Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный технический университет имени Н. Э. Баумана (национальный исследовательский университет)

**Лабораторная работа № 2  
Изучение перегрузки стандартных операций в языке Си++**

Работу выполнил:

Катков Евгений, ИУ8-24

**ЦЕЛЬ**

Цель работы состоит в овладении навыками разработки программ на языке Си++, использующих перегрузку стандартных операций.

**УСЛОВИЕ ЗАДАЧИ**

Дан класс (например, с именем *Vector*), задающий вектор размерности *n*. Поля класса: указатель на массив, задающий вектор (тип элемента *int* или *double* в зависимости от варианта), массив должен создаваться динамически, число элементов (размерность) вектора (тип *int*). Класс включает: конструктор без параметров, задающий пустой вектор (число элементов равно 0), конструктор, создающий объект вектор на основе обычного одномерного массива размерности *n*, конструктор копирования, конструктор перемещения, деструктор.

Необходимо перегрузить операции и продемонстрировать их работу. Перегрузить операцию [] (обращение к элементу вектора по индексу), операцию = (присваивание с копированием), операцию = (присваивание с перемещением), а также операцию вставки (<<) объекта в поток cout или в файл (объект класса ostream) и операцию извлечения (>>) объекта из потока cin или из файла (объект класса istream).

**ТЕКСТ ПРОГРАММЫ**

//

// main.cpp

// Sem1

//

// Created by Михаил on 28.02.2024.

//

#include <iostream>

#include <fstream>

#include <vector>

using namespace std;

class Vector

{

double\* Chleni = nullptr;

int size = 0;

public:

Vector(double\* Mass, int n) //конструктор создания вектора значений с заданным размеровм

{

size = n;

Chleni = new double [size];

for (int i = 0; i < size; ++i)

{

Chleni [i] = Mass [i];

}

}

Vector(int n) //конструктор созданя пустого вектора с заданной длинной

{

size = n;

Chleni = new double[size];

}

Vector(Vector& V1) //конструктор копированя

{

size = V1.size;

Chleni = new double[size];

for (int i = 0; i < size; ++i)

{

Chleni [i] = V1.Chleni [i];

}

}

Vector(Vector&& V2) //конструктор перемещения

{

swap (Chleni, V2.Chleni); //меняются указатели на массив

swap (size, V2.size); //меняются размерности

}

double& operator [] (int i) //перегрузка операци доступа к элементу по индексу

{

return Chleni [i];

}

Vector& operator = (Vector& V3) //перегрузка оператора прсваивания с копированиием

{

if (this != &V3)

{

size = V3.size;

if (Chleni == nullptr)

{

delete[] Chleni;

}

Chleni = new double[size];

for (int i = 0; i < size; ++i)

{

Chleni[i] = V3.Chleni[i];

}

}

return \*this;

}

Vector& operator = (Vector&& V3) //перегрузка оператора прсваивания с перемещением

{

if (this != &V3)

{

swap (Chleni, V3.Chleni); //меняются указатели на массив

swap (size, V3.size); //меняются размерности

}

return \*this;

}

friend ostream& operator << (ostream& out, Vector& V) //перегрузка ввода

{

for (int i = 0; i < V.size; ++i)

{

out << V.Chleni [i] << " ";

}

return out;

}

friend istream& operator >> (istream& in, Vector& V) //перегрузка вывода

{

for (int i = 0; i < V.size; ++i)

{

in >> V.Chleni [i];

}

return in;

}

friend Vector operator + (Vector& V1, Vector& V2) //перегрузка операции сложения

{

Vector V3 (V1.size + V2.size); //прсваивание длине нового вектора сумму длинн двух заданных

for (int i = 0; i < V1.size; ++i)

{

V3 [i] = V1.Chleni [i]; //первые элементы нового вектора - все элементы первого вектора

}

for (int i = 0; i < V2.size; ++i)

{

V3 [i + V1.size] = V2.Chleni [i]; //все последюзие элементы - все элемеенты второго вектора

}

return V3;

}

~Vector() //диструктор класса

{

delete [] Chleni;

}

};

int main()

{

ifstream inputFile ("in.txt");

if (!inputFile.is\_open())

{

cout << "Файл не открыт" << endl;

return 1;

}

int a;

int b;

inputFile >> a >> b;

Vector V1 (a);

Vector V2 (b);

inputFile >> V1 >> V2;

inputFile.close();

Vector V3 = V1 + V2;

ofstream outputFile ("out.txt");

if (!outputFile.is\_open())

{

cout << "файл не открыт 1111" <<"\n";

return 1;

}

outputFile << "Vector 1: " << V1 <<"\n";

outputFile << "Vector 2: " << V2 <<"\n";

outputFile << "Sum: " << V3 <<"\n";

outputFile.close();

return 0;

}

**ВЫВОД**

В данной работе мы создали класс Vector, который представляет собой вектор и обладает следующими функциональностями:

* Возможность создания вектора с помощью конструктора, который принимает массив значений и их количество, а также конструктора, который принимает только количество элементов вектора.
* Реализацию конструктора копирования и конструктора перемещения, что позволяет корректно копировать и перемещать объекты класса.
* Оператор доступа к элементам вектора [], позволяющий получить доступ к элементам вектора по индексу.
* Оператор присваивания для копирования и перемещения, обеспечивающий корректное копирование и перемещение объектов класса.
* Оператор сложения векторов который позваоляет на выходе получить вектор длинны равной сумме длинн двух других и состоящий из последовательно расположенных элементов первого а потов второго вектора
* Деструктор, который освобождает память, выделенную для массива значений вектора.
* Перегрузку операторов << и >>, которые позволяют выводить и считывать объекты класса Vector с помощью потоков cout и cin соответственно.

Для демонстрации работы класса был написан простой тестовый код, в котором читаются два вектора и их размерности из файлов, которые в дальнейшем суммируются с помощью перегруженной операции сложения, на выходе получается третий вектор

Этот класс может быть полезен при работе с векторами чисел в программировании на C++.