Міністерство освіти та науки України

Інститут «Телекомунікації, радіоелектроніки та радіотехніки»

Изображение выглядит как текст, круг, Шрифт, эмблема

Автоматически созданное описание

Звіт

З лабораторної роботи №12

З дисципліни «Об`єктно-орієнтоване програмування»

На тему: «Робота з файлами та винятками»

Виконав:

Студент групи АП-22

Іщак Д.А.

Прийняв:

Алтунін С. І

Львів 2024

**Мета:** створити, відлагодити та протестувати програму, у якій реалізувати двонаправлений файловий ввід/вивід для обробки даних з використанням обробників винятків.

**Хід роботи:**

1. Вибрати завдання згідно свого варіанту у ДОДАТКУ 4.

2. Ознайомитися із винятками та їх обробкою. Обдумати способи їх використання для створення програм.

3. Ознайомитися із потоками вводу/виводу даних у файл. Обдумати способи їх використання для створення програм.

4. Створити батьківський та дочірні класи згідно свого варіанту.

5. Застосувати винятки та їх обробники для перевірки введення даних, відкриття файлів тощо.

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

**Варіант завдання:**



**Код програми:**

#include <iostream>

#include <fstream>

#include <string>

#include <stdexcept>

#include <vector>

class Animal {

public:

virtual void feed() const = 0; // Віртуальна функція для годування

virtual void print() const = 0; // Віртуальна функція для виведення даних

virtual ~Animal() {}

};

class Predator : public Animal {

public:

void feed() const override {

std::cout << "I am a predator, I eat meat." << std::endl;

}

void print() const override {

std::cout << "I am a predator." << std::endl;

}

};

class Herbivore : public Animal {

public:

void feed() const override {

std::cout << "I am a herbivore, I eat plants." << std::endl;

}

void print() const override {

std::cout << "I am a herbivore." << std::endl;

}

};

class AnimalCollection {

std::vector<Animal\*> animals;

public:

void addAnimal(Animal\* animal) {

animals.push\_back(animal);

}

void printAll() const {

for (const auto& animal : animals) {

animal->print();

}

}

~AnimalCollection() {

for (auto animal : animals) {

delete animal;

}

}

};

void openFile(const std::string& filename) {

std::ifstream file(filename);

if (!file) {

throw std::runtime\_error("File not found!");

}

std::cout << "File opened successfully." << std::endl;

}

int main() {

try {

AnimalCollection collection;

collection.addAnimal(new Predator());

collection.addAnimal(new Herbivore());

collection.printAll();

// Перевірка файлу

openFile("data.txt");

} catch (const std::runtime\_error& e) {

std::cerr << "Error: " << e.what() << std::endl;

}

return 0;

}

**Результати виконання роботи:**

**Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание**

**Блок-схема програми:**

**Изображение выглядит как текст, диаграмма, чек, снимок экрана

Автоматически созданное описание**

Висновок:

У цій лабораторній роботі реалізовано батьківський клас "Подорож" із двома дочірніми класами: "Пішохідна подорож" та "Автобусна подорож". Для кожного класу створено конструктори, методи введення, перевірки даних на валідність, виводу інформації та збереження даних у файл. Програма дозволяє ефективно вводити дані про подорожі, зберігати їх у файл і зчитувати з нього. Використання перевірки валідності даних забезпечує коректну роботу програми, уникаючи помилок через некоректні введені значення. Також було застосовано механізм обробки винятків для управління помилками вводу та роботи з файлами. Додатково реалізовано використання віртуальних функцій і багатофайлову структуру проєкту, що спрощує модифікацію коду та його масштабування.