СПЕЦИФИКАЦИЯ ТРЕБОВАНИЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ «GoSorting!»

СОДЕРЖАНИЕ

[1. ВВЕДЕНИЕ 2](file:///C:\Users\Liza\AppData\Local\Temp\Rar$DIa8108.4575\Спецификация%20практика.docx#_Toc130245649)

[1.1. Назначение 2](file:///C:\Users\Liza\AppData\Local\Temp\Rar$DIa8108.4575\Спецификация%20практика.docx#_Toc130245650)

[1.2. Соглашения, принятые в документах 2](file:///C:\Users\Liza\AppData\Local\Temp\Rar$DIa8108.4575\Спецификация%20практика.docx#_Toc130245651)

[1.3. Границы проекта 2](file:///C:\Users\Liza\AppData\Local\Temp\Rar$DIa8108.4575\Спецификация%20практика.docx#_Toc130245652)

[2. ОБЩЕЕ ОПИСАНИЕ 3](file:///C:\Users\Liza\AppData\Local\Temp\Rar$DIa8108.4575\Спецификация%20практика.docx#_Toc130245653)

[2.1. Общий взгляд на продукт 3](file:///C:\Users\Liza\AppData\Local\Temp\Rar$DIa8108.4575\Спецификация%20практика.docx#_Toc130245654)

[2.2. Классы и характеристики пользователей 3](file:///C:\Users\Liza\AppData\Local\Temp\Rar$DIa8108.4575\Спецификация%20практика.docx#_Toc130245655)

[2.3. Операционная среда 4](file:///C:\Users\Liza\AppData\Local\Temp\Rar$DIa8108.4575\Спецификация%20практика.docx#_Toc130245656)

[2.4. Ограничения дизайна и реализации 4](file:///C:\Users\Liza\AppData\Local\Temp\Rar$DIa8108.4575\Спецификация%20практика.docx#_Toc130245657)

[2.5. Предположения и зависимости 4](file:///C:\Users\Liza\AppData\Local\Temp\Rar$DIa8108.4575\Спецификация%20практика.docx#_Toc130245658)

[3. ФУНКЦИИ СИСТЕМЫ 4](file:///C:\Users\Liza\AppData\Local\Temp\Rar$DIa8108.4575\Спецификация%20практика.docx#_Toc130245659)

[3.1. Сортировка пузырьком 4](file:///C:\Users\Liza\AppData\Local\Temp\Rar$DIa8108.4575\Спецификация%20практика.docx#_Toc130245660)

[3.1.1. Описание сортировки пузырьком 4](file:///C:\Users\Liza\AppData\Local\Temp\Rar$DIa8108.4575\Спецификация%20практика.docx#_Toc130245661)

[3.1.2. Функциональные требования сортировки пузырьком 4](file:///C:\Users\Liza\AppData\Local\Temp\Rar$DIa8108.4575\Спецификация%20практика.docx#_Toc130245662)

[3.2. Шейкерная сортировка 5](file:///C:\Users\Liza\AppData\Local\Temp\Rar$DIa8108.4575\Спецификация%20практика.docx#_Toc130245663)

[3.2.1. Описание шейкерной сортировки 5](file:///C:\Users\Liza\AppData\Local\Temp\Rar$DIa8108.4575\Спецификация%20практика.docx#_Toc130245664)

[3.2.2. Функциональные требования шейкерной сортировки 5](file:///C:\Users\Liza\AppData\Local\Temp\Rar$DIa8108.4575\Спецификация%20практика.docx#_Toc130245665)

[3.3. Сортировка расческой 5](file:///C:\Users\Liza\AppData\Local\Temp\Rar$DIa8108.4575\Спецификация%20практика.docx#_Toc130245666)

[3.3.1. Описание сортировки расческой 5](file:///C:\Users\Liza\AppData\Local\Temp\Rar$DIa8108.4575\Спецификация%20практика.docx#_Toc130245667)

[3.3.2. Функциональные требования сортировки расческой 6](file:///C:\Users\Liza\AppData\Local\Temp\Rar$DIa8108.4575\Спецификация%20практика.docx#_Toc130245668)

[3.4. Сортировка вставками 6](file:///C:\Users\Liza\AppData\Local\Temp\Rar$DIa8108.4575\Спецификация%20практика.docx#_Toc130245669)

[3.4.1. Описание сортировки вставками 6](file:///C:\Users\Liza\AppData\Local\Temp\Rar$DIa8108.4575\Спецификация%20практика.docx#_Toc130245670)

[3.4.2. Функциональные требования сортировки вставками 6](file:///C:\Users\Liza\AppData\Local\Temp\Rar$DIa8108.4575\Спецификация%20практика.docx#_Toc130245671)

[4. ТРЕБОВАНИЯ К ДАННЫМ 6](file:///C:\Users\Liza\AppData\Local\Temp\Rar$DIa8108.4575\Спецификация%20практика.docx#_Toc130245672)

[4.1. Логическая модель данных 7](file:///C:\Users\Liza\AppData\Local\Temp\Rar$DIa8108.4575\Спецификация%20практика.docx#_Toc130245673)

[5. ТРЕБОВАНИЯ К ВНЕШНИМ ИНТЕРФЕЙСАМ 7](file:///C:\Users\Liza\AppData\Local\Temp\Rar$DIa8108.4575\Спецификация%20практика.docx#_Toc130245674)

[6. АТРИБУТЫ КАЧЕСТВА 8](file:///C:\Users\Liza\AppData\Local\Temp\Rar$DIa8108.4575\Спецификация%20практика.docx#_Toc130245675)

[6.2. Удобство использования 8](file:///C:\Users\Liza\AppData\Local\Temp\Rar$DIa8108.4575\Спецификация%20практика.docx#_Toc130245676)

[6.3. Производительность 8](file:///C:\Users\Liza\AppData\Local\Temp\Rar$DIa8108.4575\Спецификация%20практика.docx#_Toc130245677)

[6.4. Безопасность 9](file:///C:\Users\Liza\AppData\Local\Temp\Rar$DIa8108.4575\Спецификация%20практика.docx#_Toc130245678)

[6.5. Техника безопасности 9](file:///C:\Users\Liza\AppData\Local\Temp\Rar$DIa8108.4575\Спецификация%20практика.docx#_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

GoSorting! – 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 задать массив целых чисел и выбрать метод, которым будет проводиться сортировка. Продукт должен обладать простым и удобным интерфейсом, в котором должны быть учтены все требования пользователя.

Основные функции:

1. Возможность ввода строки чисел;
2. Возможность выбора одного из четырех методов сортировки и применение его к строке чисел;
3. Вывод отсортированного массива.

## Классы и характеристики пользователей

Основными пользователями будут являться:

Оператор:

Обычный человек с базовыми знаниями работы с компьютером.

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.

## Операционная среда

Приложение кроссплатформенное. Поддерживаемые ОС: Linux, Unix-like, macOS, Windows, IOS, Android.

## Ограничения дизайна и реализации

Возможности дизайна ограничены возможностями инструментария виджетов Fyne.

## Предположения и зависимости

Приложение не требует предварительной установки дополнительных зависимостей.

# 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:

* Корректно заданный массив чисел

## Сортировка выбором

### Описание сортировки выбором

Сортировка выбором — это алгоритм сортировки массивов, в котором на каждой итерации во всей последовательности неотсортированных данных выбирается минимальный элемент (при сортировке по возрастанию) и помещается в первую позицию неотсортированной последовательности. Тем самым готовая (отсортированная) последовательность увеличивается на один элемент, а исходная (неотсортированная) последовательность на один элемент уменьшается.

### Функциональные требования сортировки выбором

К таким требованиям отнесем:

* Корректно заданный массив чисел

## Сортировка Шелла

### Описание сортировки Шелла

Сортировка Шелла - усовершенствованная разновидность сортировки вставками.

Сначала сравниваются и сортируются между собой значения, стоящие друг от друга на некотором расстоянии - d. После этого расстояние d уменьшается, и процедура повторяется до тех пор, пока значение d не станет минимальным, т. е. d = 1. Это означает, что сортировка достигла последнего шага. А на последнем шага элементы сортируются обычной сортировкой вставками.

### Функциональные требования сортировки Шелла

К таким требованиям отнесем:

* Корректно заданный массив чисел

## 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:

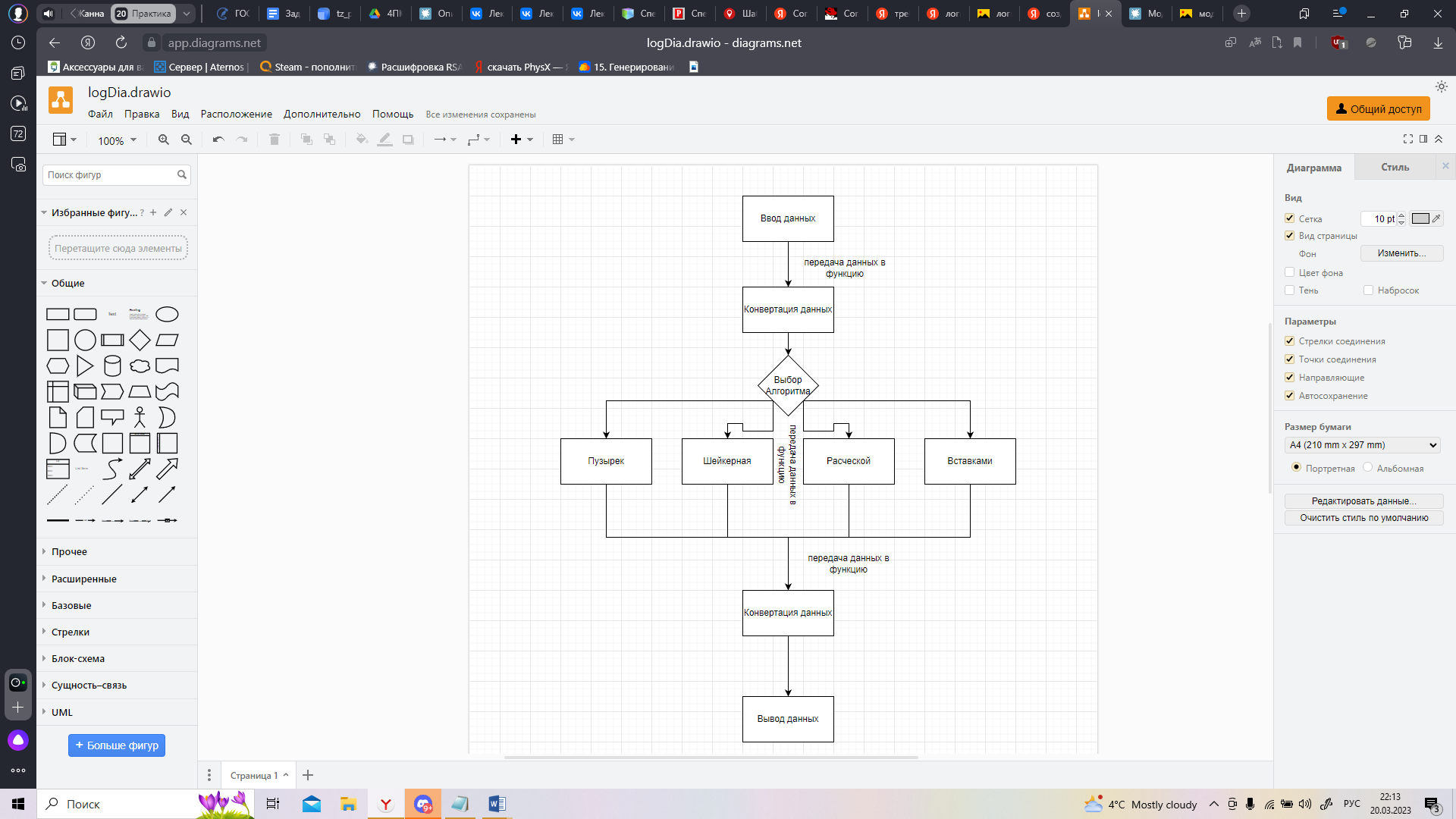
* Корректно заданный массив чисел

# 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 выглядит следующим образом:

Изображение выглядит как стол

Автоматически созданное описание

Исходный массив, отсортированный массив, кнопки выбора метода сортировки – 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 – NewEntry.

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 информацию, вопрос безопасности не поднимался.

## Техника безопасности

Особых требований к ТБ нет.