Міністерство освіти та науки України

Інститут «Телекомунікації, радіоелектроніки та радіотехніки»

Изображение выглядит как текст, круг, Шрифт, эмблема

Автоматически созданное описание

Звіт

З лабораторної роботи №9

З дисципліни «Об`єктно-орієнтоване програмування»

На тему: «Конструювання класів з використанням дружніх 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 для обробки даних.

**Хід роботи:**

1. продумайте, які типи змінних вам потрібно використати для виконання завдання.

2. створіть проєкт із кількома файлами, тобто крім основного файлу, наприклад main.cpp, додайте для кожного класу заголовний файл .h та виконуваний файл .cpp. Також модифікуйте ..\.vscode\\tasks.json, щоб вказати йому шлях до всіх файлів проекту.

3. оголосіть базовий (батьківський) та похідний (дочірній) класи, а також відповідні структури для збереження і обробки даних, конструктори та деструктори.

4. у заголовному файлі дружнього класу оголосіть конструктори та деструктори, методи для введення, виведення та обробки даних із батьківського та дочірнього класів.

5. у виконувану файлі дружнього класу реалізуйте для введення, виведення та обробки даних із батьківського та дочірнього класі.

6. можете збільшити кількість членів класу додатковими полями чи методами.

7. обчислити результат відповідно заданого варіанту, наприклад загальну вартість, тривалість, кількість, максимальне чи мінімальне значення тощо.

8. програма має вивести на консоль (термінал) вхідні дані та результат виконання.

**Варіант завдання:**

****

**Код програми:**

#include <iostream>

#include <string>

#include <limits>

// Базовий клас

class Product {

protected:

std::string name; // Назва товару

int quantity; // Кількість товару

public:

Product(const std::string& name = "", int quantity = 0)

: name(name), quantity(quantity) {}

// Методи для вводу даних

virtual void input() {

std::cout << "Введіть назву товару: ";

std::cin >> name;

while (true) {

std::cout << "Введіть кількість товару: ";

std::cin >> quantity;

if (std::cin.fail() || quantity < 0) {

std::cout << "Некоректне значення. Спробуйте ще раз.\n";

std::cin.clear();

std::cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n');

} else break;

}

}

// Метод для виводу даних

virtual void display() const {

std::cout << "Назва: " << name << ", Кількість: " << quantity;

}

};

// Дочірній клас

class PricedProduct : public Product {

private:

struct Details { // Вкладена структура

double price; // Ціна за одиницю

} details;

public:

PricedProduct(const std::string& name = "", int quantity = 0, double price = 0.0)

: Product(name, quantity) {

details.price = price;

}

// Перевизначений метод для вводу даних

void input() override {

Product::input(); // Викликаємо метод базового класу

while (true) {

std::cout << "Введіть ціну за одиницю товару: ";

std::cin >> details.price;

if (std::cin.fail() || details.price < 0) {

std::cout << "Некоректне значення. Спробуйте ще раз.\n";

std::cin.clear();

std::cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n');

} else break;

}

}

// Перевизначений метод для виводу даних

void display() const override {

Product::display(); // Викликаємо метод базового класу

std::cout << ", Ціна за одиницю: " << details.price

<< ", Загальна вартість: " << calculateTotal() << '\n';

}

// Метод для обчислення загальної вартості

double calculateTotal() const {

return quantity \* details.price;

}

};

int main() {

// Створення об'єкта дочірнього класу

PricedProduct product;

std::cout << "Введіть дані про товар:\n";

product.input(); // Введення даних користувачем

std::cout << "\nІнформація про товар:\n";

product.display(); // Вивід даних на екран

return 0;

}

#endif

**Результати виконання роботи:** **Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание**

**Блок-схема програми:**

Изображение выглядит как текст, снимок экрана, Шрифт, диаграмма

Автоматически созданное описание

Висновок:

У цій лабораторній роботі реалізовано базовий клас Trip, два похідні класи (WalkingTrip і BusTrip) та дружній клас TripFriend для обробки даних про подорожі. Програма демонструє використання спадкування для зручного управління різними типами подорожей, а також застосування дружніх класів для доступу до закритих даних. Отримані знання про об’єктно-орієнтоване програмування, зокрема спадкування, поліморфізм і дружні класи, дозволяють створювати модульні програми, які легко розширювати та підтримувати. Реалізація багатофайлової структури проєкту також сприяє більш організованому підходу до розробки складних програм.