Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

ЗВІТ

ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 12

з навчальної дисципліни

“Базові методології та технології програмування”

ПРОГРАМНА РЕАЛІЗАЦІЯ АБСТРАКТНИХ ТИПІВ ДАНИХ

ЗАВДАННЯ ВИДАВ

доцент кафедри кібербезпеки та програмного забезпечення

Доренський О. П.

[https://github.com/odorenskyi/](https://github.com/odorenskyi/Dmytro-Parkhomenko-KB18)

ВИКОНАВ

студент академічної групи КН-23

Гребенюк Д. О.

ПЕРЕВІРИВ

ст. викладач кафедри кібербезпеки   
та програмного забезпечення

Дрєєва Г. М.

Кропивницький – 2024

**Тема:** Програмна реалізація абстрактних типів даних

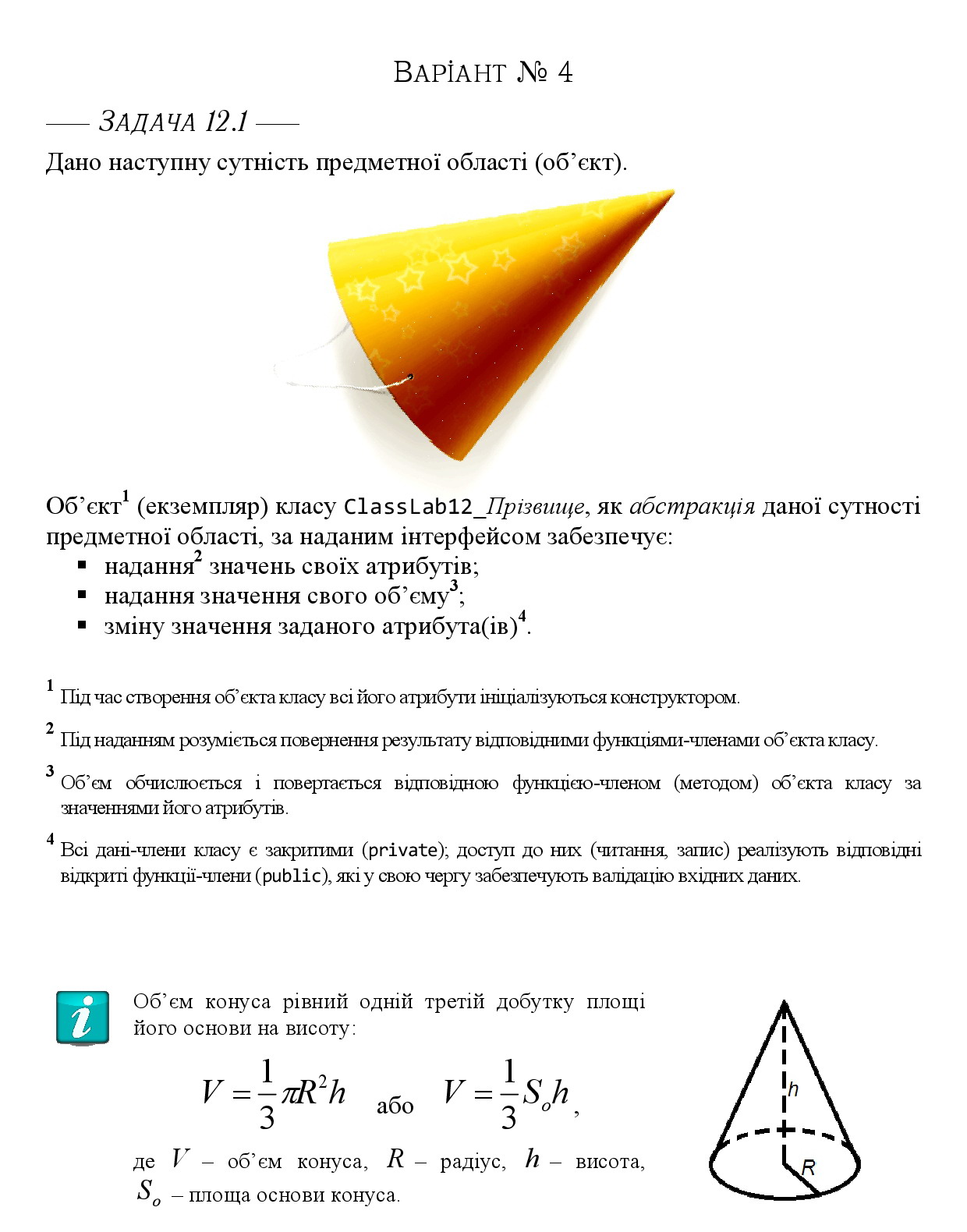
**Мета роботи:** Набуття ґрунтовних вмінь і практичних навичок об’єктного аналізу й проектування, створення класів С++ та тестування їх екземплярів, використання препроцесорних директив, макросів і макрооператорів під час реалізації програмних засобів у кросплатформовому середовищі Code::Blocks.

**Завдання до лабораторної роботи:**

1. Як складову заголовкового файлу ModulesПрізвище.h розробити клас ClassLab12\_Прізвище –– формальне представлення абстракції сутності предметної області (об’єкта) за варіантом, ― поведінка об’єкта якого реалізовує розв’язування задачі 7.1.

2. Реалізувати додаток Teacher, який видає 100 звукових сигналів і в текстовий файл TestResults.txt записує рядок “Встановлені вимоги порядку виконання лабораторної роботи порушено!”, якщо файл проекта main.срр під час його компіляції знаходився не в \Lab12\prj, інакше –– створює об’єкт класу ClassLab12\_Прізвище із заголовкового файлу ModulesПрізвище.h та виконує його unit-тестування за тест-сьютом(ами) із \Lab12\TestSuite, протоколюючи результати тестування в текстовий файл \Lab12\TestSuite\TestResults.txt.

**Варіант 4**



**Аналіз та постановка задачі 12.1**

*Концептуалізація предметної області*

Об’єктом, згідно з варіантом завдання, є святковий ковпак.

*Об’єктний аналіз*

Атрибути:

- Колір: колір ковпака англійською мовою

- Малюнок: малюнок на ковпаку (“none”, “stars”, “squares”, “triangles” тощо)

- Висота: висота ковпака

- Радіус основи: радіус основи ковпака (конуса)

*Визначення інтерфейсів сутності*

Отримання та задання значення: кольору, малюнку, висоти та радіуса основи ковпака. Прототипи функції будуть визначені при проектуванні модуля та класа.

*Аналіз вимог до програмного модуля ModulesHrebeniuk*

Назва класу - ClassLab12\_Hrebeniuk. Атрибути ініціалізуються конструктором. Об'єм конуса (ковпака) обчислюється і повертається відповідною функцією-членом за значенням атрибутів. Доступ до даних-членів повинен бути закритим (private), для атрибутів повинні бути відповідні відкриті (public) функції-члени (читання та запис значення), що забезпечують валідацію вхідних даних.

Для атрибутів будуть використовуватись такі типи даних:

- Колір (color) (string)

- Малюнок (pattern) (string)

- Висота (height) (float)

- Радіус основи (radius) (float)

Функції-члени класу:

- string getColor(), void setColor(string color)

- string getPattern(), void setPattern(string pattern)

- float getHeight(), void setHeight(float height)

- float getRadius(), void setRadius(float radius)

**Аналіз та постановка задачі 12.2**

Завдання полягає в реалізації додатка Teacher, який виконує наступні функції:

1. Перевіряє місцезнаходження файлу проекта main.cpp під час компіляції:

- Якщо файл знаходиться не в каталозі \Lab12\prj, записує в текстовий файл TestResults.txt повідомлення: "Встановлені вимоги порядку виконання лабораторної роботи порушено!", виводе 100 звукових сигналів та завершує виконання застосунку.

- Якщо файл знаходиться у вказаному каталозі, створює об'єкт класу ClassLab12\_Прізвище з заголовкового файлу ModulesПрізвище.h.

2. Виконує unit-тестування цього об'єкта, використовуючи тест-сьюти з каталогу \Lab12\TestSuite\TestSuite.txt.

3. Протоколює результати тестування у текстовий файл \Lab12\TestSuite\TestResults.txt.

**Документація формату тест-кейса та протоколу його читання**

*Формат тест-кейса у файлі TestSuite.txt*

1. Кожен рядок у файлі представляє окремий тест-кейс.

2. Поля тест-кейса розділені символом "|".

3. Порядок полів у тест-кейсі:

1) Колір ковпака: Значення кольору або пусте значення.

2) Малюнок ковпака: Значення малюнка або пусте значення.

3) Висота ковпака: Числове значення висоти або пусте значення.

4) Радіус основи ковпака: Числове значення радіусу або пусте значення.

*Приклад формату тест-кейса*

|||

red|striped|0.1|0.1

||5.0|3.0

||0.05|0.05

blue||2.0|1.0

|dotted|0.05|0.1

green|checkered|10.0|5.0

yellow|plain|-5.0|-3.0

purple|solid|100.0|50.0

black|gradient|1.0|0.0

*Протокол читання тест-кейсів*

1. Відкриття файлу TestSuite.txt для читання

Для цього використовується стандартна бібліотека fstream для роботи з файлами. Відкриття файлу для читання за допомогою ifstream.

2. Читання рядків з файлу

Кожен рядок представляє окремий тест-кейс. Читання здійснюється по одному рядку за допомогою getline.

3. Розділення рядка на поля

Кожен рядок розділяється на частини за допомогою символу "|". Для цього використовується об'єкт stringstream.

4. Створення об'єкта класу ClassLab12\_Hrebeniuk

5. Запис результатів тестування у файл TestResults.txt:

Результати кожного тест-кейса записуються у файл TestResults.txt. Формат запису результатів включає всі властивості ковпака (колір, малюнок, висоту, радіус та об’єм).

**Лістинг заголовкового файла ModulesHrebeniuk.h**

#ifndef MODULESHREBENIUK\_H\_INCLUDED

#define MODULESHREBENIUK\_H\_INCLUDED

#include <cmath>

using namespace std;

class ClassLab12\_Hrebeniuk {

public:

ClassLab12\_Hrebeniuk(string color = "", string pattern = "", float height = 0.1, float radius = 0.1) {

setColor(color);

setPattern(pattern);

setHeight(height);

setRadius(radius);

}

string getColor() { return mColor; }

string getPattern() { return mPattern; }

float getHeight() { return mHeight; }

float getRadius() { return mRadius; }

float getVolume() { return (1.0/3 \* M\_PI \* pow(mRadius, 2) \* mHeight); }

void setColor(string color);

void setPattern(string pattern);

void setHeight(float height);

void setRadius(float radius);

private:

string mColor;

string mPattern;

float mHeight;

float mRadius;

};

void ClassLab12\_Hrebeniuk::setColor(string color) {

if (color.empty()) {

color = "white";

}

mColor = color;

}

void ClassLab12\_Hrebeniuk::setPattern(string pattern) {

if (pattern.empty()) {

pattern = "none";

}

mPattern = pattern;

}

void ClassLab12\_Hrebeniuk::setHeight(float height) {

if (height < 0.1) {

height = 0.1;

}

mHeight = height;

}

void ClassLab12\_Hrebeniuk::setRadius(float radius) {

if (radius < 0.1) {

radius = 0.1;

}

mRadius = radius;

}

#endif // MODULESHREBENIUK\_H\_INCLUDED

**Лістинг проекта Teacher**

#include <iostream>

#include <Windows.h>

#include <fstream>

#include <sstream>

#include "ModulesHrebeniuk.h"

#define TEST\_SUITE\_FILE "..\\..\\TestSuite\\TestSuite.txt"

#define TEST\_RESULTS\_FILE "..\\..\\TestSuite\\TestResults.txt"

using namespace std;

int main()

{

system("chcp 1251 & cls");

string currentFilePath = \_\_FILE\_\_;

int checkResult = currentFilePath.find("\\lab12\\prj\\");

if (checkResult == -1) {

for (int i = 0; i < 100; ++i) {

Beep(500, 100);

}

ofstream resultFile(TEST\_RESULTS\_FILE);

if (resultFile.is\_open()) {

resultFile << "Встановлені вимоги порядку виконання лабораторної роботи порушено!";

resultFile.close();

} else {

std::cerr << "Не вдалося відкрити файл для запису.";

return 1;

}

} else {

ifstream inputFile(TEST\_SUITE\_FILE);

ofstream outputFile(TEST\_RESULTS\_FILE);

string line;

int testCase = 0;

while (getline(inputFile, line)) {

stringstream ss(line);

string color;

string pattern;

float height;

float radius;

getline(ss, color, '|');

getline(ss, pattern, '|');

ss >> height;

ss.ignore(1, '|');

ss >> radius;

ClassLab12\_Hrebeniuk testCap = ClassLab12\_Hrebeniuk(color, pattern, height, radius);

testCase += 1;

outputFile << "Тест №" << testCase << endl;

outputFile << "Колір ковпака: " << testCap.getColor() << endl;

outputFile << "Малюнок ковпака: " << testCap.getPattern() << endl;

outputFile << "Висота ковпака: " << testCap.getHeight() << endl;

outputFile << "Радіус основи ковпака: " << testCap.getRadius() << endl;

outputFile << "Об’єм ковпака: " << testCap.getVolume() << endl << endl;

}

cout << "Роботу застосунку завершено, результат збережений у файлі TestResults.txt" << endl;

system("pause");

inputFile.close();

outputFile.close();

}

return 0;

}

**Результати тестування у файлі TestResults.txt**

Тест №1

Колір ковпака: white

Малюнок ковпака: none

Висота ковпака: 0.1

Радіус основи ковпака: 0.1

Об’єм ковпака: 0.0010472

Тест №2

Колір ковпака: red

Малюнок ковпака: striped

Висота ковпака: 0.1

Радіус основи ковпака: 0.1

Об’єм ковпака: 0.0010472

Тест №3

Колір ковпака: white

Малюнок ковпака: none

Висота ковпака: 5

Радіус основи ковпака: 3

Об’єм ковпака: 47.1239

Тест №4

Колір ковпака: white

Малюнок ковпака: none

Висота ковпака: 0.1

Радіус основи ковпака: 0.1

Об’єм ковпака: 0.0010472

Тест №5

Колір ковпака: blue

Малюнок ковпака: none

Висота ковпака: 2

Радіус основи ковпака: 1

Об’єм ковпака: 2.0944

Тест №6

Колір ковпака: white

Малюнок ковпака: dotted

Висота ковпака: 0.1

Радіус основи ковпака: 0.1

Об’єм ковпака: 0.0010472

Тест №7

Колір ковпака: green

Малюнок ковпака: checkered

Висота ковпака: 10

Радіус основи ковпака: 5

Об’єм ковпака: 261.799

Тест №8

Колір ковпака: yellow

Малюнок ковпака: plain

Висота ковпака: 0.1

Радіус основи ковпака: 0.1

Об’єм ковпака: 0.0010472

Тест №9

Колір ковпака: purple

Малюнок ковпака: solid

Висота ковпака: 100

Радіус основи ковпака: 50

Об’єм ковпака: 261799

Тест №10

Колір ковпака: black

Малюнок ковпака: gradient

Висота ковпака: 1

Радіус основи ковпака: 0.1

Об’єм ковпака: 0.010472

**Висновок**

Виконуючи цю лабораторну роботу, я отримав ґрунтовні знання та практичний досвід роботи з програмною реалізацією абстрактних типів даних у мові програмування C++. Особливу увагу у цій лабораторній роботі було приділено роботі з файлами, зокрема перевірці місцезнаходження файлів проекту, запису та читанню даних, а також реалізації та тестуванню класів.

Згідно методичних рекомендацій потрібно було почергово проаналізувати і постановити задачі 12.1 та 12.2, проаналізувати вимоги до програмного забезпечення та вмісту вхідного файлу, спроектувати архітектуру класу та задокументувати результати у звіті. В цій лабораторній роботі особливу увагу було приділено перевірці правильного розташування файлів проекту та протоколюванню результатів тестування.

*Задача 12.1*

Для реалізації задачі 12.1 було створено клас ClassLab12\_Hrebeniuk, що представляє об’єкт "святковий ковпак". Клас містить атрибути кольору, малюнка, висоти та радіуса основи ковпака, а також методи для їх задання і отримання. Обчислення об’єму ковпака виконується за допомогою відповідної функції-члена класу.

*Задача 12.2*

Завдання полягало у створенні додатка Teacher, який перевіряє місцезнаходження файлу проекту main.cpp під час компіляції. Якщо файл знаходився не у вказаному каталозі, програма виводила 100 звукових сигналів та записувала повідомлення про порушення вимог у текстовий файл. Якщо файл знаходився у правильному каталозі, додаток створював об’єкт класу ClassLab12\_Hrebeniuk та виконував його unit-тестування за тест-сьютом, результати якого записувалися у текстовий файл TestResults.txt.

*Результати тестування*

Було реалізовано тестовий драйвер, що виконував протоколювання процесу тестування. Було розроблено кілька тестових кейсів, що включали різні комбінації значень атрибутів ковпака. Усі тестові кейси були успішно завершені зі статусом "passed".

Завдяки цій лабораторній роботі я покращив свої знання та практичні навички у проектуванні та розробці модульних проектів, роботі з файлами, використанні препроцесорних директив, та у написанні тестових драйверів. Мені було цікаво виконувати цю лабораторну роботу, дізнаватись нову інформацію та проектувати задачі такого рівня.

**Контрольні запитання і завдання**

**1.** *В чому полягає сутність процесу абстрагування як теоретичного метода пізнання на основі емпіричних даних, фактів тощо? Виходячи з означеного, сформулюйте визначення поняттю “абстракція”.*

Абстрагування є процесом виділення важливих властивостей і відношень об'єктів, ігноруючи другорядні та неістотні деталі, з метою створення спрощеної моделі реальності для легшого розуміння та аналізу. Абстракція – це теоретичний метод пізнання, який полягає у створенні моделей, що відображають лише суттєві характеристики об'єктів, відкидаючи другорядні.

**2.** *Що визначають й описують під час об’єктної декомпозиції задачі?*

Під час об’єктної декомпозиції задачі визначають і описують окремі об’єкти, їх властивості та взаємозв’язки, які необхідні для вирішення задачі. Це дозволяє структурувати задачу у вигляді ієрархії об’єктів, полегшуючи її розуміння та розв'язання.

**3.** *Сформулюйте сутність і перелічіть результати концептуалізації предметної області як метода пізнання (дослідження).*

Концептуалізація предметної області полягає у створенні концептуальної моделі, що включає основні поняття та відношення, які описують предметну область. Результати концептуалізації включають визначення ключових понять, встановлення їх взаємозв’язків, а також створення діаграм та моделей, які відображають структуру і динаміку предметної області.

**4.** *Що розуміють під аналізом сутності предметної області? Сформулюйте структуру або алгоритм об’єктного аналізу предметної області задачі.*

Аналіз сутності предметної області – це процес виявлення та опису основних елементів і взаємозв’язків, що складають предметну область. Структура об’єктного аналізу задачі може включати наступні етапи:

- Визначення мети та обсягу аналізу;

- Ідентифікація ключових об’єктів та їх властивостей;

- Встановлення відношень між об’єктами;

- Створення діаграм та моделей для візуалізації предметної області;

- Верифікація і корекція отриманих моделей.

**5.** *Сформулюйте методику створення класу ― реалізації абстрактного типу даних ― у заголовковому файлі С++.*

Методика створення класу у заголовковому файлі С++ включає наступні кроки:

- Оголосити заголовковий файл (.h);

- Визначити ім'я класу;

- Оголосити приватні дані-члени класу;

- Визначити конструктори, деструктори та інші методи класу у публічному розділі класу;

- Забезпечити доступ до приватних членів через публічні методи (гетери та сетери).

Приклад

#ifndef MYCLASS\_H

#define MYCLASS\_H

class MyClass {

private:

int data;

public:

MyClass();

~MyClass();

int getData() const;

void setData(int value);

};

#endif // MYCLASS\_H

**6.** *Що є результатом реалізації процесу визначення інтерфейсу об’єкта предметної області та яким чином він описується мовою програмування С++?*

Результатом реалізації процесу визначення інтерфейсу об’єкта є створення чітко визначених методів і властивостей, які дозволяють взаємодіяти з об’єктом без знання деталей його реалізації. В С++ інтерфейс описується за допомогою класів з чисто віртуальними методами.

Приклад:

class Interface {

public:

virtual void doSomething() = 0;

};

**7.** *В чому полягає мета перевантаження функцій під час реалізації проектних рішень мовою програмування С/С++?*

Мета перевантаження функцій полягає у можливості створення декількох варіантів однієї функції з різними параметрами. Це дозволяє підвищити гнучкість і зручність використання функцій, забезпечуючи різну поведінку залежно від типу або кількості аргументів.

**8.** *Яким є синтаксис запису перевантаження функцій мовою програмування С/С++?*

Синтаксис перевантаження функцій передбачає визначення декількох функцій з однаковим ім'ям, але з різними параметрами.

void print(int i);

void print(double flt);

void print(std::string str);

**9.** *Перелічіть випадки (задачі), за яких є доцільним використання препроцесорних макросів FILE, LINE, DATE, TIME, TIMESTAMP.*

Використання препроцесорних макросів є доцільним у наступних випадках:

- FILE і LINE для логування і відлагодження коду, щоб вказувати точне місце помилки.

- DATE і TIME для відмітки часу компіляції програми.

- TIMESTAMP для отримання останньої дати модифікації файлу.

**10.** *Яким є синтаксис запису функції з параметрами за замовчуванням мовою програмування С/С++?*

Функція з параметрами за замовчуванням має вигляд:

void display(int value = 10, char ch = 'a');

**11.** *Яке призначення препроцесорних директив #define, #undef, #if, #ifdef, #ifndef, #error, #pragma?*

- #define: Оголошення макросів.

- #undef: Скасування визначення макросу.

- #if: Умовна компіляція.

- #ifdef: Перевірка, чи макрос визначений.

- #ifndef: Перевірка, чи макрос не визначений.

- #error: Генерація повідомлення про помилку під час компіляції.

- #pragma: Специфічні інструкції для компілятора.

**12.** *В чому полягає концепція абстрактного типу даних (ADT) та чим він відрізняється від вбудованого типу С/С++?*

Абстрактний тип даних (ADT) визначає тип через його поведінку (операції), а не через реалізацію. Вбудовані типи C/С++ мають конкретну реалізацію, тоді як ADT може мати різні реалізації, приховані за інтерфейсом.

**13.** *Здійсніть порівняльний аналіз типів клас (class) та структура (struct) мови програмування С++.*

- Клас (class): за замовчуванням всі члени приватні.

- Структура (struct): за замовчуванням всі члени публічні.

- Обидва можуть мати методи і спадкування.

**14.** *Яким є синтаксис запису ADT клас (class) мовою програмування С++?*

class MyClass {

public:

virtual void function() = 0;

};

**15.** *Які призначення та синтаксис запису оператора :: С++? Наведіть приклад запису даних у глобальний об’єкт (змінну), ідентифікатор якого співпадає з ім’ям локального об’єкта (змінної).*

Оператор :: використовується для доступу до глобальних змінних або методів класу ззовні.

Приклад:

int value = 10; // глобальна змінна

void function() {

int value = 5; // локальна змінна

::value = 20; // запис у глобальну змінну

}

**16.** *Чим синтаксис оголошення даних-членів класу відрізняється від оголошення членів структури С++?*

Дані-члени класу за замовчуванням приватні, а члени структури – публічні.

**17.** *Якими є синтаксис і основні правила запису функцій-членів класу С++?*

Функції-члени класу оголошуються всередині класу та визначаються поза класом, використовуючи оператор ::

Приклад:

class MyClass {

public:

void myFunction();

};

void MyClass::myFunction() {

// реалізація

}

**18.** *Які розрізняють рівні (секції) доступу до членів класу С++ та за допомогою яких специфікаторів ці рівні визначаються?*

В С++ є три рівні доступу: public, protected, private. Специфікатори доступу визначаються відповідними ключовими словами.

**19.** *Яким є синтаксис оголошення (створення) об’єкта класу С++? Здійсніть порівняльний аналіз класу і об’єкта цього класу.*

Синтаксис створення об’єкта класу: MyClass obj;

Клас визначає структуру і поведінку, а об’єкт є конкретною реалізацією класу.

**20.** *Яке призначення та синтаксис запису спеціальних функційчленів класу С++ (конструктори, деструктори)?*

Конструктори використовуються для ініціалізації об’єктів, деструктори – для очищення ресурсів.

Приклад:

class MyClass {

public:

MyClass(); // конструктор

~MyClass(); // деструктор

};