**ООП**

**Бурлаченко Єгор**

Лабораторна робота 7

Код:

#include <iostream>

#include <string>

class Employee {

public:

std::string lastName;

std::string firstName;

std::string middleName;

std::string position;

int experience;

double salary;

// Конструктор

Employee(std::string last, std::string first, std::string middle, std::string pos, int exp, double sal)

: lastName(last), firstName(first), middleName(middle), position(pos), experience(exp), salary(sal) {}

};

// Клас контейнера для співробітників

class EmployeeContainer {

private:

// Внутрішній клас для представлення елемента списку

class Node {

public:

Employee employee;

Node\* next;

Node(const Employee& emp) : employee(emp), next(nullptr) {}

};

Node\* head; // Початок списку

public:

// Конструктор

EmployeeContainer() : head(nullptr) {}

// Метод для додавання співробітника до контейнера

void addEmployee(const Employee& emp) {

Node\* newNode = new Node(emp);

newNode->next = head;

head = newNode;

}

// Метод для видалення співробітника за прізвищем

void removeEmployee(const std::string& lastName) {

Node\* current = head;

Node\* prev = nullptr;

while (current != nullptr && current->employee.lastName != lastName) {

prev = current;

current = current->next;

}

if (current != nullptr) {

if (prev != nullptr) {

prev->next = current->next;

} else {

head = current->next;

}

delete current;

}

}

// Метод для виводу інформації про всіх співробітників

void printEmployees() const {

Node\* current = head;

while (current != nullptr) {

std::cout << "Last Name: " << current->employee.lastName << ", First Name: " << current->employee.firstName

<< ", Position: " << current->employee.position << ", Experience: " << current->employee.experience

<< ", Salary: " << current->employee.salary << std::endl;

current = current->next;

}

}

// Деструктор для видалення всіх вузлів

~EmployeeContainer() {

Node\* current = head;

while (current != nullptr) {

Node\* next = current->next;

delete current;

current = next;

}

}

};

int main() {

// Приклад використання контейнера

EmployeeContainer container;

container.addEmployee(Employee("Doe", "John", "M", "Manager", 5, 50000.0));

container.addEmployee(Employee("Smith", "Alice", "L", "Developer", 3, 60000.0));

container.addEmployee(Employee("Johnson", "Bob", "K", "Designer", 8, 70000.0));

std::cout << "Employees before removal:" << std::endl;

container.printEmployees();

container.removeEmployee("Smith");

std::cout << "\nEmployees after removal:" << std::endl;

container.printEmployees();

return 0;

}