Міністерство освіти та науки України

Інститут «Телекомунікації, радіоелектроніки та радіотехніки»

Изображение выглядит как текст, круг, Шрифт, эмблема

Автоматически созданное описание

Звіт

З лабораторної роботи №11

З дисципліни «Об`єктно-орієнтоване програмування»

На тему: «Конструювання класів з використанням віртуальних функцій»

Виконав:

Студент групи АП-22

Іщак Д.А,

Прийняв:

Алтунін С. І

Львів 2024

**Мета:** створити, відлагодити та протестувати програму, у якій створити базовий та похідні класи з використанням віртуальних функцій для обробки даних.

**Хід роботи:**

1. Вибрати завдання згідно свого варіанту у ДОДАТКУ 4.Ознайомитися із властивостями віртуальних функцій.

2. Обдумати способи їх використання для створення програм.

3. Створити батьківський та дочірні класи згідно свого варіанту.

4. Застосувати віртуальні функції (методи) для виведення даних із дочірніх класів.

5. Розробити блок-схему алгоритму програми.

6. Реалізувати програму для вводу і виводу даних полів (членів) структур, вкладених у клас.

**Варіант завдання:**

****

**Код програми:**

**#include <iostream>**

**#include <string>**

**#include <vector>**

**class Base {**

**public:**

**virtual void printData() const { // Віртуальна функція**

**std::cout << "Base class data." << std::endl;**

**}**

**virtual ~Base() {} // Віртуальний деструктор**

**};**

**class Derived1 : public Base {**

**public:**

**void printData() const override { // Перевизначення віртуальної функції**

**std::cout << "Derived1 class data." << std::endl;**

**}**

**};**

**class Derived2 : public Base {**

**public:**

**void printData() const override { // Перевизначення віртуальної функції**

**std::cout << "Derived2 class data." << std::endl;**

**}**

**};**

**// Клас з вкладеними структурами**

**class ComplexClass {**

**public:**

**struct Member {**

**int id;**

**std::string name;**

**};**

**ComplexClass(int id, std::string name) {**

[**member.id**](https://web.telegram.org/a/member.id) **= id;**

[**member.name**](https://web.telegram.org/a/member.name) **= name;**

**}**

**void printMember() const {**

**std::cout << "ID: " <<** [**member.id**](https://web.telegram.org/a/member.id) **<< ", Name: " <<** [**member.name**](https://web.telegram.org/a/member.name) **<< std::endl;**

**}**

**private:**

**Member member;**

**};**

**int main() {**

**// Створюємо масив з базовим класом для поліморфізму**

**std::vector<Base\*> objects;**

**objects.push\_back(new Derived1());**

**objects.push\_back(new Derived2());**

**// Викликаємо віртуальні функції**

**for (auto obj : objects) {**

**obj->printData();**

**}**

**// Створюємо об'єкти складних класів**

**ComplexClass complexObj1(1, "Аліса");**

**ComplexClass complexObj2(2, "Володимир");**

**// Виводимо дані членів структур**

**complexObj1.printMember();**

**complexObj2.printMember();**

**// Очищаємо пам'ять**

**for (auto obj : objects) {**

**delete obj;**

**}**

**return 0;**

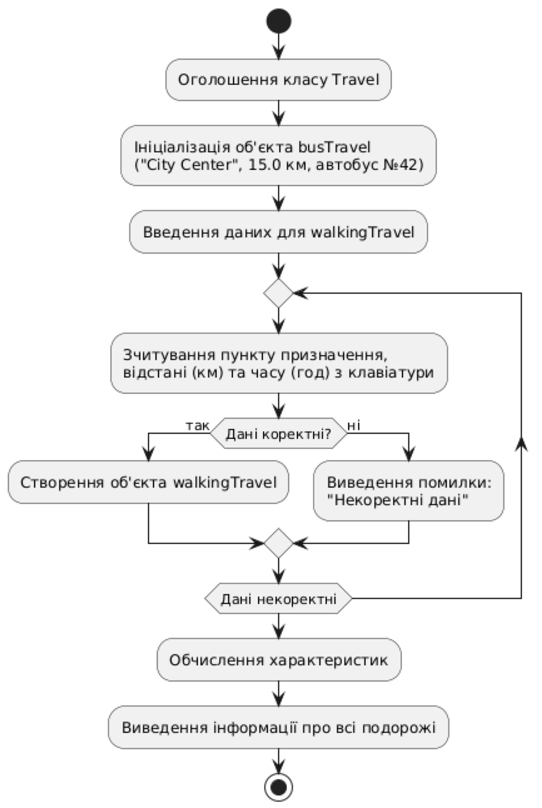
**}**

**Результати виконання роботи:**

**Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание**

**Блок-схема програми:**

****

Висновок:

У цій лабораторній роботі реалізовано систему класів для опису подорожей, зокрема, пішохідних та автобусних. Використано базовий клас "Travel" з методами та атрибутами, спільними для всіх типів подорожей, а також дочірні класи "WalkingTravel" і "BusTravel", що додають унікальні характеристики для кожного виду подорожі. Реалізовано функції для ручного введення даних із перевіркою валідності, що підвищує надійність програми. Дана структура дозволяє ефективно створювати та обробляти інформацію про подорожі, забезпечуючи зручність і захист від некоректних даних.