**Пермский национальный исследовательский политехнический университет**

Кафедра “Информационные технологии и автоматизированные системы”

**Лабораторная работа №23**

«По дисциплине «Основы алгоритмизации и программирования»

**Тема:**

АТД. Контейнеры

Выполнила:

Студентка группы ИВТ-19-1б

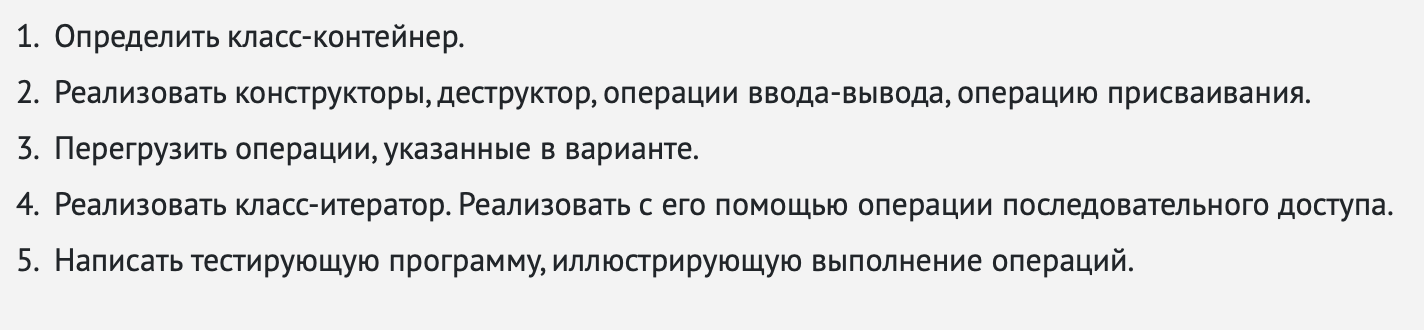
Еремеева Полина Алексеевна

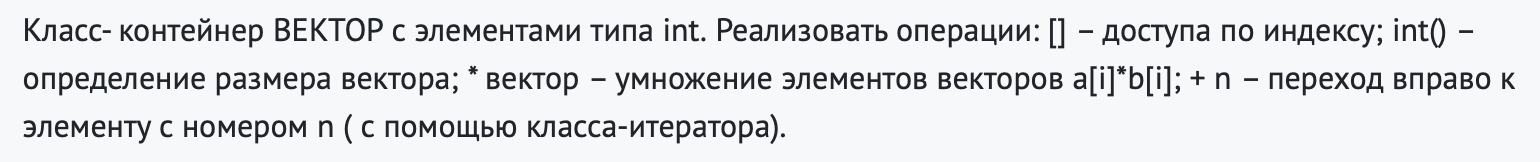
Проверила:

доцент кафедры “ИТАС”

Полякова О.А.

Пермь, 2020

**Постановка задачи**



**Код**

Vector.hpp

#include <iostream>

**using** **namespace** std;

**class** Iterator {

**friend** **class** Vector;

**public**:

Iterator() { elem = 0; };

Iterator(**const** Iterator& it) { elem = it.elem; }

**bool** **operator**==(**const** Iterator& it) { **return** elem == it.elem; }

**bool** **operator**!=(**const** Iterator& it) { **return** elem != it.elem; }

**void** **operator**++() { ++elem; }

**void** **operator**++(**int**) { ++elem; }

**void** **operator**--() { --elem; }

**void** **operator**--(**int**) { --elem; }

**void** **operator**+(**const** **int** k);

**int**& **operator**\*() **const** { **return** \*elem; }

**private**:

**int**\* elem;

};

**class** Vector {

**public**:

Vector(**int** s, **int** k = 0);

Vector(**const** Vector& v);

~Vector();

Vector& **operator**=(**const** Vector& v);

**int**& **operator**[](**int** index);

Vector **operator**\*(**const** Vector&v);

**int** **operator**()();

**friend** ostream& **operator**<<(ostream& out, **const** Vector& v);

**friend** istream& **operator**>>(istream& in, Vector& v);

Iterator first() { **return** beg; }

Iterator last() { **return** end; }

**private**:

**int** size;

**int**\* data;

Iterator beg;

Iterator end;

};

Vector.cpp

#include "Vector.hpp"

**using** **namespace** std;

Vector::Vector(**int** s, **int** k) {

size = s;

data = **new** **int**[size];

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

data[i] = k;

}

beg.elem = &data[0];

end.elem = &data[size];

}

Vector::Vector(**const** Vector& v) {

size = v.size;

data = **new** **int**[size];

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

data[i] = v.data[i];

}

beg = v.beg;

end = v.end;

}

Vector::~Vector()

{

**delete**[] data;

data = 0;

}

Vector& Vector::**operator**=(**const** Vector& v) {

**if** (**this** == &v)**return** \***this**;

size = v.size;

**if** (data != 0) **delete**[] data;

data = **new** **int**[size];

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

data[i] = v.data[i];

}

beg = v.beg;

end = v.end;

**return** \***this**;

}

**int**& Vector::**operator**[] (**int** index) {

**if** (index > size) {

cout << "\nError! index > size";

exit(1);

}

**return** data[index];

}

Vector Vector::**operator**\*(**const** Vector& v) {

Vector temp(size);

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

temp.data[i] = v.data[i]\* data[i];

}

**return** temp;

}

**int** Vector::**operator**()() {

**return** size;

}

ostream& **operator**<<(ostream& out, **const** Vector& v) {

**for** (**int** i = 0; i < v.size ; ++i) {

out << v.data[i] << " ";

}

**return** out;

}

istream& **operator**>>(istream& in, Vector& v) {

**for** (**int** i = 0; i < v.size; ++i) {

in >> v.data[i];

}

**return** in;

}

**void** Iterator::**operator**+(**const** **int** k) {

elem += k;

}

Main.cpp

#include <iostream>

#include "Vector.hpp"

**using** **namespace** std;

**int** main() {

setlocale(LC\_ALL, "ru");

cout << "Создаем контейнер A с 5 элементами:\n";

Vector a(5);

cout << a << "\n";

cin >> a;

cout << a <<"\n";

cout << "Элемент с индексом 2 равен 100:\n";

a[2] = 100;

cout << a << "\n";

cout << "Создаем контейнер В с 10 элементами:\n";

Vector b(10);

cout << b << "\n";

cout << "Приравниваем В к А:\n";

b = a;

cout << b << "\n";

cout << "Создаем контейнер С с 10 элементами:\n";

Vector c(10);

cout << "Контейнер С равен перемноженым элементам А и В:\n";

c = a \* b;

cout << c << "\n";

cout << "Вывод длины А= " << a() << endl;

cout << "Nачинаем работу с итератором(А)!\n\n";

cout << "Выводим первый элемент:\n";

cout << \*(a.first()) << endl;

Iterator it = a.first();

cout << "Переходим ко второму элемнту:\n";

it++;

cout << \*it << endl;

cout << "Переходим еще через 2 элемнта:\n";

it + 2;

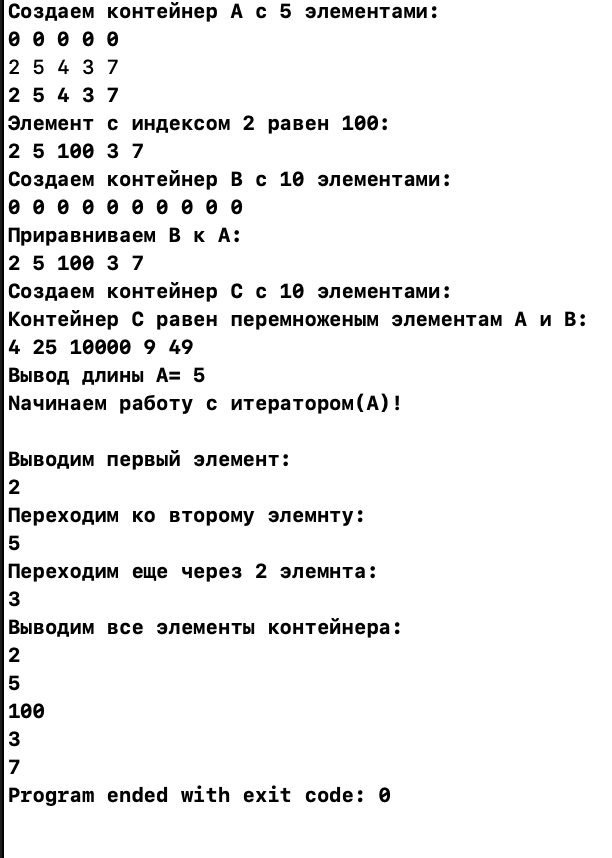
cout << \*it << endl;

cout << "Выводим все элементы контейнера:\n";

**for** (it = a.first(); it != a.last(); it++) cout << \*it << endl;

}

**Тестирование**

****