СПЕЦИФИКАЦИЯ ТРЕБОВАНИЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ «СОРТИРОВЩИК ЧИСЕЛ»

СОДЕРЖАНИЕ

[1. ВВЕДЕНИЕ 2](#_Toc130245649)

[1.1. Назначение 2](#_Toc130245650)

[1.2. Соглашения, принятые в документах 2](#_Toc130245651)

[1.3. Границы проекта 2](#_Toc130245652)

[2. ОБЩЕЕ ОПИСАНИЕ 3](#_Toc130245653)

[2.1. Общий взгляд на продукт 3](#_Toc130245654)

[2.2. Классы и характеристики пользователей 3](#_Toc130245655)

[2.3. Операционная среда 4](#_Toc130245656)

[2.4. Ограничения дизайна и реализации 4](#_Toc130245657)

[2.5. Предположения и зависимости 4](#_Toc130245658)

[3. ФУНКЦИИ СИСТЕМЫ 4](#_Toc130245659)

[3.1. Сортировка пузырьком 4](#_Toc130245660)

[3.1.1. Описание сортировки пузырьком 4](#_Toc130245661)

[3.1.2. Функциональные требования сортировки пузырьком 4](#_Toc130245662)

[3.2. Шейкерная сортировка 5](#_Toc130245663)

[3.2.1. Описание шейкерной сортировки 5](#_Toc130245664)

[3.2.2. Функциональные требования шейкерной сортировки 5](#_Toc130245665)

[3.3. Сортировка расческой 5](#_Toc130245666)

[3.3.1. Описание сортировки расческой 5](#_Toc130245667)

[3.3.2. Функциональные требования сортировки расческой 6](#_Toc130245668)

[3.4. Сортировка вставками 6](#_Toc130245669)

[3.4.1. Описание сортировки вставками 6](#_Toc130245670)

[3.4.2. Функциональные требования сортировки вставками 6](#_Toc130245671)

[4. ТРЕБОВАНИЯ К ДАННЫМ 6](#_Toc130245672)

[4.1. Логическая модель данных 7](#_Toc130245673)

[5. ТРЕБОВАНИЯ К ВНЕШНИМ ИНТЕРФЕЙСАМ 7](#_Toc130245674)

[6. АТРИБУТЫ КАЧЕСТВА 8](#_Toc130245675)

[6.2. Удобство использования 8](#_Toc130245676)

[6.3. Производительность 8](#_Toc130245677)

[6.4. Безопасность 9](#_Toc130245678)

[6.5. Техника безопасности 9](#_Toc130245679)

# ВВЕДЕНИЕ

## Назначение

Данная спецификация описывает приложение для сортировки целых чисел. Сортировка – это упорядочение (размещение) чисел по возрастанию или по убыванию их значений. Данный документ предназначен как для непосредственно пользователей, так и тестировщикам.

## Границы проекта

Сортировщик чисел – это программа для выполняющая функцию сортировки входных данных, с возможностью выбора пользователем одного из четырех алгоритма сортировки. Главное предназначение — это быстрое упорядочивание массивов данных различных размеров в короткие сроки.

# ОБЩЕЕ ОПИСАНИЕ

## ****Общий взгляд на продукт****

Данный продукт является сортировкой 4 методами строки чисел. С его помощью можно будет указать количество чисел в строке, их самих и выбрать метод которым будет проводится сортировка, при этом будет указано время потраченное на выбранный метод. Продукт должен обладать простым и удобным интерфейсом, в котором должны быть учтены все требования пользователя.

Основные функции:

1. Возможность ввода количества элементов;
2. Возможность ввода строки чисел;
3. Возможность выбора одного из четырех методов сортировки и применение его к строке чисел;
4. Отображение времени, потраченного на сортировку;

## Классы и характеристики пользователей

Основными пользователями будут являться:

Оператор:

Обычный человек с базовыми знаниями работы с компьтером.

Оператор-тестировщик:

Среди технических навыков тестировщика можно выделить знание одного или нескольких языков программирования, умение работать с базами данных и с консолью. Так же необходимо базовое понимание клиент-серверной архитектуры, умение тестировать API и пользоваться снифферами трафика.

## Операционная среда

Приложение рассчитано на ОС семейства Windows. Предпочтительные версии Windows 2003 и выше.

## Ограничения дизайна и реализации

Возможности дизайна ограничены возможностями Visual Studio, в частности средой разработки С++.

## Предположения и зависимости

До установки данного продукта может потребоваться обновить Microsoft .NET Framework до актуальной версии.

# ФУНКЦИИ СИСТЕМЫ

## Сортировка пузырьком

### Описание сортировки пузырьком

Описание алгоритма сортировки пузырьком заключается в следующем: сначала проходим по массиву слева направо. Если текущий элемент больше следующего, то меняем их местами. Это продолжаем делать, пока массив не будет отсортирован. Важно отметить, что после первой итерации наибольший элемент будет находиться в конце массива на правильном месте. После двух итераций на правильном месте будут находиться два наибольших элемента, и так далее. Однако, необходимо учитывать, что массив будет отсортирован не более чем после n итераций. В худшем и среднем случае асимптотика составляет O(n2), а в лучшем случае - O(n).

### Функциональные требования сортировки пузырьком

К таким требованиям отнесем:

* Корректный массив чисел
* Корректно заданный размер массива

## Шейкерная сортировка

### Описание шейкерной сортировки

Шейкерная сортировка представляет собой улучшенную версию сортировки пузырьком, которая может работать эффективнее на тестах, где маленькие элементы находятся в конце массива. В отличие от сортировки пузырьком, при которой мы проходим по массиву слева направо и меняем местами соседние элементы, шейкерная сортировка проходит по массиву в обе стороны, поддерживая два указателя begin и end, которые обозначают, какой отрезок массива еще не отсортирован. При достижении end на очередной итерации мы вычитаем из него единицу и движемся справа налево, аналогично, при достижении begin прибавляем единицу и двигаемся слева направо. Время выполнения шейкерной сортировки такое же, как и у сортировки пузырьком, однако она работает быстрее на практике.

### Функциональные требования шейкерной сортировки

К таким требованиям отнесем:

* Корректный массив чисел
* Корректно заданный размер массива

## Сортировка расческой

### Описание сортировки расческой

Описание сортировки расчёской. Эта сортировка является модификацией сортировки пузырьком и позволяет избежать «черепах» за счет перестановки элементов, находящихся на определенном расстоянии друг от друга. На каждом шаге алгоритма зафиксируем расстояние между элементами и будем идти по массиву слева направо, сравнивая и переставляя элементы на данном расстоянии, если это необходимо. Это помогает быстрее переместить большие элементы в начало массива. Изначально расстояние выбирается равным длине массива, а затем уменьшается на определенный коэффициент, приблизительно равный 1.247. Когда расстояние становится равным единице, выполняется сортировка пузырьком. В лучшем случае асимптотика этой сортировки равна O(nlogn), в худшем - O(n2). В среднем случае асимптотика также составляет O(nlogn), но на практике это может немного отличаться.

### Функциональные требования сортировки расческой

К таким требованиям отнесем:

* Корректный массив чисел
* Корректно заданный размер массива

## Сортировка вставками

### Описание сортировки вставками

СОписание сортировки вставками заключается в том, что элементы из исходного массива поочередно вставляются в новый массив таким образом, чтобы последний всегда оставался отсортированным. Асимптотика этого алгоритма в среднем и худшем случае - O(n^2), а в лучшем - O(n). Реализация алгоритма может быть упрощена следующим образом: вместо того, чтобы создавать новый массив и вставлять в него элементы, сортируется префикс исходного массива путем последовательного перестановки текущего элемента с предыдущим до тех пор, пока они не будут стоять в правильном порядке.

### Функциональные требования сортировки вставками

К таким требованиям отнесем:

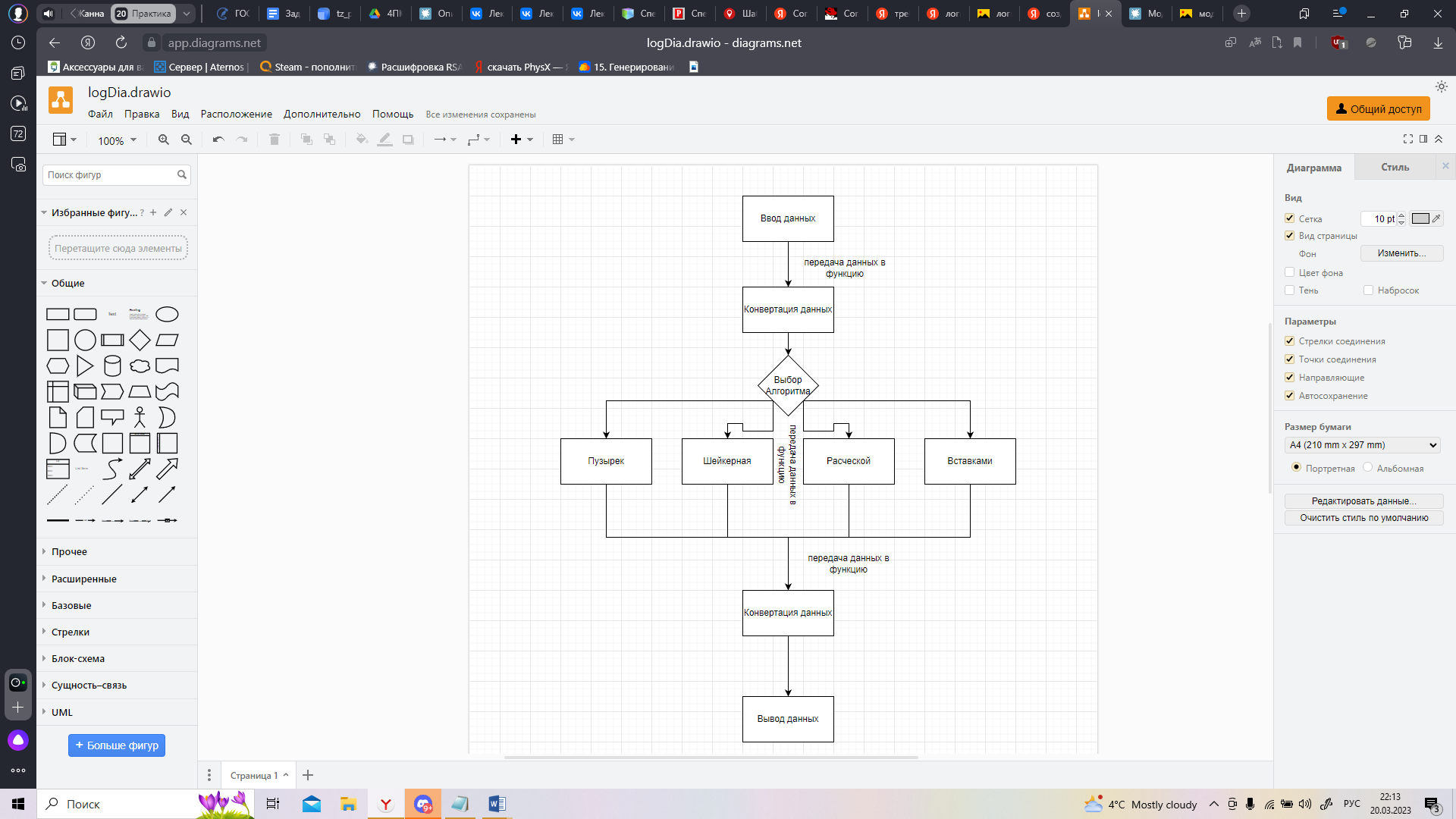
* Корректный массив чисел
* Корректно заданный размер массива

# ТРЕБОВАНИЯ К ДАННЫМ

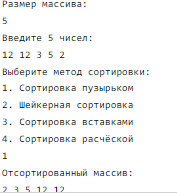
Основными типами данных в программе являются: int и string.

Алгоритмы сортировки принимают исключительно int, но входные данные всегда будут string. Поэтому нам требуется конвертировать типы данных.

## Логическая модель данных



# ТРЕБОВАНИЯ К ВНЕШНИМ ИНТЕРФЕЙСАМ

Графический интерфейс реализован в файл консоле. Выглядит он следующим образом:  


Исходный массив, отсортированный массив, размер массива и названия сортировок – это элементы графического интерфейса, позволяющие пользователю ориентироваться в программе и корректно вводить данные.

*Отсортировать* – функциональная кнопка, запускающая сам процесс сортировки.

Поля для ввода и вывода представлены белыми полями – TextBox.

Аппаратные интерфейсы это: стандартное оборудование компьютера, включающее монитор, клавиатуру, мышь, модем.

# АТРИБУТЫ КАЧЕСТВА

## Удобство использования

Минималистичный дизайн интерфейса, позволяет легко ориентироваться в программе.

## Производительность

Программа была оптимизирована для работы на слабых ПК.

## Безопасность

Так как программа не работает с данными которые несут конфиденциальную вопрос безопасности не поднимался.

## Техника безопасности

Особых требований к ТБ нет.