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

ОТЧЕТ  
по лабораторной работе № 2  
по учебной дисциплине «Алгоритмические языки»  
на тему: «Изучение перегрузки стандартных операций в языке Си++»

Вариант 23

Выполнил:   
Студент 1 курса, гр. ИУ8-24  
Спиридонов Олег

**Цель работы:**

Овладение навыками разработки программ на языке C++, использующих перегрузку стандартных операций.

**Задачи работы:**

* Изучить необходимые учебные материалы, посвященные перегрузке стандартных операций в языке Си++
* Разработать программу на языке Си++ для решения заданного варианта задания
* Отладить программу
* Выполнить решение контрольного примера с помощью программы и ручной расчет контрольного примера
* Подготовить отчет по лабораторной работе

**Условие задачи:**

1. **Описание класса Vector:**
   * Класс Vector задает вектор размерности n.
   * Поля класса:
     1. Число элементов (размерность) вектора.
     2. Указатель на массив типа int, задающий вектор.
2. **Конструкторы класса Vector:**
   * Конструктор без параметров, задающий пустой вектор (число элементов равно 0).
   * Конструктор, создающий объект вектора на основе обычного одномерного массива размерности n.
   * Конструктор копирования.
   * Конструктор перемещения.
   * Деструктор.
3. **Необходимые операции для перегрузки:**
   * Операция []: обращение к элементу вектора по индексу.
   * Операция = (присваивание с копированием).
   * Операция = (присваивание с перемещением).
   * Операция < (сравнение вектора и массива).
   * Операция вставки (<<) объекта в поток (объект класса ostream).
   * Операция извлечения (>>) объекта из потока (объект класса istream).
4. **Демонстрация разницы между конструктором копирования и конструктором перемещения.**
5. **Исходные коды разместить в файлах**
   * Заголовочный файл класса
   * Файл реализации класса
6. **Обработка входных и выходных данных:**
   * Все входные данные читаются из текстового файла input.txt.
   * Результаты выводятся в файл output.txt.

**Выполнение работы:**

Файл **input.txt:**

5

1 2 3 4 5

Заголовочный файл класса Vector – **Vector.h**:

#ifndef VECTOR\_H

#define VECTOR\_H

#include <iostream>

#include <fstream>

#include <vector>

using namespace std;

class Vector {

friend ostream& operator<<(ostream& os, Vector& Vector);

friend istream& operator>>(istream& is, Vector& Vector);

friend bool operator<(Vector Vector, int\* massiv);

public:

Vector();

Vector(double\* massiv, int size);

Vector(Vector& Vector2);

Vector(Vector&& Vector2);

~Vector() {

if (p != nullptr) {

delete[] p; // Зачем []?

}

}

int operator[](int index) {

return p[index];

}

void operator=(const Vector& Vector2) {

n = Vector2.n;

if (p == nullptr) {

p = new int[n];

}

for (int i = 0; i < n; ++i) {

p[i] = Vector2.p[i]; // Почему нельзя Vector2[i]?

}

}

void operator=(Vector&& Vector2) {

n = Vector2.n;

if (p == nullptr) {

p = new int[n];

}

for (int i = 0; i < n; ++i) {

p[i] = Vector2.p[i];

}

Vector2.n = 0;

delete[] Vector2.p;

Vector2.p = nullptr;

}

private:

int n = 0;

int\* p = nullptr;

};

Vector::Vector() {

};

Vector::Vector(double\* massiv, int size) {

this->n = size;

this->p = new int[n];

for (int i = 0; i < n; ++i) {

this->p[i] = massiv[i];

}

};

Vector::Vector(Vector& Vector2) {

this->n = Vector2.n;

this->p = new int[n];

for (int i = 0; i < n; ++i) {

this->p[i] = Vector2[i];

}

};

Vector::Vector(Vector&& Vector2) { // noexcept?

this->n = Vector2.n;

this->p = new int[n];

for (int i = 0; i < n; ++i) {

this->p[i] = Vector2[i];

}

Vector2.n = 0;

delete[] Vector2.p;

Vector2.p = nullptr;

};

ostream& operator<<(ostream& os, Vector& Vector) {

ofstream file("output.txt");

if (file.is\_open()) {

file << Vector.n << endl;

for (int i = 0; i < Vector.n; ++i) {

file << Vector.p[i] << ' ';

}

}

else {

cout << "Output file didn't open." << endl;

}

os << Vector.n << endl;

for (int i = 0; i < Vector.n; ++i) {

os << Vector.p[i] << ' ';

}

return os;

}

istream& operator>>(istream& is, Vector& Vector) {

int amount;

int value;

vector <int> placeholder;

is >> amount;

for (int i = 0; i < amount; ++i) {

is >> value;

placeholder.push\_back(value);

}

Vector.n = placeholder.size();

Vector.p = new int[Vector.n];

for (int i = 0; i < Vector.n; ++i) {

Vector.p[i] = placeholder[i];

}

return is;

}

bool operator<(Vector Vector, int\* massiv) {

for (int i = 0; i < Vector.n; ++i) {

if (Vector[i] < massiv[i]) {

return true;

}

else if (Vector[i] > massiv[i]) {

return false;

}

}

return false;

}

#endif

Файл реализации – **AL Laba 2 (operation overload).cpp:**

#include "Vector.h"

#include <iostream>

#include <fstream>

using namespace std;

int main() {

ofstream output\_file("output.txt");

output\_file.close();

ifstream input\_file("input.txt");

if (!input\_file.is\_open()) {

cout << "Input file didn't open." << endl;

}

Vector vector1;

input\_file >> vector1;

cout << "vector1 after creation based on data from input.txt:" << endl << vector1 << endl;

Vector vector2(vector1);

cout << "vector1 and vector2 after creating vector2 via copying data from vector1:" << endl << vector1 << endl << vector2 << endl;

Vector vector3(move(vector2));

cout << "vector2 and vector3 after creating vector3 via moving data from vector2:" << endl << vector2 << endl << vector3 << endl;

vector2 = vector3;

cout << "vector2 and vector3 after assigning (with copying) data of vector3 to vector2:" << endl << vector2 << endl << vector3 << endl;

vector2 = move(vector3);

cout << "vector2 and vector3 after assigning (with moving) data of vector3 to vector2:" << endl << vector2 << endl << vector3 << endl;

int array1[5] = { 1, 5, 4, 3, 2 };

cout << "This array will be compared with vector1:" << endl;

for (int i = 0; i < 5; ++i) {

cout << array1[i] << ' ';

}

cout << endl;

cout << "Result of comparing vector1 with array1:" << endl << (vector1 < array1) << endl;

}

**Пример и контрольный расчет примера:**

Исходные данные:

Вектор длины 5 с элементами 1 2 3 4 5

Массив длины 5 с элементами 1 5 4 3 2

Ожидаемый результат:

1

Полученный результат:

1

**Вывод:**

В ходе выполнения лабораторной работы были изучены основы перегрузки стандартных операций в языке C++. Были реализованы операции работы с векторами, такие как обращение к элементу по индексу, копирование и перемещение векторов, сравнение вектора и массива. В процессе работы были изучены способы перегрузки операций в классах C++, в том числе перегрузка как членами класса, так и дружественными функциями.