МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ

И ПРОГРАММНОЙ ИНЖЕНЕРИИ (КАФЕДРА №43)

ОТЧЕТ

ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Старший преподаватель |  |  |  | Е. О. Шумова |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №7

«Стандартная библиотека С++. Последовательные и ассоциативные контейнеры. Обобщенные алгоритмы»

ПО ДИСЦИПЛИНЕ «ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ»

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. | 4831 |  |  |  | К.А. Корнющенков |
| к |  |  | подпись, дата |  | инициалы, фамилия |

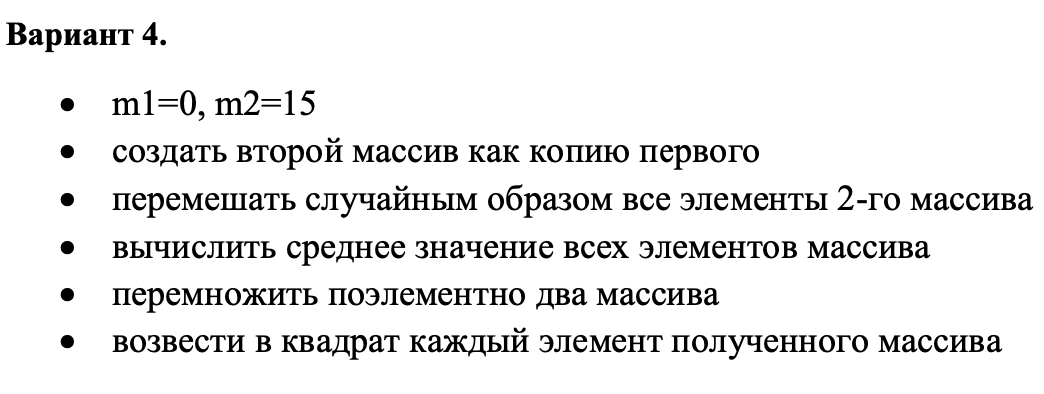
Санкт-Петербург 2020

Задание:

Изучить принципы построения консольных приложений, применив на практике знания базовых синтаксических конструкций языка C++ и объектно-ориентированного программирования. Обязательно использовать обобщенные алгоритмы.

Закрепить знания по теме: Классы, конструкторы и деструкторы, права доступа. Операторные функции в пространстве имен и как члены класса. Стандартная библиотека С++. Библиотека ввода-вывода.

Приложение должно осуществлять ввод и вывод информации о реализованном классе. Заполнить массив данных (vector) случайными числами в диапазоне m1 - m2. Выполнить набор действий при помощи обобщенных алгоритмов, объектов-функций и предикатов. После выполнения каждого действия выводить на экран результат.



Листинг программы:

Main.cpp

#include "WorkWithVector.hpp"

#include <vector>

#include <iostream>

using namespace std;

int main() {

WorkWithVector workWithVector;

vector<int> data;

data = workWithVector.randData(data, 0, 15);

cout << "Массив 1 заполненный случайныеми числами" << endl;

workWithVector.printData(data);

vector<int> data2 = workWithVector.copy(data);

cout << endl << endl << "Массив 2 - копия массива 1" << endl;

workWithVector.printData(data2);

data2 = workWithVector.mix(data2);

cout << endl << endl << "Среднее значение: " << workWithVector.meanValue(data);

vector<int> data3 = workWithVector.vectorMultiplication(data, data2);

cout << endl << endl << "перемножение 2-х массивов" << endl;

workWithVector.printData(data3);

data3 = workWithVector.powVector(data3);

cout << endl << endl << "возвели каждый элемент в квадрат " << endl;

workWithVector.printData(data3);

cout << endl << endl;

}

WorkWithVector.cpp

#include "WorkWithVector.hpp"

#include <iostream>

#include <vector>

#include <cmath>

#include <time.h>

int WorkWithVector::randomElement(int start, int finish){

return rand() % (start+finish) - start;

}

void WorkWithVector::printData(vector<int> data){

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

cout << data[i] << " ";

}

}

vector<int> WorkWithVector::randData(vector<int> data, int min, int max){

srand(time(NULL));

for (int i=0;i<10;i++){

data.push\_back(randomElement(min, max));

}

return data;

}

vector<int> WorkWithVector::copy(vector<int> data){

vector<int> copyData;

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

copyData.push\_back(data[i]);

}

return copyData;

}

vector<int> WorkWithVector::mix(vector<int> data){

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

swap(data[i],data[rand()%data.size()]);

}

return data;

}

float WorkWithVector::meanValue(vector<int> data){

int sum = 0;

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

sum += data[i];

}

return sum/data.size();

}

vector<int> WorkWithVector::vectorMultiplication(vector<int> first,vector<int> second){

vector<int> returnData;

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

returnData.push\_back(first[i]\*second[i]);

}

return returnData;

}

vector<int> WorkWithVector::powVector(vector<int> data){

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

data[i] = pow(data[i], 2);

}

return data;

}

WorkWithVector.hpp

#ifndef WorkWithVector\_hpp

#define WorkWithVector\_hpp

#include <iostream>

#include <vector>

using namespace std;

class WorkWithVector{

private:

int randomElement(int start,int finish);

public:

void printData(vector<int> data);

vector<int> randData(vector<int> data,int min,int max);

vector<int> copy(vector<int> data);

vector<int> mix(vector<int> data);

float meanValue(vector<int> data);

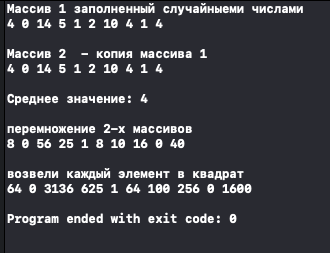
vector<int> vectorMultiplication(vector<int> first,vector<int> second);

vector<int> powVector(vector<int> data);

};

#endif /\* WorkWithVector\_hpp \*/

Скриншоты с результатами:



Вывод: