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”

На тему:

“Програмна реалізація абстрактних типів даних”

ВИКОНАВ

студент академічної групи КІ-23

Замай В.А

ПЕРЕВІРИВ

ст. викладач кафедри кібербезпеки   
Дрєєва Г.М.

м. Кропивницький 2024

**Мета роботи :**

Полягає у набутті 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 :**

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:** №20

**Аналіз виконання Задачі 12.1:**

1. Атрибути книги:
   * Назва книги (title): рядок
   * Автор книги (author): рядок
   * Кількість сторінок (pages): ціле число
   * Довжина палітурки (length): дійсне число
   * Ширина палітурки (width): дійсне число
2. Конструктор:
   * Ініціалізує всі атрибути книги під час створення об'єкта.
3. Гетери:
   * Повертають значення приватних атрибутів.
4. Сетер:
   * Дозволяють змінювати значення атрибутів з перевіркою вхідних даних.
5. Обчислення площі палітурки:
   * Метод, який обчислює площу палітурки за формулою S= a×b.

**Лістинг main.cpp**

#include <iostream>

#include <string>

#include <cassert>

#include <fstream>

class ClassLab12\_Zamai {

private:

std::string title;

std::string author;

int pages;

double length; // довжина палітурки

double width; // ширина палітурки

public:

ClassLab12\_Zamai(std::string t, std::string a, int p, double l, double w)

: title(t), author(a), pages(p), length(l), width(w) {}

std::string getTitle() const {

return title;

}

std::string getAuthor() const {

return author;

}

int getPages() const {

return pages;

}

double getLength() const {

return length;

}

double getWidth() const {

return width;

}

double getCoverArea() const {

return length \* width;

}

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

title = t;

}

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

author = a;

}

void setPages(int p) {

if (p > 0) {

pages = p;

} else {

std::cerr << "Number of pages must be positive.\n";

}

}

void setLength(double l) {

if (l > 0) {

length = l;

} else {

std::cerr << "Length must be positive.\n";

}

}

void setWidth(double w) {

if (w > 0) {

width = w;

} else {

std::cerr << "Width must be positive.\n";

}

}

};

void testGetters(const ClassLab12\_Zamai& book, std::ofstream& log) {

log << "Running testGetters...\n";

assert(book.getTitle() == "Программирование. Принципы и практика с использованием C++");

assert(book.getAuthor() == "Бьярне Страуструп");

assert(book.getPages() == 1360);

assert(book.getLength() == 25.0);

assert(book.getWidth() == 20.0);

assert(book.getCoverArea() == 25.0 \* 20.0);

log << "testGetters passed\n";

}

void testSetters(ClassLab12\_Zamai& book, std::ofstream& log) {

log << "Running testSetters...\n";

book.setTitle("Программирование. Принципы и практика с использованием C++");

book.setAuthor("Бьярне Страуструп");

book.setPages(1400);

book.setLength(26.0);

book.setWidth(21.0);

assert(book.getTitle() == "Программирование. Принципы и практика с использованием C++");

assert(book.getAuthor() == "Бьярне Страуструп");

assert(book.getPages() == 0);

assert(book.getLength() == 26.0);

assert(book.getWidth() == 21.0);

assert(book.getCoverArea() == 26.0 \* 21.0);

log << "testSetters passed\n";

}

void testInvalidInputs(ClassLab12\_Zamai& book, std::ofstream& log) {

log << "Running testInvalidInputs...\n";

book.setPages(-50);

assert(book.getPages() == 1400);

book.setLength(-30.0);

assert(book.getLength() == 26.0);

book.setWidth(-15.0);

assert(book.getWidth() == 21.0); // width should remain unchanged

log << "testInvalidInputs passed\n";

}

int main() {

std::ofstream log("test\_results.txt");

if (!log.is\_open()) {

std::cerr << "Unable to open file for writing test results.\n";

return 1;

}

ClassLab12\_Zamai book("Программирование. Принципы и практика с использованием C++", "Бьярне Страуструп", 1360, 25.0, 20.0);

testGetters(book, log);

testSetters(book, log);

testInvalidInputs(book, log);

log << "All tests passed!\n";

log.close();

return 0;

}

**Висновок**

Реалізація класу ClassLab12\_Zamai дозволяє ефективно моделювати книгу з її основними характеристиками: назвою, автором, кількістю сторінок та розмірами палітурки. Клас забезпечує доступ до цих атрибутів через гетери, а також дозволяє змінювати їх за допомогою сетерів з перевіркою вхідних даних. Крім того, клас надає метод для обчислення площі палітурки книги.

Клас відповідає принципам об'єктно-орієнтованого програмування, зокрема інкапсуляції, оскільки атрибути оголошені як приватні, а доступ до них здійснюється через публічні методи. Сетери забезпечують перевірку вхідних даних, що дозволяє уникнути некоректних значень атрибутів.

Загалом, клас інкапсулює всі дані та методи, пов'язані з моделлю книги, що спрощує його використання і підтримку. Такий підхід робить код більш зрозумілим, надійним та легким для розширення в майбутньому.