ФАКУЛЬТЕТ «Информатика и системы управления»  
КАФЕДРА «Информационная безопасность»

ОТЧЕТ  
по лабораторной работе № 4  
по учебной дисциплине «Алгоритмические языки»  
на тему: «Использование своих классов в STL»

Вариант 9

Выполнил:   
Студент 1 курса, гр. ИУ8-24  
Катков Евгений

**Цель работы:**

Цель работы по данному коду состоит в исследовании и практическом применении возможностей стандартной библиотеки шаблонов (STL) в языке программирования C++, сосредотачиваясь на использовании собственных классов в контейнерах STL.

**Задачи работы:**

1.Изучение основных компонентов STL, таких как векторы, деки, алгоритмы сортировки и другие контейнеры и алгоритмы.

2.Создание пользовательского класса (например, Employee), который будет представлять собой сущность, содержащую информацию о сотрудниках.

3.Разработка программы на языке C++, использующей STL для управления данными типа Employee.

4.Реализация операций вставки, сортировки и вывода данных о сотрудниках с использованием контейнеров и алгоритмов STL.

5.Анализ преимуществ и недостатков использования собственных классов в контексте STL.

6.Формирование отчета, содержащего описание работы, ее цели, описание методов и результатов исследования, а также выводы и рекомендации.

**Выполнение работы:**

*#include* <iostream> *// Подключение заголовочного файла для стандартного ввода-вывода  
#include* <fstream> *// Подключение заголовочного файла для работы с файлами  
#include* <utility> *// Подключение заголовочного файла для использования утилит, таких как move()  
#include* <vector> *// Подключение заголовочного файла для использования векторов  
#include* <deque> *// Подключение заголовочного файла для использования деков  
#include* <string> *// Подключение заголовочного файла для работы со строками  
#include* <algorithm> *// Подключение заголовочного файла для использования алгоритмов, таких как sort()  
  
using namespace* std; *// Использование пространства имен std  
  
class* Employee { *// Определение класса Employee* string name; *// Имя сотрудника* string date; *// Дата приема на работу* string position; *// Должность сотрудника  
 double* salary; *// Зарплата сотрудника  
public*:  
 *// Конструктор по умолчанию* Employee() : name(""), date(""), position(""), salary(0.0) {}  
  
 *// Пользовательский конструктор* Employee(string name, string date, string position, *double* salary) :  
 name(move(name)), date(move(date)), position(move(position)), salary(salary) {}  
  
 *// Перегрузка оператора вывода  
 friend* ostream& *operator*<<(ostream& os, *const* Employee& employee) {  
 os << "Name: " << employee.name << ", Hire date: " << employee.date  
 << ", Position: " << employee.position << ", Salary: " << employee.salary;  
 *return* os;  
 }  
  
 *// Перегрузка оператора сравнения меньше  
 bool operator*<(*const* Employee& other) *const* {  
 *return* salary > other.salary; *// Сравнение по убыванию зарплаты* }  
};  
  
*int* main() { *// Главная функция программы* vector<Employee> employees; *// Вектор для хранения объектов класса Employee  
  
 // Чтение данных из файла "input.txt"* ifstream file\_1("input.txt"); *// Открытие файла для чтения  
 if* (!file\_1.is\_open()) { *// Проверка на успешное открытие файла  
 return false*; *// Возврат false в случае ошибки* }  
  
 string name, date, position; *// Переменные для хранения данных о сотруднике  
 double* salary;  
 *while* (inFile >> name >> date >> position >> salary) { *// Чтение данных из файла* employees.emplace\_back(name, date, position, salary); *// Добавление нового объекта в вектор* }  
 inFile.close(); *// Закрытие файла после чтения данных  
  
 // Сортировка данных о сотрудниках по убыванию зарплаты* sort(employees.begin(), employees.end());  
  
 *// Вывод отсортированных данных о сотрудниках на экран  
 for* (*const auto*& employee : employees) {  
 cout << employee << endl;  
 }  
  
 deque<Employee> employee\_deque; *// Дек для хранения объектов класса Employee* copy(employees.begin(), employees.end(), back\_inserter(employee\_deque)); *// Копирование элементов из вектора в дек  
  
 // Запись данных о сотрудниках в файл "output.txt"* ofstream file\_2("output.txt"); *// Открытие файла для записи  
 if* (!file\_2.is\_open()) { *// Проверка на успешное открытие файла  
 return false*; *// Возврат false в случае ошибки* }  
  
 *// Запись данных о сотрудниках в файл  
 for* (*const auto*& employee : employees) {  
 file\_2 << employee << endl; *// Запись данных о сотруднике в файл* }  
 *// Запись данных о сотрудниках из дека в файл  
 for* (*const auto*& employee : employee\_deque) {  
 file\_2 << employee << endl; *// Запись данных о сотруднике в файл* }  
 file\_2.close(); *// Закрытие файла после записи данных  
  
 return* 0; *// Возврат нулевого значения, указывающего на успешное завершение программы*}

**Вывод:**

Лабораторная работа позволяет закрепить навыки работы с контейнерами STL, алгоритмами сортировки, операциями ввода-вывода и работой с пользовательскими классами в C++. Кроме того, она позволяет изучить принципы организации данных в контейнерах и применение различных контейнеров для разных целей.