Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«Пермский национальный исследовательский политехнический университет»**

Кафедра **«**Информационные технологии и автоматизированные системы**»**

**ОТЧЕТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ №6**

Дисциплина: «Основы алгоритмизации и программирования»

Семестр 2

Тема: АТД. Контейнеры

Вариант 6.

Выполнил работу

Студент группы РИС-22-1Б

Поважный В. Е.

Проверил

Доцент кафедры ИТАС

Полякова О.А.

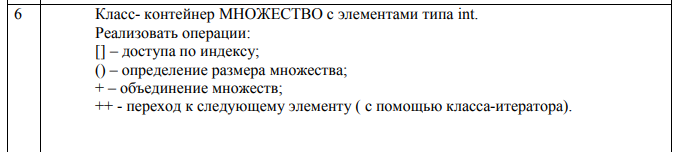
г. Пермь-2023

**Введение**

Для выполнения лабораторной работы требуется сделать отчет программы. Создание программ – отличный способ практики программирования.

**Постановка задачи**

1. Определить пользовательский класс.
2. Реализовать конструкторы, деструктор, операции ввода-вывода, операцию присваивания.
3. Перегрузить операции, указанные в варианте.
4. Реализовать класс-итератор. Реализовать с его помощью операции последовательного доступа.
5. Написать тестирующую программу, иллюстрирующую выполнение операций.



**Код программы**

**ооп 6.cpp**

#include <iostream>

#include "Header.h"

using namespace std;

int main()

{

many arc;

many marc(2);

arc++;

arc[0] = 1;

arc[1] = 23;

marc[0] = 4;

marc[1] = 5;

many marcs = arc + marc;

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

cout << marcs[i];

}

**Header.h**

**class** many

{

int\* array;

int size;

public:

many();

many(int size);

int& operator [] (int i);

int operator () ();

void operator ++ (int i);

many operator + (many& other);

~many();

};

**Source.cpp**

**#include** <iostream>

#include "Header.h"

using namespace std;

many::many()

{

array = new int[1];

array[0] = 0;

size = 1;

}

many::many(int size)

{

array = new int [size] {};

this->size = size;

}

int& many::operator [] (int i)

{

return array[i];

}

int many::operator () ()

{

return size;

}

void many::operator ++ (int i)

{

size++;

int\* new\_array = new int[size];

for (int a = 0; a < size - 1; a++)

new\_array[a] = array[a];

new\_array[size - 1] = 0;

delete[] array;

array = new\_array;

}

many many::operator + (many& other)

{

many newmany(this->size + other.size);

for (int i = 0; i < this->size; i++)

newmany.array[i] = this->array[i];

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

newmany.array[size + i] = other.array[i];

return newmany;

}

many::~many()

{

if (!array)

delete[] array;

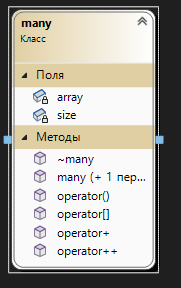
}

**Вывод программы**

****

***Рисунок 1 – вывод программы***

**UML диаграмма**

****

***Рисунок 2 – UML диаграмма***

**Вывод**

Программа выполняет свою задачу.

**Заключение**

Для решения задачи потребовались знания языка программирования, а конкретнее C++. Программа выполняет те условия, что были указаны в постановке задачи и работает без проблем.