Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

Кафедра кібербезпеки та програмного забезпечення

**Лабораторна робота №3**

**Тема:** «Успадкування, агрегація та композиція»

|  |
| --- |
| Виконав: ст. гр. КН-23 |
| Гончаренко В.В |
| Перевірив:  Казірова Л.В. |
|  |

Кропивницький

2024

**ОБ’ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ**

**Лабораторна робота №3**

**Успадкування, агрегація та композиція**

Мета: ознайомитись з основними поняттями успадкування, агрегація та композиція в ООП та навчитись їх програмно реалізовувати мовою С++.

**Завдання 1**

Варіант 2

1. Створіть клас "Бібліотека", який містить такі властивості:

• Назва бібліотеки

• Адреса бібліотеки

• Список книг, які є в наявності в бібліотеці (використовуйте агрегацію)

• Клас "Бібліотекар", який представляє бібліотекаря, що працює в бібліотеці (використовуйте композицію)

2. Створіть клас "Книга", який містить такі властивості:

• Назва книги

• Автор книги

• Рік видання

3. Створіть клас "Бібліотекар", який містить такі властивості:

• Ім'я бібліотекаря

• Посада бібліотекаря

4. Реалізуйте методи для додавання та видалення книг зі списку книг у бібліотеці.

5. Напишіть демонстраційну функцію main(), в якій створюються об'єкти класів "Бібліотека", "Книга" та "Бібліотекар". Додайте книги до списку книг у бібліотеці та виведіть інформацію про них.

6. Продемонструйте успадкування, створивши похідний клас від "Книга" (наприклад, "Роман") з додатковими властивостями та методами.

7. Розширте функціональність, додавши додаткові методи та властивості до класів за власним бажанням. Не забудьте додати коментарі та пояснення до коду, щоб роз'яснити його функціональність та зрозумілість.

**-- Лістинг --**

#ifndef LIBRARYSYSTEM\_H

#define LIBRARYSYSTEM\_H

#include <iostream>

#include <string>

#include <vector>

using namespace std;

class Librarian {

private:

string name;

string position;

public:

Librarian(string name, string position)

: name(name), position(position) {}

void displayInfo() {

cout << "Librarian: " << name << ", Position: " << position << endl;

}

};

class Book {

protected:

string title;

string author;

int publicationYear;

public:

Book(string title, string author, int publicationYear)

: title(title), author(author), publicationYear(publicationYear) {}

virtual void displayInfo() {

cout << "Title: " << title << ", Author: " << author << ", Year: " << publicationYear << endl;

}

string getTitle() {

return title;

}

};

class Novel : public Book {

private:

string genre;

public:

Novel(string title, string author, int publicationYear, string genre)

: Book(title, author, publicationYear), genre(genre) {}

void displayInfo() override {

cout << "Novel - Title: " << title << ", Author: " << author

<< ", Year: " << publicationYear << ", Genre: " << genre << endl;

}

};

class Library {

private:

string name;

string address;

vector<Book\*> books;

Librarian librarian;

public:

Library(string name, string address, Librarian librarian)

: name(name), address(address), librarian(librarian) {}

void addBook(Book\* book) {

books.push\_back(book);

}

void removeBook(const string& bookTitle) {

for (size\_t i = 0; i < books.size(); ++i) {

if (books[i]->getTitle() == bookTitle) {

books.erase(books.begin() + i);

break;

}

}

}

void displayInfo() {

cout << "Library: " << name << ", Address: " << address << endl;

librarian.displayInfo();

cout << "Books available in the library:" << endl;

for (Book\* book : books) {

book->displayInfo();

}

}

};

#endif

**File main.cpp**

#include "Library.h"

int main() {

Librarian librarian("Bob Sherlock", "Head librarian");

Library library("Library named after Volodymyr Panchenko", "St. Andriy Matvienko, 1-b", librarian);

Book\* book1 = new Book("Charlie and the Chocolate Factory", "Roald Dahl", 1995);

Novel\* novel1 = new Novel("Anna Karenina", "Leo Tolstoy", 1877, "Classical Novel");

library.addBook(book1);

library.addBook(novel1);

library.displayInfo();

library.removeBook("Anna Karenina");

std::cout << "After removing a book:" << std::endl;

library.displayInfo();

// Освобождение памяти

delete book1;

delete novel1;

return 0;

}

**Завдання 1**

Створіть використовуючи композицію два класи, які матимуть свої властивості, перший реалізуйте за допомогою включення (composition), а другий за допомогою посилання (reference), контейнером для цих класів використайте клас з лабораторної роботи №1, створіть також третій – похідний клас, використовуючи наслідування, базовим класом може бути або клас з лр№1, або класи які ви створили за допомогою композиції. За бажанням можете зобразити схематично як ваші класи залежать один від одного.

**-- Лістинг --**

#include <iostream>

#include <string>

#include "BankAccount.h"

#include "BankAccount.cpp"

using namespace std;

class Person {

private:

BankAccount bankAccount;

string name;

public:

Person(const string &personName, const string &accNumber, double initialBalance) {

name = personName;

bankAccount.setOwnerName(personName);

bankAccount.setAccountNumber(accNumber);

bankAccount.setBalance(initialBalance);

}

void displayPersonInfo() const {

cout << "Ім'я: " << name << endl;

bankAccount.displayAccountInfo();

}

};

class Company {

private:

BankAccount &bankAccount;

string companyName;

public:

Company(BankAccount &account, const string &name) : bankAccount(account) {

companyName = name;

}

void displayCompanyInfo() const {

cout << "Назва компанії: " << companyName << endl;

bankAccount.displayAccountInfo();

}

};

class VipPerson : public Person {

private:

double bonusBalance;

public:

VipPerson(const string &personName, const string &accNumber, double initialBalance, double bonus)

: Person(personName, accNumber, initialBalance), bonusBalance(bonus) {}

void displayVIPInfo() const {

displayPersonInfo();

cout << "Бонусний баланс: " << bonusBalance << endl;

}

};

int main() {

BankAccount companyAccount;

companyAccount.setOwnerName("Owner Company");

companyAccount.setAccountNumber("10987654321");

companyAccount.setBalance(50000.0);

Person person1("Vlad Honcharenko", "987654321", 10000.0);

person1.displayPersonInfo();

Company company1(companyAccount, "TechCorp");

company1.displayCompanyInfo();

VipPerson vipPerson1("Mark Zuck", "1122334455", 5000.0, 200.0);

vipPerson1.displayVIPInfo();

return 0;

}