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.

## Соглашения, принятые в документах

В данном документе вы встретите определённые слова, выделенные в тексте различными шрифтами, стилями, размерами. Такое оформление не случайно; разные слова отображаются одним стилем, когда они принадлежат определённой категории. Примеры слов, оформленных таким образом, приведены ниже:

имя файла

Таким стилем оформлены имена файлов, названия каталогов, пути, названия RPM пакетов.

**приложение**

Такое оформление означает, что указанная программа — приложение конечного пользователя.

текст, отображаемый в графическом интерфейсе

Заголовок, слово, или фраза, выводимые в графическом экране или окне оформляются таким стилем. Текст с таким оформлением используется для обозначения конкретного графического экрана или элемента на графическом экране (например, текста, связанного с флажком или полем ввода)

*кнопка в графическом экране или окне*

Этим стилем оформляется название кнопки на графическом экране.

## 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# c Windows Form.

## Предположения и зависимости

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

# ФУНКЦИИ СИСТЕМЫ

## Сортировка пузырьком

### Описание сортировки пузырьком

Будем идти по массиву слева направо. Если текущий элемент больше следующего, меняем их местами. Делаем так, пока массив не будет отсортирован. Заметим, что после первой итерации самый большой элемент будет находиться в конце массива, на правильном месте. После двух итераций на правильном месте будут стоять два наибольших элемента, и так далее. Очевидно, не более чем после n итераций массив будет отсортирован. Таким образом, асимптотика в худшем и среднем случае – O(n2), 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

Заметим, что сортировка пузырьком работает медленно на тестах, в которых маленькие элементы стоят в конце (их еще называют «черепахами»). Такой элемент на каждом шаге алгоритма будет сдвигаться всего на одну позицию влево. Поэтому будем идти не только слева направо, но и справа налево. Будем поддерживать два указателя begin и end, обозначающих, какой отрезок массива еще не отсортирован. На очередной итерации при достижении end вычитаем из него единицу и движемся справа налево, аналогично, при достижении begin прибавляем единицу и двигаемся слева направо. Асимптотика у алгоритма такая же, как и у сортировки пузырьком, однако реальное время работы лучше.

### Функциональные требования шейкерной сортировки

К таким требованиям отнесем:

* Корректный массив чисел
* Корректно заданный размер массива

## Сортировка расческой

### Описание сортировки расческой

Еще одна модификация сортировки пузырьком. Для того, чтобы избавиться от «черепах», будем переставлять элементы, стоящие на расстоянии. Зафиксируем его и будем идти слева направо, сравнивая элементы, стоящие на этом расстоянии, переставляя их, если необходимо. Очевидно, это позволит «черепахам» быстро добраться в начало массива. Оптимально изначально взять расстояние равным длине массива, а далее делить его на некоторый коэффициент, равный примерно 1.247. Когда расстояние станет равно единице, выполняется сортировка пузырьком. В лучшем случае асимптотика равна O(nlogn), в худшем – O(n2). Какая асимптотика в среднем мне не очень понятно, на практике похоже на O(nlogn).

### Функциональные требования сортировки расческой

К таким требованиям отнесем:

* Корректный массив чисел
* Корректно заданный размер массива

## Сортировка вставками

### Описание сортировки вставками

Создадим массив, в котором после завершения алгоритма будет лежать ответ. Будем поочередно вставлять элементы из исходного массива так, чтобы элементы в массиве-ответе всегда были отсортированы. Асимптотика в среднем и худшем случае – O(n2), в лучшем – O(n). Реализовывать алгоритм удобнее по-другому (создавать новый массив и реально что-то вставлять в него относительно сложно): просто сделаем так, чтобы отсортирован был некоторый префикс исходного массива, вместо вставки будем менять текущий элемент с предыдущим, пока они стоят в неправильном порядке.

### Функциональные требования сортировки вставками

К таким требованиям отнесем:

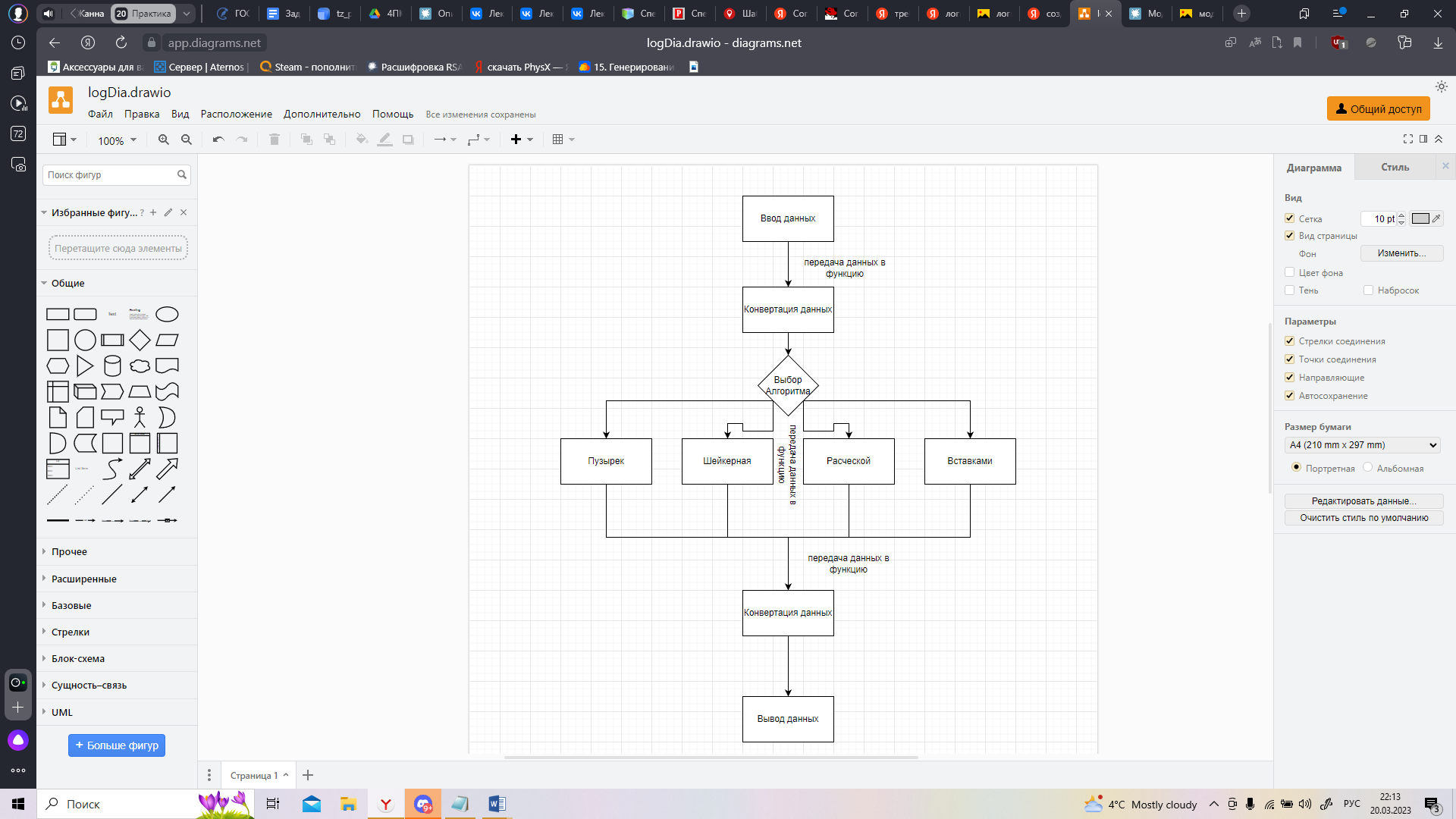
* Корректный массив чисел
* Корректно заданный размер массива

# ТРЕБОВАНИЯ К ДАННЫМ

Основными типами данных в программе являются: int и string.

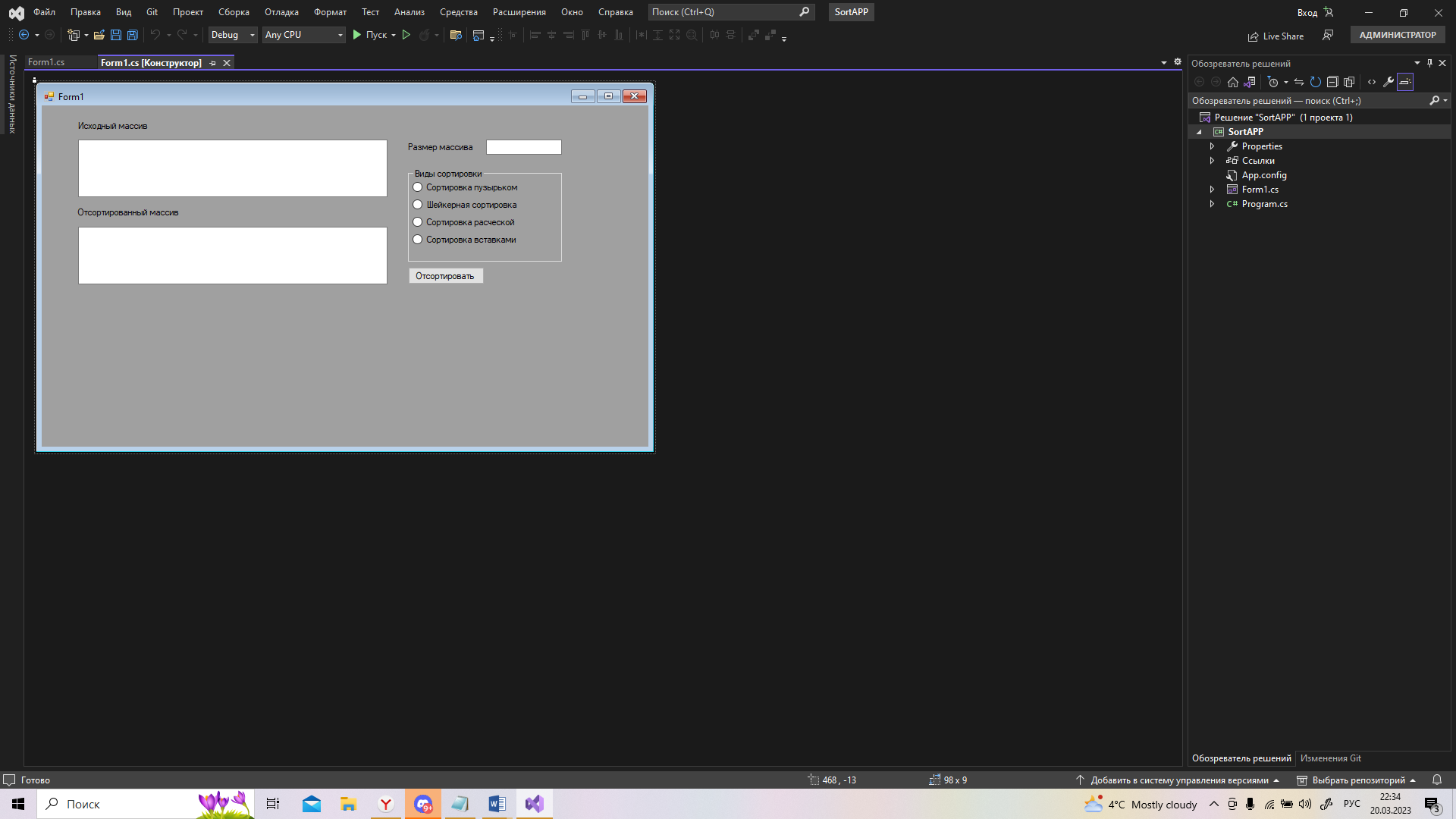
Алгоритмы сортировки принимают исключительно int, но входные данные всегда будут string. Поэтому нам требуется конвертировать типы данных.

## Логическая модель данных



# ТРЕБОВАНИЯ К ВНЕШНИМ ИНТЕРФЕЙСАМ

Графический интерфейс реализован в файле Form1.cs. 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.