

# **ОПИСАНИЕ АЛГОРИТМОВ РАБОТЫ ПРОГРАММЫ**

## **ВВЕДЕНИЕ**

В данном документе описаны два основных алгоритма работы системы CURSA:

1. Проверка документа на соответствие стандарту
2. Автоматическое исправление документа

Для каждого алгоритма представлена блок-схема и описание последовательности действий.

## **1 АЛГОРИТМ ПРОВЕРКИ ДОКУМЕНТА**

### **1.1 Что делает алгоритм**

Алгоритм проверяет загруженный документ (формат DOCX) на соответствие выбранному стандарту (например, ГОСТ 7.32-2017). Находит все ошибки форматирования и выставляет оценку документу.

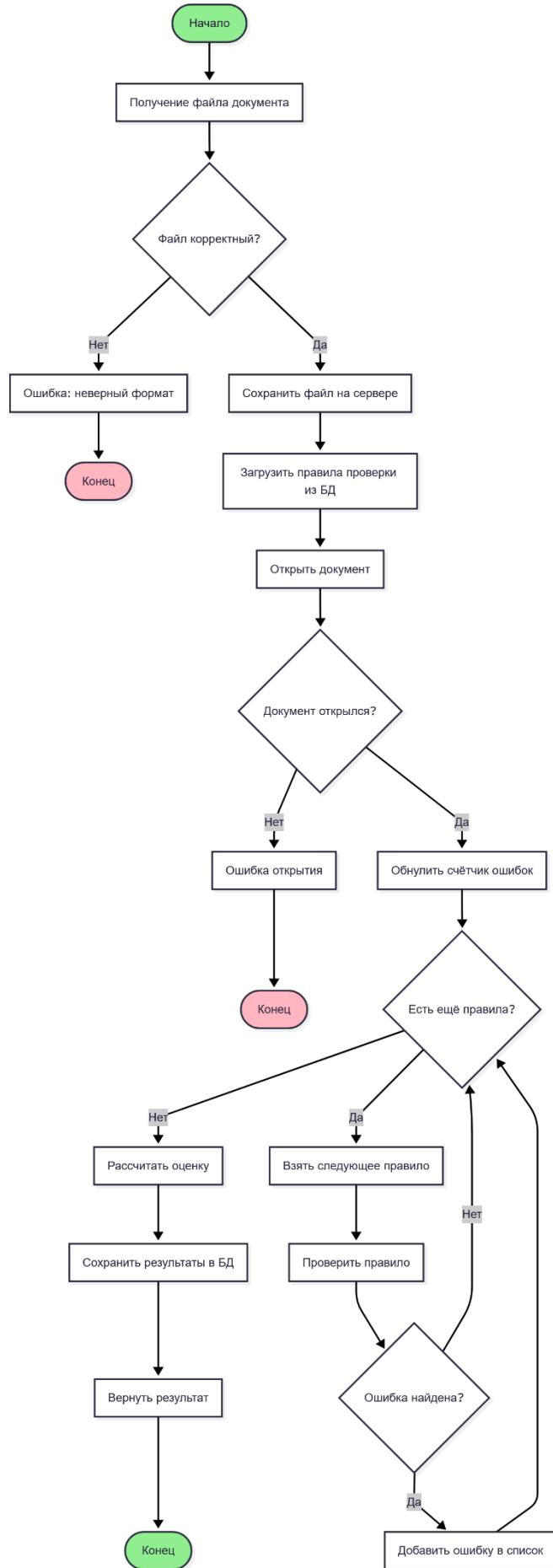
### **1.2 Что получает на вход**

- Файл документа (например, diplom\_ivanov.docx)
- Выбранный стандарт проверки (например, ГОСТ 7.32-2017)
- ID пользователя

### **1.3 Что возвращает**

- Список найденных ошибок с описанием
- Оценка документа от 0 до 100 баллов
- Статус проверки (успешно/ошибка)

## 1.4 Блок-схема алгоритма



## **1.5 Как работает алгоритм**

**Шаг 1.** Программа получает файл документа от пользователя.

**Шаг 2.** Проверяется, что файл имеет формат DOCX и размер не превышает 10 МБ. Если файл не подходит — выдаётся ошибка.

**Шаг 3.** Файл сохраняется на сервере с уникальным именем.

**Шаг 4.** Из базы данных загружаются все правила для выбранного стандарта. Например, для ГОСТ 7.32-2017 это:

- Поля: слева 30 мм, справа 15 мм
- Шрифт: Times New Roman, 14 пт
- Межстрочный интервал: 1.5
- И другие правила

**Шаг 5.** Документ открывается для анализа.

**Шаг 6.** Программа проходит по всем правилам и проверяет документ:

- Правильные ли поля?
- Правильный ли шрифт?
- Правильный ли интервал?
- И так далее

**Шаг 7.** Каждая найденная ошибка записывается в список с указанием:

- Где находится (страница, абзац)
- Что не так (например, "шрифт Arial вместо Times New Roman")
- Насколько это важно (критично/средне/незначительно)

**Шаг 8.** Рассчитывается оценка по формуле:

$$\text{Оценка} = 100 - (\text{критичные} \times 10 + \text{средние} \times 5 + \text{незначительные} \times 2)$$

Если получается меньше 0, ставится 0.

**Шаг 9.** Все результаты сохраняются в базе данных.

**Шаг 10.** Программа возвращает пользователю список ошибок и оценку.

## **2 АЛГОРИТМ АВТОМАТИЧЕСКОГО ИСПРАВЛЕНИЯ**

### **2.1 Что делает алгоритм**

Алгоритм исправляет найденные ошибки автоматически (те, которые можно исправить без участия человека). Например, меняет размер полей, шрифт, интервалы.

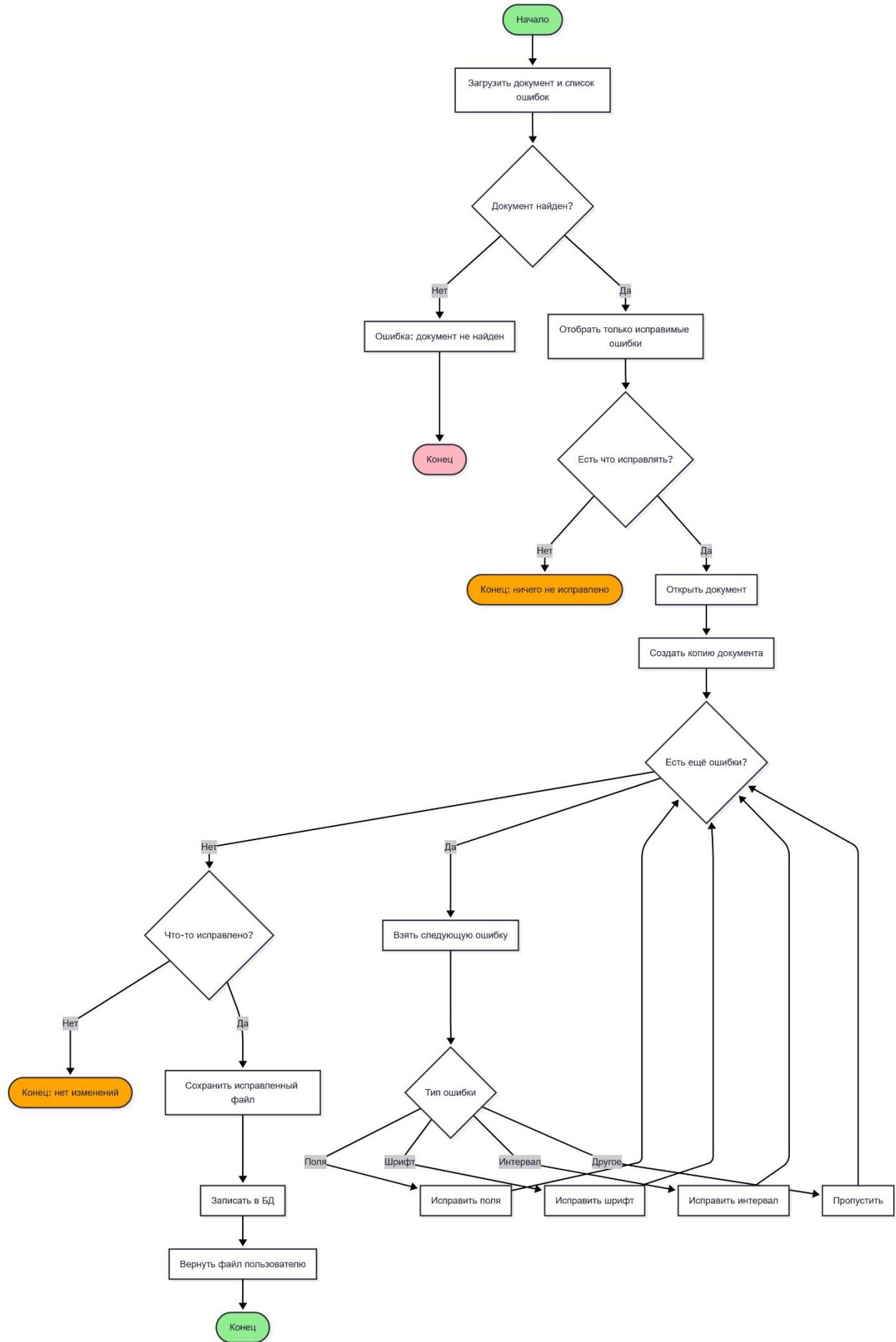
### **2.2 Что получает на вход**

- ID документа, который нужно исправить
- Список ошибок из проверки

### **2.3 Что возвращает**

- Исправленный файл
- Количество исправленных ошибок
- Статус (успешно/частично/ошибка)

## 2.4 Блок-схема алгоритма



## **2.5 Как работает алгоритм**

**Шаг 1.** Программа загружает документ и список найденных ошибок из базы данных.

**Шаг 2.** Из всех ошибок выбираются только те, которые можно исправить автоматически. Это:

- Неправильные поля страницы
- Неправильный шрифт (название и размер)
- Неправильный межстрочный интервал
- Неправильное выравнивание текста

Ошибки, которые нельзя исправить автоматически (например, отсутствие заголовка), пропускаются.

**Шаг 3.** Если исправимых ошибок нет, алгоритм завершается — менять нечего.

**Шаг 4.** Документ открывается, создаётся его копия для исправления (оригинал не трогаем).

**Шаг 5.** Программа проходит по всем исправимым ошибкам:

### **Исправление полей:**

Установить левое поле = 30 мм

Установить правое поле = 15 мм

Установить верхнее поле = 20 мм

Установить нижнее поле = 20 мм

### **Исправление шрифта:**

Установить название шрифта = Times New Roman

Установить размер = 14 пт

### **Исправление интервала:**

Установить межстрочный интервал = 1.5

**Шаг 6.** После исправления всех ошибок проверяется, было ли что-то изменено. Если нет — документ не сохраняется.

**Шаг 7.** Исправленный документ сохраняется с новым именем (например, `diplom_ivanov_corrected.docx`).

**Шаг 8.** Информация об исправлении записывается в базу данных.

**Шаг 9.** Программа возвращает пользователю ссылку на скачивание исправленного файла.

### **3 ОБОЗНАЧЕНИЯ В БЛОК-СХЕМАХ**

В блок-схемах используются следующие элементы (по ГОСТ 19.701-90):

- **Овал** — начало или конец программы
- **Прямоугольник** — действие (вычисление, присваивание)
- **Ромб** — проверка условия (да/нет)
- **Параллелограмм** — ввод или вывод данных
- **Стрелки** — направление выполнения программы

**Цвета в схемах:**

- Зелёный — начало работы или успешное завершение
- Красный — ошибка или завершение с ошибкой
- Оранжевый — завершение без изменений

### **ЗАКЛЮЧЕНИЕ**

В документе описаны два основных алгоритма системы CURSA:

1. **Алгоритм проверки** — анализирует документ, находит все ошибки форматирования и выставляет оценку.

**2. Алгоритм исправления** — автоматически исправляет найденные ошибки (поля, шрифт, интервалы).

Оба алгоритма работают с файлами формата DOCX и используют базу данных для хранения результатов.