ГУАП

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

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

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ |
| ««Стандартная библиотека С++. Последовательные и ассоциативные контейнеры. Обобщенные алгоритмы»» |
| по курсу: ОБЪЕКТНО ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4134к |  |  |  | Костяков Н.А. |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2022

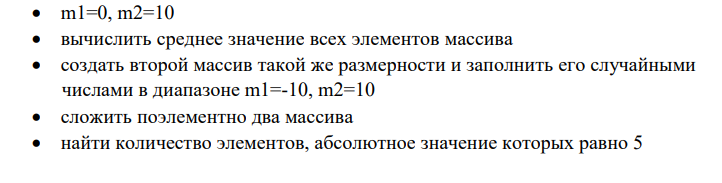
**Цель работы**

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

**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.

**Вариант 3**



**Ход выполнения**

**Листинг**

#include <iostream>

#include <algorithm>

#include <numeric>

#include <vector>

class arr

{

public:

arr(int size, int min, int max);

~arr();

void fill(int index);

void show();

int get\_size() { return size; }

std::vector <int> get\_data() { return data; }

double avg();

arr plus(arr right);

int count\_mod();

private:

std::vector <int> data;

int size;

};

arr::arr(int size, int min, int max)

{

this->size = size;

std::srand(time(0));

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

{

data.push\_back((int)((std::rand() % (abs(min) + abs(max) + 1)) + min)) ;

}

}

arr::~arr()

{

}

void arr::fill(int index)

{

int a;

std::cin >> a;

data[index] = a;

}

void arr::show()

{

for (auto n : data)

std::cout << n << "\t";

std::cout << '\n';

}

double arr::avg()

{

double sum = std::accumulate(data.begin(), data.end(), 0);

//algoritm numeric

return sum/size;

}

arr arr::plus(arr right)

{

std::vector <int> v;

v = right.data;

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

{

v[i] += data[i];

}

right.data = v;

return right;

}

int arr::count\_mod()

{

std::vector <int>v = data;

int count\_div5 = std::count\_if(v.begin(), v.end(), [](int i) {return i % 5 == 0; });

return count\_div5;

}

int main()

{

system("color f1");

int length;

std::cout << "Enter length of array: ";

std::cin >> length;

std::cout << "Array A: \n";

arr A(length, 0, 10);

if (A.get\_data().empty())

{

std::cout << "Length is 0\n";

return 0;

}

A.show();

std::cout << "Average of A is: " << A.avg()<<'\n' << '\n';

std::cout << "Array B: \n";

arr B(length, -10, 10);

B.show();

std::cout << "Average of B is:" << B.avg() << '\n' << '\n';

std::cout << "Sum of A+B = C: \n\n";

A.show();

B.show();

arr C = A.plus(B);

C.show();

std::cout << "Average of C is: \n" << C.avg() << '\n' << '\n';

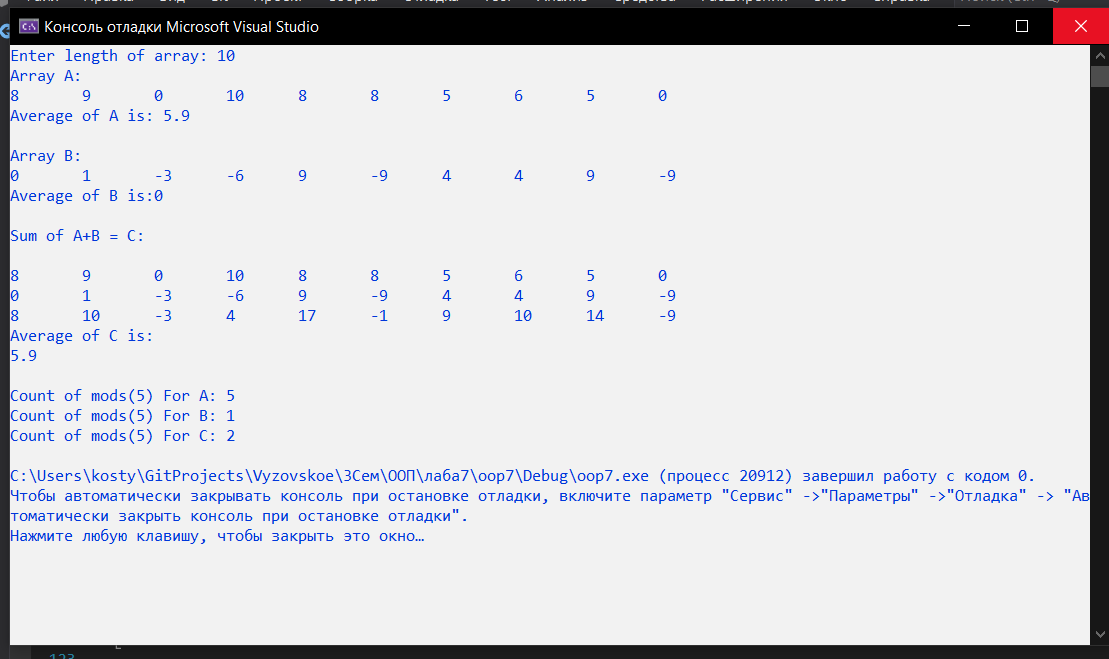
std::cout << "Count of mods(5) For A: " << A.count\_mod()<<'\n';

std::cout << "Count of mods(5) For B: " << B.count\_mod() << '\n';

std::cout << "Count of mods(5) For C: " << C.count\_mod() << '\n';

}

**Результат работы**



**Вывод**

Я написал несколько алгоритмов, которые удобно применять к элементам разного типа. Закрепил навыки написания классов на языке C++