char* a, char* t, char* e);//конструктор похідного класу
    ~Pupil() {};//деструктор класу
    void SearchName(char* sn);//метод пошуку за учнем
    void SearchClass(char* c);//метод пошуку за класом
    void SearcGPA(double g);//метод пошуку за середнім балом
    void OutputFile();//збереження даних у файл
}mas[n];

```

7 Підключити директиви у файлі заголовків визначення класів (Header.h).

```

#include "MyForm.h"
#include<stdio.h>
#include<string.h>
#define n 3

```

8 Реалізувати у базовому та похідному класах методи та конструктори:

— конструктор базового класу;

```

School::School(char* na, char* a, char* t, char* e)
{
    NameSchool = na;
    AdressSchool = a;
    TelepfoneSchool = t;
    EmailSchool = e;
}

```

— конструктор похідного класу;

```

Pupil::Pupil(char* np, char* cl, char* ct, double g, char* na, char* a, char* t, char* e) :School(na, a, t, e)
{
    NamePupil = np;
    Class = cl;
    ClassTeacher = ct;
    GPA = g;
}

```

- метод пошуку за учнем;

```
void Pupil::SearchName(char* sn)
{
    if (strcmp(NamePupil, sn) == 0)
    {
        //збереження у файл
        FILE* f;
        f = fopen("search_name_pupil.txt", "a");
        char* name = GetNameSchool();
        fprintf(f, "%s\t%s\t%s\t%s\t%.2f\n", name, NamePupil, Class, ClassTeacher, GPA);
        fclose(f);
    }
}
```

- метод пошуку за класом;

```
void Pupil::SearchClass(char* c)
{
    if (strcmp(Class, c) == 0)
    {
        //збереження у файл
        FILE* f;
        f = fopen("search_class_pupil.txt", "a");
        char* name = GetNameSchool();
        fprintf(f, "%s\t%s\t%s\t%s\t%.2f\n", name, NamePupil, Class, ClassTeacher, GPA);
        fclose(f);
    }
}
```

- метод пошуку за середнім балом;

```
void Pupil::SearcGPA(double g)
{
    if (GPA == g)
    {
        //збереження у файл
        FILE* f;
        f = fopen("search_gpa_pupil.txt", "a");
        char* name = GetNameSchool();
        fprintf(f, "%s\t%s\t%s\t%s\t%.2f\n", name, NamePupil, Class, ClassTeacher, GPA);
        fclose(f);
    }
}
```

- метод збереження даних про учнів у файл.

```
void Pupil::OutputFile()
{
    FILE * f = fopen("pupils.txt", "a");
    char* name = GetNameSchool();
    char* adress = GetAdressSchool();
    char* telephone = GetTelepfoneSchool();
    char* email = GetEmailSchool();
    fprintf(f, "%s\t%s\t%s\t%s\t%s\t%.2f\n", name, adress, telephone, email, NamePupil, Class, ClassTeacher, GPA);
    fclose(f);
}
```

9 Спроекувати головну форму програми (див. Рис. 1, Рис. 2). Задати для властивості форми ControlBox значення False.

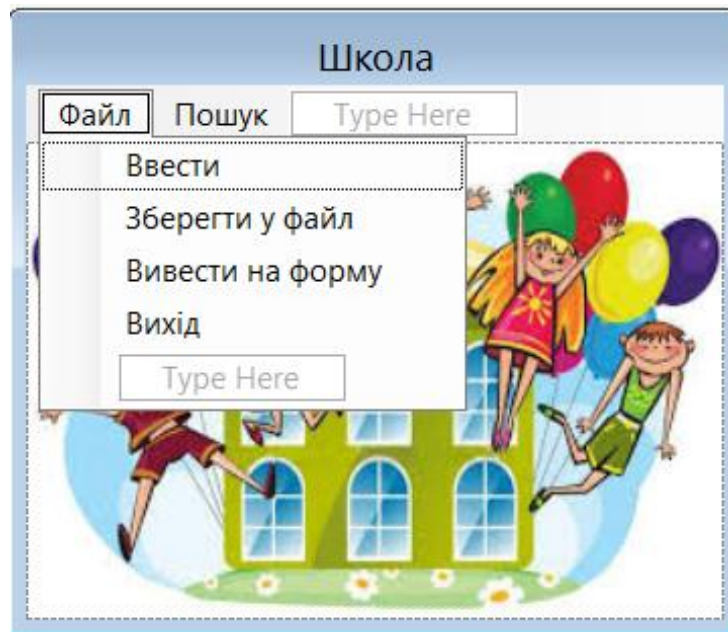


Рисунок 1 – Меню «Файл» головної форми

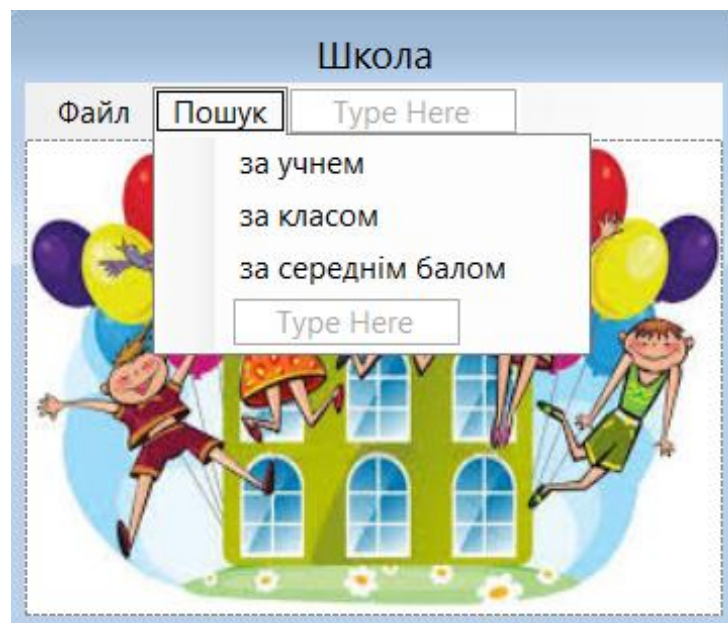


Рисунок 2 – Меню «Пошук» головної форми

10 На головній формі розмістити компоненти:

- menuStrip1 – для створення меню;
- pictureBox1 – для розміщення картинки.

11 Задати властивість Icon для форми. Для цього вибрати файл \*.ico із зображенням школи.

12 Спроекувати форму для введення даних про учня та школу. Під час створення форми задати назву файлу Input.h (див. Рис. 3). Для форми задати наступні властивості:

- ControlBox – False;

- ShowIcon – False.

The screenshot shows a Windows-style dialog box with a blue title bar and a light gray background. The title bar contains the text 'Введення даних про учня'. Inside the dialog, there are eight text input fields arranged vertically, each preceded by a label. The labels are: 'Назва школи', 'Адреса школи', 'Телефон школи', 'Email-адреса школи', 'Прізвище, ім'я, по батькові учня', 'Назва класу', 'Класний керівник', and 'Середній бал'. The input fields are white with thin gray borders. At the bottom right of the dialog, there is a button labeled 'Ввести'.

Рисунок 3 – Форма для введення даних у похідний клас

13 Спроекувати форму виведення даних про учня та школу. Під час створення форми задати назву файлу Output.h (див. Рис. 4). Для форми задати наступні властивості:

- ControlBox – False;
- ShowIcon – False.

Для компоненти dataGridView1 задати поля та властивість RowHeadersVisible – False.

Назва школи	Адреса	Телефон	Email	ПІП учня	Назва класу	Класний керівник	Середній бал

Закрити

Рисунок 4 – Форма для виведення даних про учнів та школу

#### 14 Запрограмувати команду меню «Файл\Ввести»

```
private: System::Void ввестиToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^ e) {
    Input ^ form1 = gcnew Input();
    //виведення форми Input
    this->Hide();
    form1->ShowDialog();
    this->Show();
}
```

#### 15 Запрограмувати команду меню «Файл\Зберегти у файл»

```
private: System::Void зберегтиУФайлToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^ e) {
    for(int i = 0; i < qountity; i++)
        //запис у файл
        mas[i].OutputFile();
}
```

#### 16 Запрограмувати команду меню «Файл\Вивести на форму»

```
private: System::Void вивестиНаФормуToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^ e) {
    //виведення форми Output
    Output^ form2 = gcnew Output();
    this->Hide();
    form2->ShowDialog();
    this->Show();
}
```

#### 17 Запрограмувати команду меню «Файл\Вихід»

```
private: System::Void вихідToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^ e) {
    this->Close();
}
```

18 Запрограмувати кнопку «Ввести» для введення даних про учнів у похідний клас (форма Input.h).

```
private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) {
    //зберегти у класі
    char* str1 = (char*)(void*)Marshal::StringToHGlobalAnsi(textBox1->Text);
    char* str2 = (char*)(void*)Marshal::StringToHGlobalAnsi(textBox2->Text);
    char* str3 = (char*)(void*)Marshal::StringToHGlobalAnsi(textBox3->Text);
    char* str4 = (char*)(void*)Marshal::StringToHGlobalAnsi(textBox4->Text);
    char* str5 = (char*)(void*)Marshal::StringToHGlobalAnsi(textBox5->Text);
    char* str6 = (char*)(void*)Marshal::StringToHGlobalAnsi(textBox6->Text);
    char* str7 = (char*)(void*)Marshal::StringToHGlobalAnsi(textBox7->Text);
    double str8 = Convert::ToDouble(textBox8->Text);

    mas[qountity] = { str1, str2, str3, str8, str4, str5, str6, str7};
    qountity++;
    this->Close();
}
```

19 Запрограмувати кнопку «Закрити» на формі Output.h.

```
private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) {
    this->Close();
}
```

20 Задати у конструкторі форми Output виведення даних в компоненту dataGridView1 із файлу pupils.txt.

```
public:
    Output(void)
    {
        InitializeComponent();
        //
        //TODO: Add the constructor code here
        //
        dataGridView1->Rows->Clear();
        array<String^>^ lines = System::IO::File::ReadAllLines("pupils.txt", System::Text::Encoding::GetEncoding(1251));
        for each(String^ str in lines)
        {
            DataGridViewRow^ row = gcnew DataGridViewRow();
            row->CreateCells(dataGridView1);
            for (int i = 0; i < str->Split('\t')->Length; i++)
                row->Cells[i]->Value = str->Split('\t')[i];
            dataGridView1->Rows->Add(row);
        }
    }
```

21 Спроекувати форму виведення даних про учня та школу. Під час створення форми задати назву файлу Search.h (див. Рис. 5). Для форми задати наступні властивості:

- ControlBox – False;
- ShowIcon – False.

Для компоненти dataGridView1 задати поля та властивість RowHeadersVisible – False.



Рисунок 5 – Форма пошуку

22 Змінити доступ до компонент label1, textBox1 та dataGridView1 форми Search з private на public.

```
public: System::Windows::Forms::Label^ label1;
public: System::Windows::Forms::TextBox^ textBox1;
public: System::Windows::Forms::DataGridView^ dataGridView1;
```

23 Запрограмувати команду меню «Знайти\за учнем» головної форми (файл MyForm.h).

```
private: System::Void заУчнемToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^ e) {
    Search^ form3 = gcnew Search();
    this->Hide();

    form3->label1->Text = "Введіть прізвище, ім'я та по батькові учня";
    form3->k = 1;
    form3->ShowDialog();
    this->Show();
}
```

24 Запрограмувати команду меню «Знайти\за класом» головної форми (файл MyForm.h).

```
private: System::Void заКласомToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^ e) {
    Search^ form3 = gcnew Search();
    this->Hide();
    form3->label1->Text = "Введіть назву класу для пошуку";
    form3->k = 2;
    form3->ShowDialog();
    this->Show();
}
```

25 Запрограмувати команду меню «Знайти\за середнім балом» головної форми (файл MyForm.h).



```
private: System::Void заСереднімБаломToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^ e) {
    Search^ form3 = gcnew Search();
    this->Hide();
    form3->label1->Text = "Введіть середній бал для пошуку";
    form3->k = 3;
    form3->ShowDialog();
    this->Show();
}
```

26 Оголосити відкриті змінні в класі форми Search.

```
public:
    int k = 0; String^ s;
```

27 Запрограмувати кнопку «Знайти» форми «Пошук».

```
private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) {
    if ( k == 1)
    {
        s = "search_name_pupil.txt";
        char* name = (char*)(void*)Marshal::StringToHGlobalAnsi(textBox1->Text);
        for (int i = 0; i < qountity; i++)
            mas[i].SearchName(name);
    }
    if ( k == 2)
    {
        s = "search_class_pupil.txt";
        char* clas = (char*)(void*)Marshal::StringToHGlobalAnsi(textBox1->Text);
        for (int i = 0; i < qountity; i++)
            mas[i].SearchClass(clas);
    }
    if (k == 3)
    {
        s = "search_gpa_pupil.txt";
        double b = Convert::ToDouble(textBox1->Text);
        for (int i = 0; i < qountity; i++)
            mas[i].SearcGPA(b);
    }

    dataGridView1->Rows->Clear();
    array<String^>^ lines = System::IO::File::ReadAllLines(s, System::Text::Encoding::GetEncoding(1251));

    for each(String^ str in lines)
    {
        DataGridViewRow^ row = gcnew DataGridViewRow();
        row->CreateCells(dataGridView1);
        for (int i = 0; i < str->Split('\t')->Length; i++)
            row->Cells[i]->Value = str->Split('\t')[i];
        dataGridView1->Rows->Add(row);
    }
}
```

28 Реалізувати очищення результуючих файлів після закінчення виконання програми.

### Індивідуальні завдання

Створити настільний додаток на основі технології Microsoft .NET з використанням мови CLR (Common Language Runtime) для розв'язку індивідуального завдання. У програмі передбачити збереження даних у файлі і можливість читання з раніше

збереженого файлу. Реалізувати в класах конструктори та деструктори, методи введення, виведення та пошуку за різними критеріями.

1. Створити клас для зберігання даних про список файлів на диску. Утворити похідний клас, залучивши до нього як мінімум два додаткових поля таким чином, щоб клас набув більшої спеціалізованості. Для другого класу використати конструктор, аби він містив усі аргументи, необхідні для ініціалізації об'єкту похідного класу. Створити необхідні функції, що дозволяють виводити інформацію на екран та можливість додавати, знищувати та шукати дані.

2. Створити клас: пряма призма, нижня основа якої паралельна координатній площині  $XOY$ . Для цього успадкувати клас трикутник на площині. Доповнити члени-дані успадкованого класу висотою і відстанню від нижньої основи до площини  $XOY$ . Створити конструктори, деструктори базового та похідних класів. Визначити функцію знаходження площі поверхні призми.

3. Нехай дано базовий клас коло та похідний клас сектор. Доповнити члени-дані успадкованого класу. Створити конструктори, деструктори базового та похідних класів. Реалізувати в похідному класі метод знаходження площі сектора.

4. Нехай дано базовий клас працівник та похідний клас викладач. Створити конструктори та деструктори базового та похідного класів. В похідному класі реалізувати метод пошуку за прізвищем викладача.

5. Створити клас: тетраедр, нижня основа якого паралельна координатній площині  $XOY$ . Для цього успадкувати клас трикутник на площині. Доповнити члени-дані успадкованого класу висотою. Створити конструктори, деструктори базового та похідних класів. Визначити функцію знаходження площі поверхні тетраедра.

6. Нехай дано базовий клас коло та похідний клас сегмент. Доповнити члени-дані успадкованого класу. Створити конструктори, деструктори базового та похідних класів. Реалізувати в похідному класі метод знаходження площі сегмента.

7. Дано базовий клас банк та похідний клас вкладник. Реалізувати конструктори для ініціалізації закритих членів базового та похідного класів; методи пошуку за кодом та прізвищем вкладника.

8. Створити клас: куб, нижня основа якого паралельна координатній площині  $XOY$ . Для цього успадкувати клас квадрат на площині. Доповнити члени-дані успадкованого

класу висотою. Створити конструктори, деструктори базового та похідних класів. Визначити функцію знаходження об'єму конуса.

9. Нехай дано базовий клас чотирикутник та похідний клас ромб. Доповнити члени-дані успадкованого класу. Створити конструктори, деструктори базового та похідних класів. Реалізувати в похідному класі метод знаходження площі ромба.

10. Нехай дано базовий клас працівник та похідний клас робітник. Створити конструктори та деструктори базового та похідного класів. В похідному класі реалізувати методи пошуку за прізвищем робітника та обчислення його заробітної плати.

11. Створити клас: циліндр, нижня основа якого паралельна координатній площині  $ХОУ$ . Для цього успадкувати клас коло на площині. Доповнити члени-дані успадкованого класу висотою. Створити конструктори, деструктори базового та похідних класів. Визначити функцію знаходження об'єму циліндра.

12. Нехай дано базовий клас чотирикутник та похідний клас паралелограм. Доповнити члени-дані успадкованого класу. Створити конструктори, деструктори базового та похідних класів. Реалізувати в похідному класі метод знаходження площі паралелограма.

13. Нехай дано базовий клас підприємство та похідний клас матеріальні цінності. Створити конструктори та деструктори базового та похідного класів. В похідному класі реалізувати методи пошуку за інвентарним номером та назвою матеріальних цінностей.

14. Створити клас: піраміда, нижня основа якого паралельна координатній площині  $ХОУ$ . Для цього успадкувати клас чотирикутник на площині. Доповнити члени-дані успадкованого класу висотою. Створити конструктори, деструктори базового та похідних класів. Визначити функцію знаходження площі бічної поверхні піраміди.

15. Нехай дано базовий клас трикутник та похідний клас рівнобедрений трикутник. Доповнити члени-дані успадкованого класу. Створити конструктори, деструктори базового та похідних класів. Реалізувати в похідному класі метод знаходження площі рівнобедреного трикутника.

16. Нехай дано базовий клас музей та похідний клас експонат. Створити конструктори та деструктори базового та похідного класів. В похідному класі реалізувати метод пошуку за назвою експонату.

17. Створити клас: прямокутний паралелепіпед, нижня основа якого паралельна координатній площині  $ХОУ$ . Для цього успадкувати клас прямокутник на площині.

Доповнити члени-дані успадкованого класу висотою. Створити конструктори, деструктори базового та похідних класів. Визначити функцію знаходження об'єму паралелепіпеда.

18. Створити клас для зберігання даних про книги у бібліотеці. Утворити похідний клас, залучивши до нього як мінімум два додаткових поля таким чином, щоб клас набув більшої спеціалізованості. Для другого класу використати конструктор, аби він містив усі аргументи, необхідні для ініціалізації об'єкту похідного класу. Створити необхідні функції, що дозволяють виводити інформацію на екран та можливість додавати, знищувати та шукати дані.

19. Нехай дано базовий клас трикутник та похідний клас рівносторонній трикутник. Доповнити члени-дані успадкованого класу. Створити конструктори, деструктори базового та похідних класів. Реалізувати в похідному класі метод знаходження площі рівностороннього трикутника.

20. Створити клас для зберігання даних про розклад занять навчальної групи. Утворити похідний клас, залучивши до нього як мінімум два додаткових поля таким чином, щоб клас набув більшої спеціалізованості. Для другого класу використати конструктор, аби він містив усі аргументи, необхідні для ініціалізації об'єкту похідного класу. Створити необхідні функції, що дозволяють виводити інформацію на екран та можливість додавати, знищувати та шукати дані.

21. Нехай дано базовий клас склад та похідний клас товар. Створити конструктори та деструктори базового та похідного класів. В похідному класі реалізувати метод пошуку за назвою товару та метод обчислення відпускної ціни товару.

22. Створити клас: конус, нижня основа якого паралельна координатній площині  $HOY$ . Для цього успадкувати клас коло на площині. Доповнити члени-дані успадкованого класу висотою. Створити конструктори, деструктори базового та похідних класів. Визначити функцію знаходження об'єму конуса.

23. Створити клас: призма, нижня основа якого паралельна координатній площині  $HOY$ . Для цього успадкувати клас паралелограм на площині. Доповнити члени-дані успадкованого класу висотою. Створити конструктори, деструктори базового та похідних класів. Визначити функцію знаходження об'єму призми.

24. Нехай дано базовий клас чотирикутник та похідний клас прямокутник. Доповнити члени-дані успадкованого класу. Створити конструктори, деструктори

базового та похідних класів. Реалізувати в похідному класі метод знаходження площі прямокутника.

25. Нехай дано базовий клас архів та похідний клас матеріал. Створити конструктори та деструктори базового та похідного класів. В похідному класі реалізувати метод пошуку за архівним номером.

26. Створити клас для зберігання даних про користувачів локальної мережі. Утворити похідний клас, залучивши до нього як мінімум два додаткових поля таким чином, щоб клас набув більшої спеціалізованості. Для другого класу використати конструктор, аби він містив усі аргументи, необхідні для ініціалізації об'єкту похідного класу. Створити необхідні функції, що дозволяють виводити інформацію на екран та можливість додавати, знищувати та шукати дані.

27. Нехай дано базовий клас трикутник та похідний клас прямокутний трикутник. Доповнити члени-дані успадкованого класу. Створити конструктори, деструктори базового та похідних класів. Реалізувати в похідному класі метод знаходження гіпотенузи прямокутного трикутника.

28. Створити клас: конус, нижня основа якого паралельна координатній площині XOY. Для цього успадкувати клас коло на площині. Доповнити члени-дані успадкованого класу висотою. Створити конструктори, деструктори базового та похідних класів. Визначити функцію знаходження площі поверхні конуса.

29. Створити клас для зберігання даних про архів програмного забезпечення. Утворити похідний клас, залучивши до нього як мінімум два додаткових поля таким чином, щоб клас набув більшої спеціалізованості. Для другого класу використати конструктор, аби він містив усі аргументи, необхідні для ініціалізації об'єкту похідного класу. Створити необхідні функції, що дозволяють виводити інформацію на екран та можливість додавати, знищувати та шукати дані.

30. Створити клас: шар. Для цього успадкувати клас коло на площині. Доповнити члени-дані успадкованого класу висотою. Створити конструктори, деструктори базового та похідних класів. Визначити функцію знаходження площі поверхні конуса.

### **Контрольні запитання**

1 Дати визначення поняття конструктора та описати його призначення.

2 Дати визначення поняття деструктора та описати його призначення.

- 3 Описати процес ініціалізації даних-членів класу за допомогою конструкторів.
- 4 Охарактеризувати конструктори, що задані за замовчуванням.
- 5 Охарактеризувати конструктори копіювання.
- 6 Описати призначення конструктори та деструктори похідних класів.
- 7 Пояснити порядок виклику конструкторів і деструкторів базових та похідних класів.
- 8 Пояснити призначення конструкторів з апараметрами.