**

Государственное образовательное учреждение высшего профессионального образования «Московский Государственный Технический Университет

имени Н.Э. Баумана»

**ОТЧЕТ**

По лабораторной работе №8

По курсу «Анализ алгоритмов»

Тема: «Реализация программы непрерывной обработки данных»

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | Студент: | | | Жарова Е.А. | |
|  | | Группа | | | ИУ7-51 | |
|  | |  | | |  | |
|  | |  | |  | | |
|  | | Москва, 2017 | | |  | |

**Оглавление**

[Постановка задачи 1](#_Toc507885698)

[Описание алгоритма 1](#_Toc507885699)

[Реализация 2](#_Toc507885700)

[Пример работы 3](#_Toc507885701)

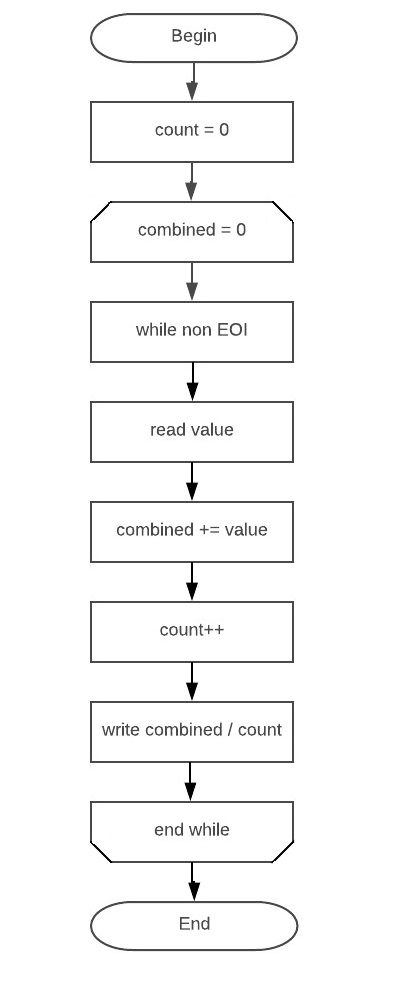
[Заключение 3](#_Toc507885702)

# **Постановка задачи**

Необходимо изучить и реализовать поточный алгоритм обработки данных: реализовать конвейер для выполнения какой-либо задачи. Провести сравнение по производительности с последовательной реализацией.

# **Описание алгоритма**

В ходе работы был реализован алгоритм непрерывного нахождения значения среднего арифметического введенных элементов.



# **Реализация**

Алгоритм был реализован на С++.

#include "stdafx.h"

#include <iostream>

class AverageCounter {

public:

AverageCounter();

double getAverage();

friend std::istream& operator>>(std::istream& input, AverageCounter& counter);

friend std::ostream& operator<<(std::ostream& output, AverageCounter& counter);

private:

void update(double value);

unsigned count;

double combined;

};

AverageCounter::AverageCounter(): count(0), combined(0.0) {}

void AverageCounter::update(double value) {

combined += value;

++count;

}

std::istream& operator>>(std::istream& input, AverageCounter& counter) {

double value = 0;

if (input >> value) {

counter.update(value);

return input;

}

return input >> value;

}

std::ostream& operator<<(std::ostream& output, AverageCounter& counter) {

return output << "Average value: " << counter.getAverage();

}

double AverageCounter::getAverage() {

return combined / count;

}

int \_tmain(int argc, \_TCHAR\* argv[])

{

AverageCounter counter;

while (std::cin >> counter) {

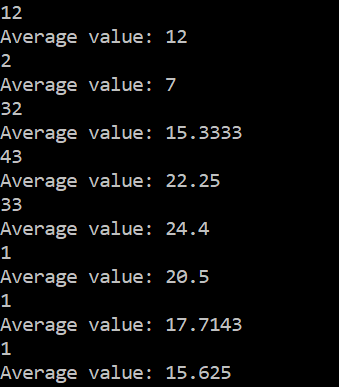
std::cout << counter << std::endl;

}

return 0;

}

# **Пример работы**



# **Заключение**

В ходе работы был реализован алгоритм непрерывной обработки данных на примере алгоритма расчета среднего арифметического значения.