Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н. Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н. Э. Баумана)

Факультет (ИУ) «Информатика, искусственный интеллект и системы управления»

Кафедра Информационная безопасность (ИУ8)

Лабораторная работа № 3

ПО КУРСУ

«Алгоритмические языки»

на тему «Изучение перегрузки стандартных операций в языке C++

Студент ИУ8-24 (Группа) А. А. Урнышева (И. О. Фамилия)

Преподаватель: Д. В. Барыкин (И.О. Фамилия)

Цель работы: Работа с классами, сортировка объектов класса, копирование из одного контейнера в другой

Условие задачи: В приложении организовать контейнер объектов своего класса (использовать шаблоны std::list, std::vector или std::deque в зависимости от варианта, элементы контейнера - объекты класса, **не указатели!!!!**). Варианты заданий заданы в ячейках таблицы 1. Параметры приложений указаны в заголовках строк и столбцов таблицы 1.

Класс должен иметь необходимые конструкторы, конструктор копирования и перемещения при необходимости (обосновать отсутствие или наличие необходимости), перегруженные операции присваивания с копированием и перемещением при необходимости (обосновать отсутствие или наличие необходимости), перегруженную операцию вставки в поток <<.

Обеспечить копирование одного контейнера в другой с помощью алгоритма std::copy. А также сортировку объектов в исходном контейнере, для шаблона list при сортировке использовать метод list::sort без параметров, для шаблона vector или deque при сортировке использовать алгоритм std::sort с двумя параметрами: итератор на начало и итератор на конец контейнера.

Исходные данные прочитать из текстового файла input.txt. Вывести в выходной файл output.txt исходный контейнер, контейнер после сортировки, использовать при этом перегруженную операцию вставки в поток, также вывести в выходной файл контейнер, в который скопирован исходный контейнер.

Вариант 24: Объект - банковский кредит (поля: название, сумма кредита, тип валюты, ставка в % годовых). Исходный контейнер list, копируем в vector.

Текст программы с комментариями:

Файл Header1.h:

```
#pragma once
#include <iostream>
#include <string>

using namespace std;

class Kredit {
	string name; //название кредита
	int sum; //сумма кредита
	string type_valuta; //тип валюты
```

```
	int stavka; //ставка в %

public:
	Kredit(string name, int sum, string type_valuta, int stavka) { //конструктор
		this->name = name;
		this->sum = sum;
		this->type_valuta = type_valuta;
		this->stavka = stavka;
	}

	Kredit(const Kredit& kredit) { //конструктор копирования
		this->name = kredit.name;
		this->sum = kredit.sum;
		this->type_valuta = kredit.type_valuta;
		this->stavka = kredit.stavka;
	}

	Kredit() { //конструктор с нулями
		name = "";
		sum = 0;
		type_valuta = "";
		stavka = 0;
	}

	//дружественные функции для перегрузки операторов ввода-вывода
	friend ostream& operator<<(ostream& os, const Kredit& k1);
	friend istream& operator >>(istream& is, Kredit&);

	//дружественные функции для перегрузки операторов сравнения
	friend bool operator>(const Kredit& k1, const Kredit& k2);
	friend bool operator<(const Kredit& k1, const Kredit& k2);
};

ostream& operator<<(ostream& os, const Kredit& k1) { //перегрузка оператора вывода
	os << "Name of kredit: " << k1.name << endl;
	os << "Sum of kredit: " << k1.sum << endl;
	os << "Type of valuta: " << k1.type_valuta << endl;
	os << "Stavka: " << k1.stavka << "%" << endl;
	return os;
}

istream& operator>>(istream& is, Kredit& k1) { //перегрузка оператора ввода
	is >> k1.name;
	is >> k1.sum;
	is >> k1.type_valuta;
	is >> k1.stavka;
	return is;
}

bool operator >(const Kredit& k1, const Kredit& k2) { //перегрузка оператора сравнения в одну сторону
	return k1.stavka > k2.stavka;
}

bool operator <(const Kredit& k1, const Kredit& k2) { //перегрузка оператора сравнения в другую сторону
	return k1.stavka < k2.stavka;
	}
```

Файл main.cpp:

```
#include "Header1.h"

#include <vector>
#include <list>
#include <algorithm>
```

```
#include <fstream>

void SaveToFile(const string& filename, const Kredit& k1) { //функция сохранения данных типа Kredit в файл
	ofstream file(filename, ios::app);
	if (file.is_open()) {
		file << k1 << endl;
		cout << k1 << endl;
		file.close();
	}
	else cerr << "Can't open this file :(" << endl;
}

void SaveToFile(const string& filename, const string& s) { //перегрузка функции для сохранения в файл данных типа string
	ofstream file(filename, ios::app);
	if (file.is_open()) {
		file << s << endl;
		cout << s << endl;
		file.close();
	}
	else cerr << "Can't open this file :(" << endl;
}

int main() {
	ofstream clear("output.txt");
	clear.close();
	ifstream input("input.txt");

	int n;
	list<Kredit>K1; //создание двусвязного списка
	cout << "Enter amount of kredits: ";
	cin >> n;
	cout << endl;

	for (size_t i = 0; i < n; ++i) {
		Kredit k;
		input >> k;
		K1.push_back(k);
	}

	string s = "Before sort: ";
	SaveToFile("output.txt", s);

	for (const auto e : K1) SaveToFile("output.txt", e);

	s = "After sort";

	K1.sort(); //сортировка двусвязного списка

	SaveToFile("output.txt", s);
	for (const auto e : K1) SaveToFile("output.txt", e);

	s = "Vector from list: ";
	SaveToFile("output.txt", s);

	vector<Kredit>K2;

	copy(K1.begin(), K1.end(), back_inserter(K2));

	for (const auto e : K2) SaveToFile("output.txt", e);

	return 0;
	}
```

Файл input.txt:

name1

1500

type1

10

name2

3000

type2

20

name3

4000

type3

5

Файл output.txt:

Before sort:

Name of kredit: name1

Sum of kredit: 1000

Type of valuta: type1

Stavka: 10%

Name of kredit: name2

Sum of kredit: 2000

Type of valuta: type2

Stavka: 20%

Name of kredit: name3

Sum of kredit: 3000

Type of valuta: type3

Stavka: 5%

After sort

Name of kredit: name3

Sum of kredit: 3000

Type of valuta: type3

Stavka: 5%

Name of kredit: name1

Sum of kredit: 1000

Type of valuta: type1

Stavka: 10%

Name of kredit: name2

Sum of kredit: 2000

Type of valuta: type2

Stavka: 20%

Vector from list:

Name of kredit: name3

Sum of kredit: 3000

Type of valuta: type3

Stavka: 5%

Name of kredit: name1

Sum of kredit: 1000

Type of valuta: type1

Stavka: 10%

Name of kredit: name2

Sum of kredit: 2000

Type of valuta: type2

Stavka: 20%

Вывод: В данной лабораторной работе мы прочитали данные типа класса Kredit из текстового файла input.txt, поместили их в контейнер list, отсортировали, скопировали в контейнер vector и поместили в файл output.txt. программа работает.