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

ОТЧЕТ  
по лабораторной работе № 4  
по учебной дисциплине «Алгоритмические языки»  
на тему: «Использование объектов своих классов в последовательных контейнерах библиотеки STL в языке Си++»

Вариант 23

Выполнил:   
Студент 1 курса, гр. ИУ8-24  
Терехин Данила

**Цель работы:**

Практическое применение контейнеров стандартной библиотеки шаблонов (STL) для хранения объектов пользовательского класса, а также использование методов и алгоритмов STL для работы с этими контейнерами.

**Задачи работы:**

* Изучить необходимые учебные материалы, посвященные последовательным контейнерам библиотеки шаблонов в языке Си++
* Разработать программу на языке Си++ для решения заданного варианта задания
* Отладить программу
* Представить результаты работы программы
* Подготовить отчет по лабораторной работе

**Выполнение работы:**

**Main.cpp**

#include <iostream>

#include <string>

#include <vector>

#include <deque>

#include <algorithm>

#include <fstream>

class BankCredit {

public:

    // Конструктор для инициализации объекта BankCredit с заданными параметрами

    BankCredit(const std::string& name, double amount, const std::string& currency, double rate)

        : name(name), amount(amount), currency(currency), rate(rate) {

        // Инициализация полей класса с переданными значениями

    }

    // Конструктор копирования для создания копии объекта BankCredit

    BankCredit(const BankCredit& other)

        : name(other.name), amount(other.amount), currency(other.currency), rate(other.rate) {

        // Копирование значений из другого объекта BankCredit

    }

    // Оператор присваивания копированием для копирования значений из другого объекта BankCredit

    BankCredit& operator=(const BankCredit& other) {

        if (this != &other) { // Проверка на самоприсваивание

            name = other.name; // Копирование названия кредита

            amount = other.amount; // Копирование суммы кредита

            currency = other.currency; // Копирование типа валюты

            rate = other.rate; // Копирование ставки в процентах годовых

        }

        return \*this; // Возвращение текущего объекта

    }

    // Перегруженная операция вставки в поток для удобного вывода объекта

    friend std::ostream& operator<<(std::ostream& os, const BankCredit& credit) {

        os << credit.name << " " << credit.amount << " " << credit.currency << " " << credit.rate;

        // Вывод названия, суммы, типа валюты и ставки кредита в поток

        return os; // Возвращение потока

    }

    // Перегруженный оператор < для сравнения объектов по ставке кредита

    bool operator<(const BankCredit& other) const {

        return rate < other.rate; // Сравнение ставок кредитов

    }

private:

    std::string name; // Название кредита

    double amount; // Сумма кредита

    std::string currency; // Тип валюты

    double rate; // Ставка в процентах годовых

};

int main() {

    std::ifstream input("input.txt"); // Открытие файла для чтения

    std::ofstream output("output.txt"); // Открытие файла для записи

    if (!input.is\_open() || !output.is\_open()) {

        std::cerr << "Error opening file!" << std::endl;

        return 1;

    }

    std::vector<BankCredit> creditsVector; // Вектор для хранения объектов BankCredit

    std::deque<BankCredit> creditsDeque; // Дек для хранения объектов BankCredit

    std::string name;

    double amount, rate;

    std::string currency;

    // Чтение данных из файла и заполнение вектора

    while (input >> name >> amount >> currency >> rate) {

        creditsVector.emplace\_back(name, amount, currency, rate);

    }

    // Вывод исходного контейнера в файл

    output << "Original Vector:" << std::endl;

    for (const auto& credit : creditsVector) {

        output << credit << std::endl;

    }

    // Сортировка вектора по ставке кредита

    std::sort(creditsVector.begin(), creditsVector.end());

    // Вывод отсортированного контейнера в файл

    output << "Sorted Vector:" << std::endl;

    for (const auto& credit : creditsVector) {

        output << credit << std::endl;

    }

    // Копирование отсортированного вектора в дек

    std::copy(creditsVector.begin(), creditsVector.end(), std::back\_inserter(creditsDeque));

    // Вывод скопированного контейнера в файл

    output << "Copied Deque:" << std::endl;

    for (const auto& credit : creditsDeque) {

        output << credit << std::endl;

    }

    input.close(); // Закрытие файла для чтения

    output.close(); // Закрытие файла для записи

    return 0;

}

Файл со входными данными– **input.txt:**Credit1 1000 USD 2.0

Credit2 4200 EUR 3.5

Credit4 3000 JPY 2.1

Credit3 1500 GBP 4.2

Файл с выходными данными **– output.txt:**

Original Vector:

Credit1 1000 USD 2

Credit2 4200 EUR 3.5

Credit4 3000 JPY 2.1

Credit3 1500 GBP 4.2

Sorted Vector:

Credit1 1000 USD 2

Credit4 3000 JPY 2.1

Credit2 4200 EUR 3.5

Credit3 1500 GBP 4.2

Copied Deque:

Credit1 1000 USD 2

Credit4 3000 JPY 2.1

Credit2 4200 EUR 3.5

Credit3 1500 GBP 4.2

**Вывод:**

В ходе выполнения лабораторной работы было успешно реализовано использование объектов собственных классов в последовательных контейнерах библиотеки STL с применением различных методов и алгоритмов для работы с ними. Это позволило познакомиться с применением стандартной библиотеки шаблонов в практических задачах программирования.