Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет прикладної математики
Кафедра системного програмування і спеціалізованих комп’ютерних систем

# ЗВІТ

# з Домашньої Контрольної Роботи

*з дисципліни: Об’єктно-орієнтоване програмування*

Виконала студентка КВ-34

Фіалковська Ольга

16 варіант - 1 Варіант

Київ

КПІ ім. Ігоря Сікорського

2024

Одне практичне завдання: побудова ієрархії класів згідно із індивідуальним завданням. Обов’язковим є використання
ієрархій поліморфних класів. Особливу увагу необхідно приділити розробці інтерфейсів, зокрема, константності параметрів та методів. Клас-колекцію для збереження елементів реалізувати самостійно у вигляді шаблону класу.
Варіант 1.
Потяг (номер потягу; тип потягу: вантажний/пасажирський/змішаний;
кількість вагонів, вивід інформації про всі вагони), локомотив (тип: дизель/електричний, потужність), вантажний вагон (вантажопідйомність, маса вантажу, тип вантажу), пасажирський вагон (кількість місць, кількість пасажирів).
Короткі
Створена ієрархія класів відображає реальні взаємозв'язки між об'єктами: потягом, локомотивом, і вагонами. Використання базового класу `Wagon` забезпечує поліморфізм і легке розширення системи.
Використано основні принципи об'єктно-орієнтованого програмування: наслідування, інкапсуляція, поліморфізм і композиція. Це сприяє гнучкості та масштабованості програми.
Реалізовано шаблон класу `Collection`, який дозволяє універсально працювати з колекціями об'єктів, забезпечуючи їх повторне використання.
Код:
-Collection.h

```
#ifndef COLLECTION_H
#define COLLECTION_H

#include <vector>
#include <iostream>

template<typename T>
class Collection {
private:
    std::vector<T> items;
public:
    void add(const T& item) {
        items.push_back(item);
    }
```

```
    void displayAll() const {
        for (const auto& item : items) {
            item.displayInfo();
        }
    }
};

#endif // COLLECTION_H
```

-Locomotive.h

```
#ifndef LOCOMOTIVE_H
#define LOCOMOTIVE_H

#include <string>
#include <iostream>

class Locomotive {
private:
    std::string type;
    int power;
public:
    Locomotive(const std::string& type, int power);
    void displayInfo() const;
};

#endif // LOCOMOTIVE_H
```

-Locomotive.cpp

```
#include "Locomotive.h"

// Реалізація Locomotive
Locomotive::Locomotive(const std::string& type, int power)
    : type(type), power(power) {}

void Locomotive::displayInfo() const {
    std::cout << "Locomotive - Type: " << type << ", Power: " <<
power << " kW\n";
```

```
}
```

-Train.h

```
#ifndef TRAIN_H
#define TRAIN_H

#include <vector>
#include <memory>
#include "Wagon.h"
#include "Locomotive.h"

class Train {
private:
    int trainNumber;
    std::string trainType;
    Locomotive locomotive;
    std::vector<std::shared_ptr<Wagon>> wagons;
public:
    Train(int trainNumber, const std::string& trainType, const
Locomotive& locomotive);
    void addWagon(const std::shared_ptr<Wagon>& wagon);
    void displayInfo() const;
};

#endif // TRAIN_H
```

-Train.cpp

```
#include "Train.h"

// Реалізація Train
Train::Train(int trainNumber, const std::string& trainType, const
Locomotive& locomotive)
    : trainNumber(trainNumber), trainType(trainType),
locomotive(locomotive) {}

void Train::addWagon(const std::shared_ptr<Wagon>& wagon) {
    wagons.push_back(wagon);
```

```
}

void Train::displayInfo() const {
    std::cout << "Train Number: " << trainNumber << ", Type: " <<
trainType << "\n";
    locomotive.displayInfo();
    std::cout << "Wagons:\n";
    for (const auto& wagon : wagons) {
        wagon->displayInfo();
    }
}
```

-Wagon.h

```
#ifndef WAGON_H
#define WAGON_H

#include <string>
#include <iostream>

// Базовий клас Wagon
class Wagon {
public:
    virtual void displayInfo() const = 0; // Чисто віртуальний метод
    virtual ~Wagon() = default;
};

// Вантажний вагон
class FreightWagon : public Wagon {
private:
    double loadCapacity;
    double cargoWeight;
    std::string cargoType;
public:
    FreightWagon(double loadCapacity, double cargoWeight, const
std::string& cargoType);
    void displayInfo() const override;
```

```
};

// Пасажирський вагон
class PassengerWagon : public Wagon {
private:
    int seats;
    int passengers;
public:
    PassengerWagon(int seats, int passengers);
    void displayInfo() const override;
};

#endif // WAGON_H
```

-Wagon.cpp

```
#include "Wagon.h"

// Реалізація FreightWagon
FreightWagon::FreightWagon(double loadCapacity, double
cargoWeight, const std::string& cargoType)
    : loadCapacity(loadCapacity), cargoWeight(cargoWeight),
cargoType(cargoType) {}

void FreightWagon::displayInfo() const {
    std::cout << "Freight Wagon - Load Capacity: " << loadCapacity
              << " tons, Cargo Weight: " << cargoWeight
              << " tons, Cargo Type: " << cargoType << "\n";
}

// Реалізація PassengerWagon
PassengerWagon::PassengerWagon(int seats, int passengers)
    : seats(seats), passengers(passengers) {}

void PassengerWagon::displayInfo() const {
    std::cout << "Passenger Wagon - Seats: " << seats
              << ", Passengers: " << passengers << "\n";
}
```

-main.cpp

```
#include <iostream>
#include <memory>
#include "Train.h"
#include "Wagon.h"

int main() {
    Locomotive locomotive("Diesel", 3000);

    Train train(12345, "Mixed", locomotive);

    train.addWagon(std::make_shared<FreightWagon>(50.0, 30.0,
"Coal"));
    train.addWagon(std::make_shared<PassengerWagon>(100, 80));

    train.displayInfo();

    return 0;
}
```

Вивід:

```
apple — TrainApp — 80×24
Last login: Sun Dec 29 15:24:06 on ttys004
apple@MacBook-Air-apple ~ % /Users/apple/programming/DKR\ oop/TrainApp ; exit;
Train Number: 12345, Type: Mixed
Locomotive - Type: Diesel, Power: 3000 kW
Wagons:
Freight Wagon - Load Capacity: 50 tons, Cargo Weight: 30 tons, Cargo Type: Coal
Passenger Wagon - Seats: 100, Passengers: 80

Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.

[Процесс завершен]
```

```
ПРОБЛЕМЫ  ВЫХОДНЫЕ ДАННЫЕ  КОНСОЛЬ ОТЛАДКИ  ТЕРМИНАЛ  ПОРТЫ

apple@MacBook-Air-apple DKR oop % cd "/Users/apple/programming/DKR oop/" && g++ -st
ogramming/DKR oop/"#include <iostream>
zsh: parse error near `&&'
apple@MacBook-Air-apple DKR oop % cd "/Users/apple/programming/" && g++ -std=c++14
Train Number: 12345, Type: Mixed
Locomotive - Type: Diesel, Power: 3000 kW
Wagons:
Freight Wagon - Load Capacity: 50 tons, Cargo Weight: 30 tons, Cargo Type: Coal
Passenger Wagon - Seats: 100, Passengers: 80
apple@MacBook-Air-apple programming %
```