МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

ЗВІТ

ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 1

з навчальної дисципліни

# “ ОБ’ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ ”

Основні поняття ООП. Класи та об’єкти. Функції доступу. Вказівник this

ВИКОНАВ

студент академічної групи КБ-23

Сергій Козирь

ПЕРЕВІРИВ

асистент кафедри кібербезпеки

та програмного забезпечення

Козірова Н.Л.

Кропивницький – 2024

ОБ’ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ

Лабораторна робота №1

Основні поняття ООП. Класи та об’єкти. Функції доступу. Вказівник this

Мета: ознайомитись з основними поняттями ООП. Вивчити поняття клас, об’єкт, сеттер, геттер та навчитись їх програмно реалізовувати мовою С++.

Варіант 1

Завдання 1

1. Напишіть клас "Book" для представлення книги.

Клас повинен мати наступні властивості та функціональність:

Приватні поля класу:

• title (назва книги)

• author (автор книги)

• year (рік видання книги)

Публічні методи класу:

• Метод setTitle(), який дозволяє задати назву книги.

• Метод getTitle(), який повертає назву книги.

• Метод setAuthor(), який дозволяє задати автора книги.

• Метод getAuthor(), який повертає автора книги.

• Метод setYear(), який дозволяє задати рік видання книги.

• Метод getYear(), який повертає рік видання книги.

2. Створіть об'єкт класу "Book".

3. Задайте значення полів об'єкта за допомогою відповідних методів.

4. Виведіть інформацію про книгу на екран, використовуючи методи доступу до полів.

5. Реалізувати програму за допомогою роздільної компіляції. У вашому рішенні можуть бути додаткові методи та поля, якщо ви вважаєте їх необхідними.

Завдання 2 Реалізувати вище наведену задачу за допомогою структурного програмування. У висновку описати різницю цих методів.

**Завдання 1**

**1**

// main.cpp

#include <iostream>

#include "Book.h"

int main() {

setlocale(LC\_ALL, "ukr");

// Створення об'єкта класу Book

Book myBook;

// Задання значень полів

myBook.setTitle("The Great Gatsby");

myBook.setAuthor("F. Scott Fitzgerald");

myBook.setYear(1925);

// Виведення інформації про книгу

std::cout << "Назва книги: " << myBook.getTitle() << std::endl;

std::cout << "Автор книги: " << myBook.getAuthor() << std::endl;

std::cout << "Рік видання: " << myBook.getYear() << std::endl;

return 0;

}

**2**

// Book.cpp

#include "Book.h"

// Конструктор за замовчуванням

Book::Book() : title(""), author(""), year(0) {}

// Конструктор з параметрами

Book::Book(std::string t, std::string a, int y) : title(t), author(a), year(y) {}

// Методи доступу до полів

void Book::setTitle(const std::string& t) {

title = t;

}

std::string Book::getTitle() const {

return title;

}

void Book::setAuthor(const std::string& a) {

author = a;

}

std::string Book::getAuthor() const {

return author;

}

void Book::setYear(int y) {

year = y;

}

int Book::getYear() const {

return year;

}

**3**

// Book.h

#ifndef BOOK\_H

#define BOOK\_H

#include <string>

class Book {

private:

std::string title;

std::string author;

int year;

public:

// Конструктори

Book(); // Конструктор за замовчуванням

Book(std::string title, std::string author, int year); // Конструктор з параметрами

// Методи доступу

void setTitle(const std::string& title);

std::string getTitle() const;

void setAuthor(const std::string& author);

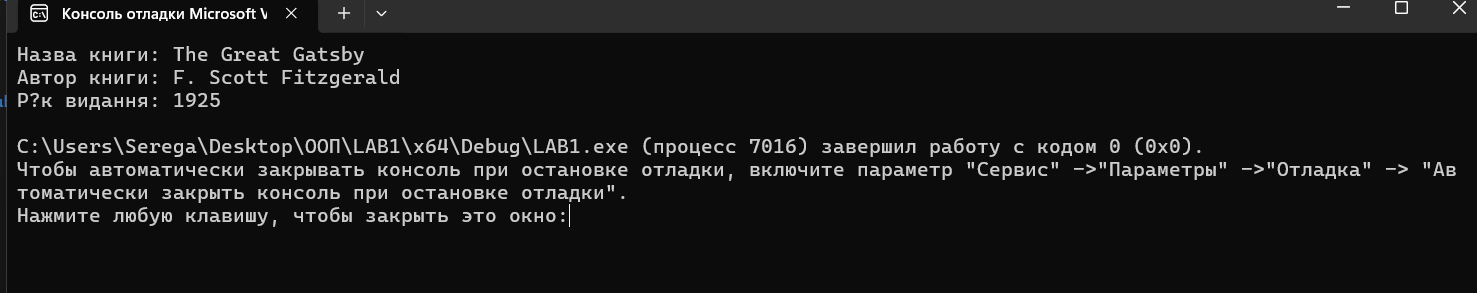
std::string getAuthor() const;

void setYear(int year);

int getYear() const;

};

#endif

****

**Завдання 2**

#include <iostream>

#include <string>

struct Book {

std::string title;

std::string author;

int year;

};

// Функції для роботи зі структурою

void setTitle(Book& book, const std::string& title) {

book.title = title;

}

std::string getTitle(const Book& book) {

return book.title;

}

void setAuthor(Book& book, const std::string& author) {

book.author = author;

}

std::string getAuthor(const Book& book) {

return book.author;

}

void setYear(Book& book, int year) {

book.year = year;

}

int getYear(const Book& book) {

return book.year;

}

int main() {

setlocale(LC\_ALL, "ukr");

// Створення об'єкта структури Book

Book myBook;

// Задання значень полів

setTitle(myBook, "The Great Gatsby");

setAuthor(myBook, "F. Scott Fitzgerald");

setYear(myBook, 1925);

// Виведення інформації про книгу

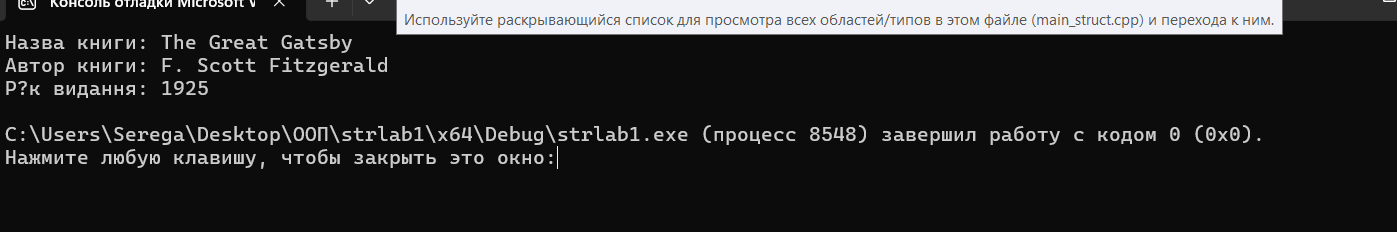
std::cout << "Назва книги: " << getTitle(myBook) << std::endl;

std::cout << "Автор книги: " << getAuthor(myBook) << std::endl;

std::cout << "Рік видання: " << getYear(myBook) << std::endl;

return 0;

}

****

**Висновок**

**Класове програмування (ОOP):**

1. **Інкапсуляція**: Дані захищені (приватні поля) та доступні тільки через публічні методи.
2. **Гнучкість**: Можна додавати додаткову функціональність без змін зовнішнього інтерфейсу.
3. **Модульність**: Код легко підтримувати і розширювати через роздільну компіляцію.

**Структурне програмування:**

1. **Простота**: Прямий доступ до полів структури, що може бути корисно для простих задач.
2. **Менше захисту**: Поля не захищені, що може призвести до некоректних змін в даних.
3. **Менше абстракції**: Важче додавати нову функціональність і підтримувати великий код.

Використання класів надає кращу структуризацію, інкапсуляцію і полегшує подальший розвиток програми, тоді як структурний підхід є простішим, але менш захищеним та гнучким.