Санкт-Петербургский политехнический университет Петра Великого

Кафедра компьютерных систем и программных технологий

**Отчёт по проекту**

**Дисциплина**: Алгоритмы и структуры данных

**Тема:** Создание GUI приложения для генерации случайного кроссворда с заданными пользователем словами

Выполнил студент гр. 3530901/90003 В.С. Андрианов

(подпись)

Преподаватель М.Х. Ахин

(подпись)

“ ” 2020 г.

Санкт-Петербург

2020

**Оглавление**

[**Техническое задание** 3](#_Toc26066265)

[**GitHub** 4](#_Toc26066266)

[**Метод решения** 5](#_Toc26066266)

[**Описание классов и методов** 6](#_Toc26066266)

[**Работа программы** 10](#_Toc26066266)

# **ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

1. Сведения о приложении

С помощью текстового файла пользователь будет вводить желаемые слова в специальное поле. После чего алгоритм будет располагать слова в кроссворде, учитывая пересечения. Минимальное количество слов – 3. Также приложение будет отрисовывать и представлять кроссворд с заданными словами. Приложение будет написано на языке Kotlin (с помощью TornadoFx);

* Распределение и отрисовка будет проходить на поле размером 10 на 10 клеток (нужные клетки будут перекрашиваться их белого цвета в другой);
* Приложение будет работать в три этапа:

1. этап старта приложения (будет две кнопки: “Старт”, “Выйти”
2. этап ввода слов с помощью файла (также будет кнопка начала представления кроссворда)
3. этап представления распределенных и отрисованных слов (на этом этапе будет две кнопки: “Новый кроссворд”, “Выход”)
4. Примерный вид приложения



Рис. 1 Главный экран

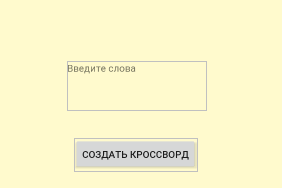


Рис. 2 Ввод слов

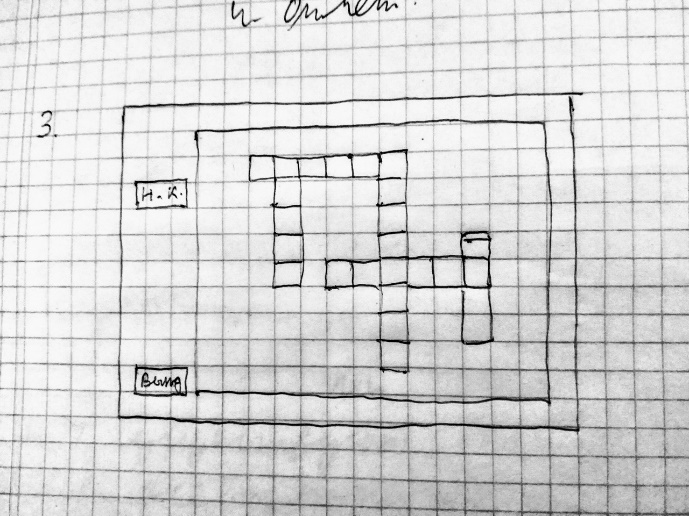


Рис. 3 Созданный кроссворд

**GitHub репозиторий проекта:** <https://github.com/SlimeLoveAllTheTime/CrossWord>

# **Метод решения**

В основе решения лежит две наиболее важных идеи.

Первая – дополнить существующий двумерный массив (или сетку) несколькими возможностями. А именно, логическая сетка должна уметь удалять столбцы и строки, проверять конкретный столбец (или строку) на содержание элементов, вставлять в столбец или строку слово и менять размер. Такой подход помогает не ограничивать количество возможных слов, из которых можно составить кроссворд.

Вторая – реализовать возможность создания кроссворда, и, впоследствии, решение его. Работа производится непосредственно со словами. Сначала слова сортируются по длине. Начиная с наиболее длинных слов общее решение будет проще строиться, потому что наиболее длинные слова имеют наибольшее количество символов, и, в следствии, наибольшее количество пересечений с другими словами. Далее первое слово размещается в середине сетки. После этого остальные слова располагаются в сетки. Если финальное решение не было получено, то создатель кроссвордов начнет строить решение начиная со второго слова и так далее. Стоит отметить, что создатель кроссворда в любом случае выдаст результат, даже если все слова не удастся расположить в сетки. Такой подход помогает получить наилучшее решение, если не удается создать финальное.

С помощью двух этих идей строится основной метод решения. Создатель кроссвордов получает список слов и сортирует его. Далее он перебирает список со словами, а именно берет слово и начинает строить вокруг него кроссворд. Если кроссворд удалось построить, то он останавливается. Если не удалось, он начинает строить кроссворд вокруг следующего слова. Делает он это до того момента, как найдет решение или же переберет все слова. Если решение не удалось найти, то создатель кроссвордов представит наиболее выгодное решение по общей длине задействованных символов.

**Описание классов и методов**

**Класс MainView**

Класс, который строит все представление перед пользователем. Стоит отметить, что все окна приложения строятся на одной сцене, путем удаления не нужных объектов.

**Init{}**

Инициализируем и строим первые два окна приложения

**fun createCrossWord()**

Метод, который создает отображения получившегося кроссворда. Стоит сказать, что ячейки сетки (GridPane) – это кнопки.

**Класс Styles**

Некоторые стили для представления в главном классе представления

**Класс CrossWord**

 Класс Кроссворд - главный класс логики, который создает нужную сетку, располагает слова и "решает" его. В качестве параметра класса передаем список слов.

**fun solver()**

Метод, который выдает финальное решение, то есть располагает все слова в сетке, изменяет размеры финальной сетки, выдает лучшее решение, если полное решение невозможно.

**fun placeWord()**

 Метод, который получает слово (которое необходимо расположить) и размещает его в сетку.

**fun canPlace()**

Метод, который проверяет можно ли расположить указанное слово в сетку (начиная с указанного места (координат))

**fun coordinates()**

Метод, который вычисляет координаты расположения слова, которые было найдено с пересечением предыдущего

**fun wordsBePlaced()**

Метод, который проверяет все ли слова были расположены в сетке

**fun setSomeCoordinates()**

Метод, который записывает в список someCoordinates координаты всех слов

**fun defaultCoordinates()**

Метод, который задает всем словам дефолтные координаты (-1, -1)

**Класс Grid**

Класс Сетка – это класс, который собой представляет логическую сетку (логический grid), где расставляются буквы слов. По факту это просто таблица, в которой потом будут располагаться буквы.

**fun listOfWords()**

Метод, который преобразует файл со словами в список слов

**fun createGrayGrid()**

Метод, который заполняет таблицу пустыми элементами. Пустой элемент – это ‘#’, и называется он grayChar. Такое название из-за того, что в представлении пустая клетка сетки имеет серый цвет

**fun swap()**

Метод, который меняет значения двух клеток местами

**fun placeWordVertical()**

Метод, который вертикально располагает слово в сетки

**fun placeWordHorizontal()**

Метод, который горизонтально располагает слово в сетки

**fun columnIsEmpty()**

Метод, который проверяет столбец на наличие элементов

**fun rowIsEmpty()**

Метод, который проверяет строку на наличие элементов

**fun removeColumn()**

метод, который удаляет заданный столбец

**fun removeRow()**

метод, который удаляет заданную строку

**fun resizeGrid()**

Метод, который меняет размер сетки, путем удаления пустых столбцов и строк

**Класс Word**

 Класс Word – это класс, который дополняет обычное представление слова в виде строки. А именно класс-оболочка строки, который также задает координаты начала (первой буквы) слова на доске, вид расположения слова на доске

**fun isPlaced()**

Метод, который проверят расположено ли слово

**fun setCoordinates()**

Метод, который назначает координаты слову

**Работа программы**

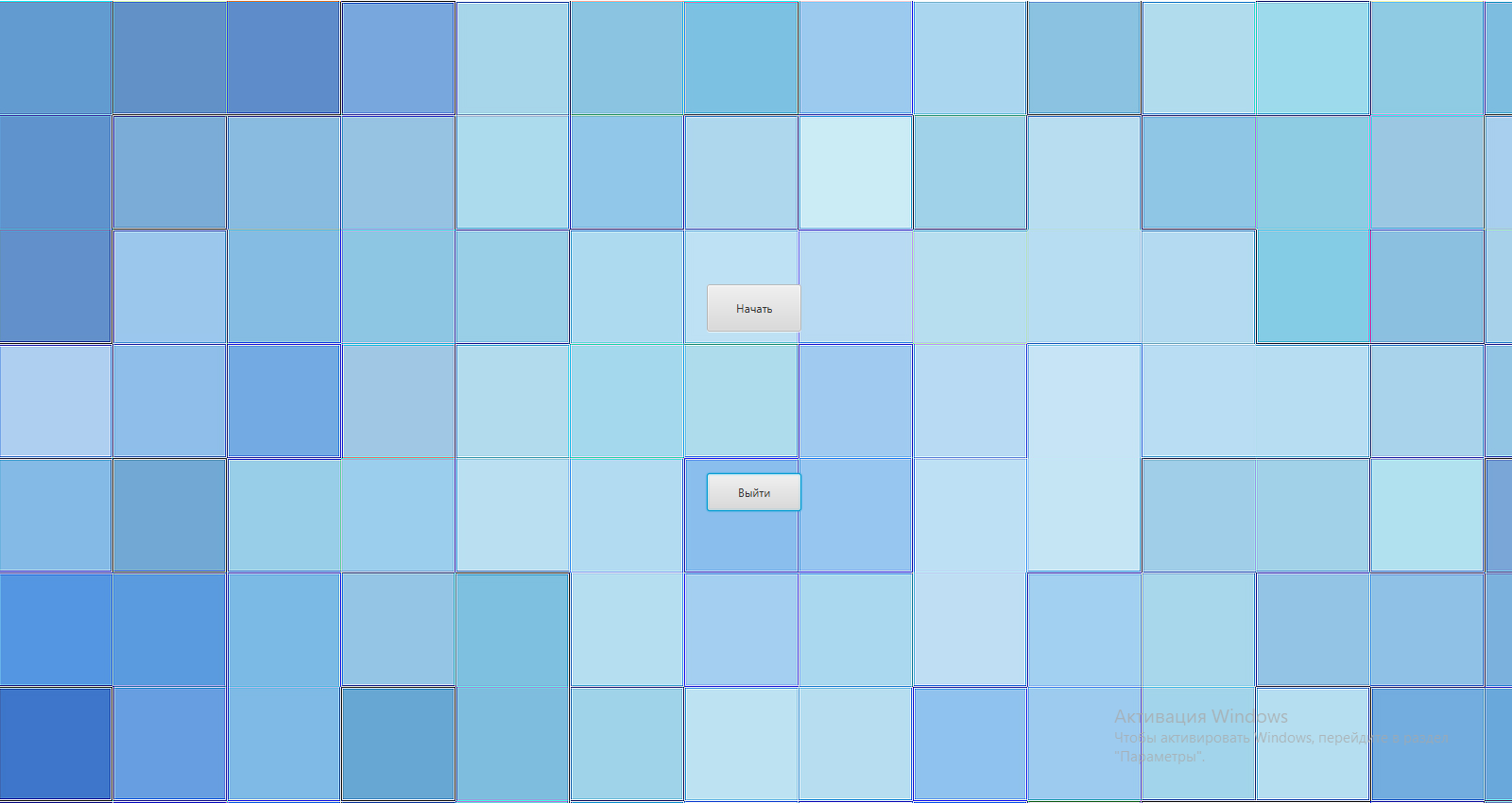


Рис. 4 Главный экран

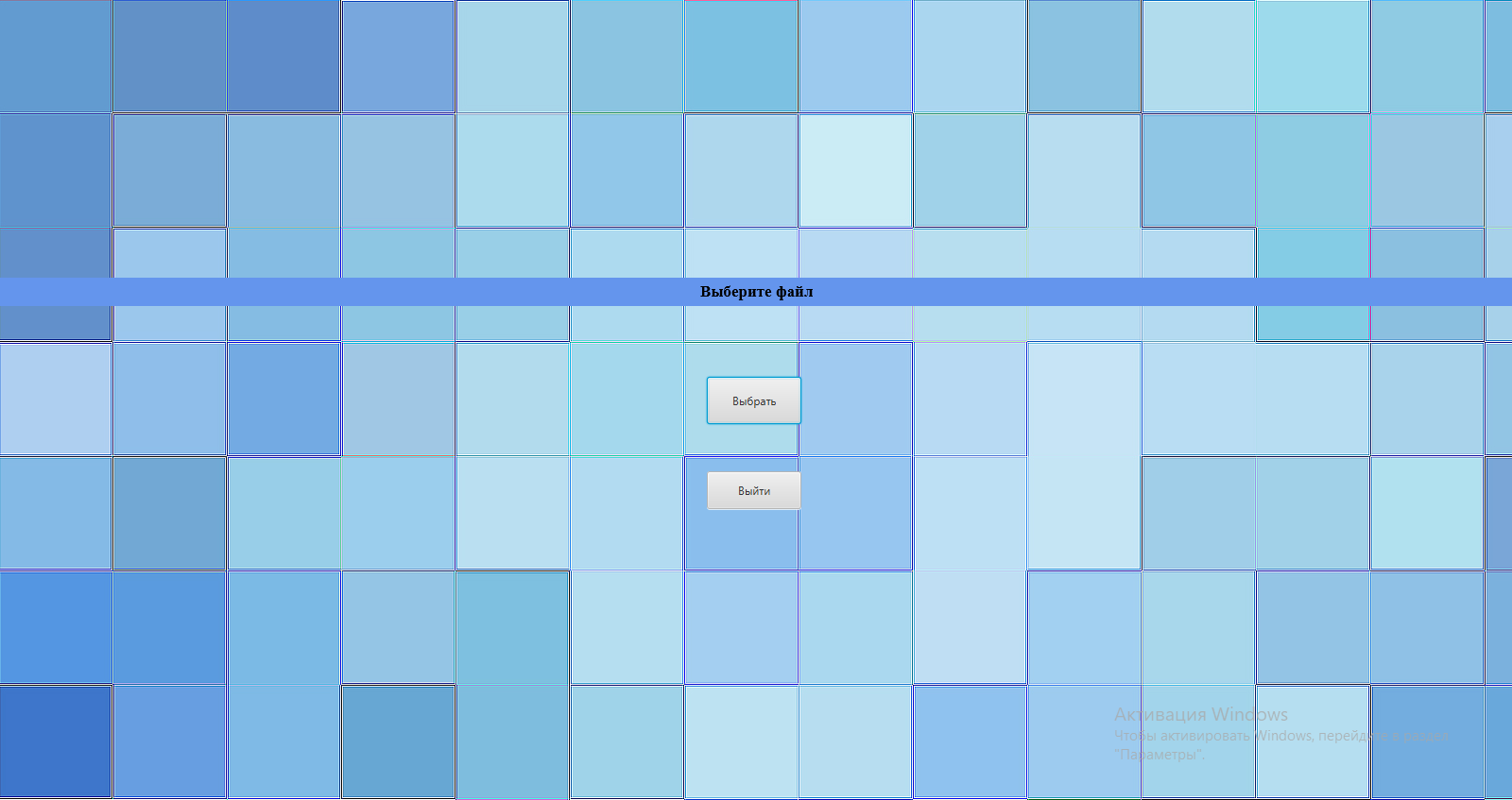


Рис. 5 Экран выбора файла со словами

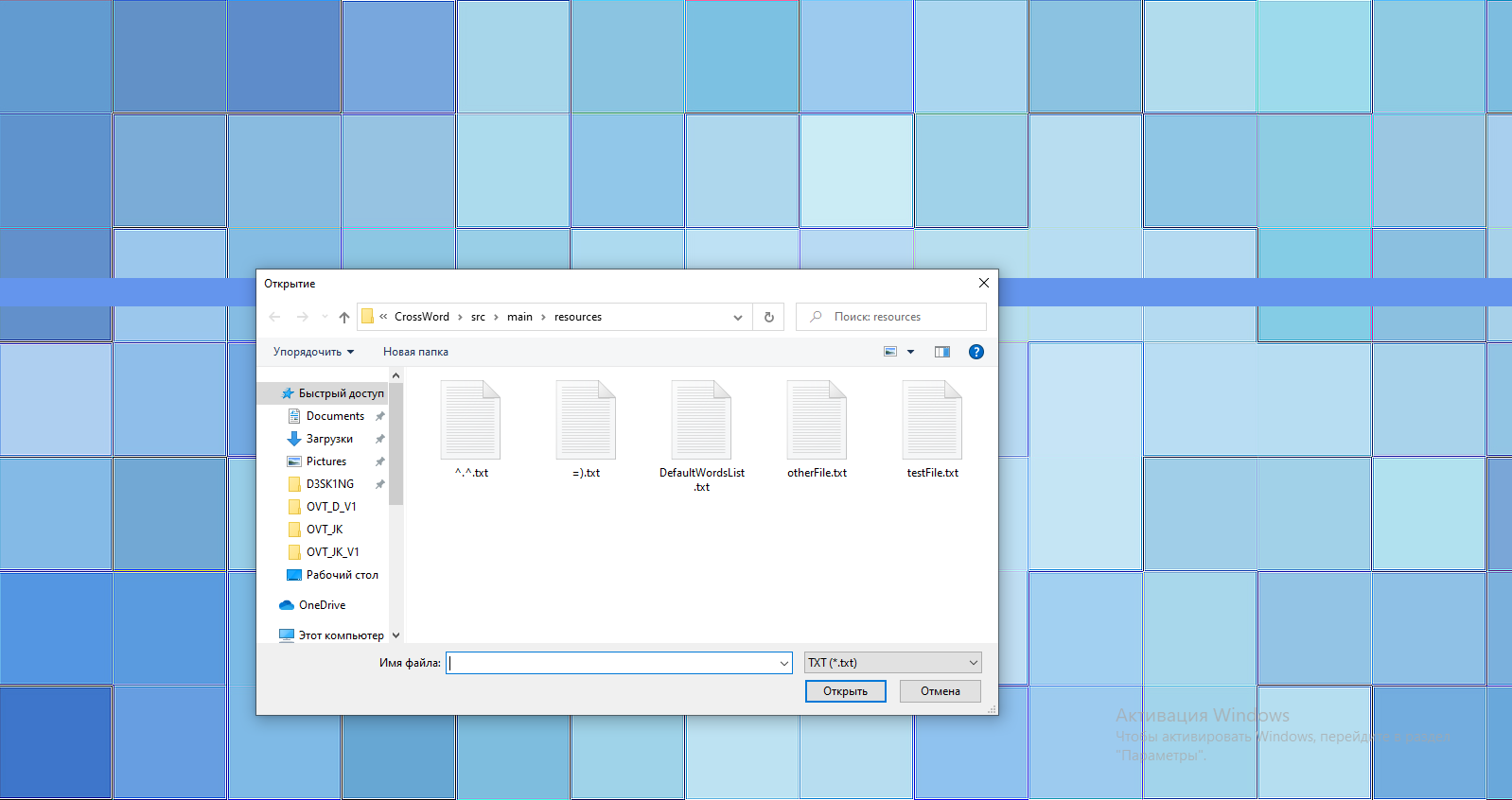


Рис. 6 Диалоговое окно выбора файла



Рис. 7 Отображение Выбранного файла

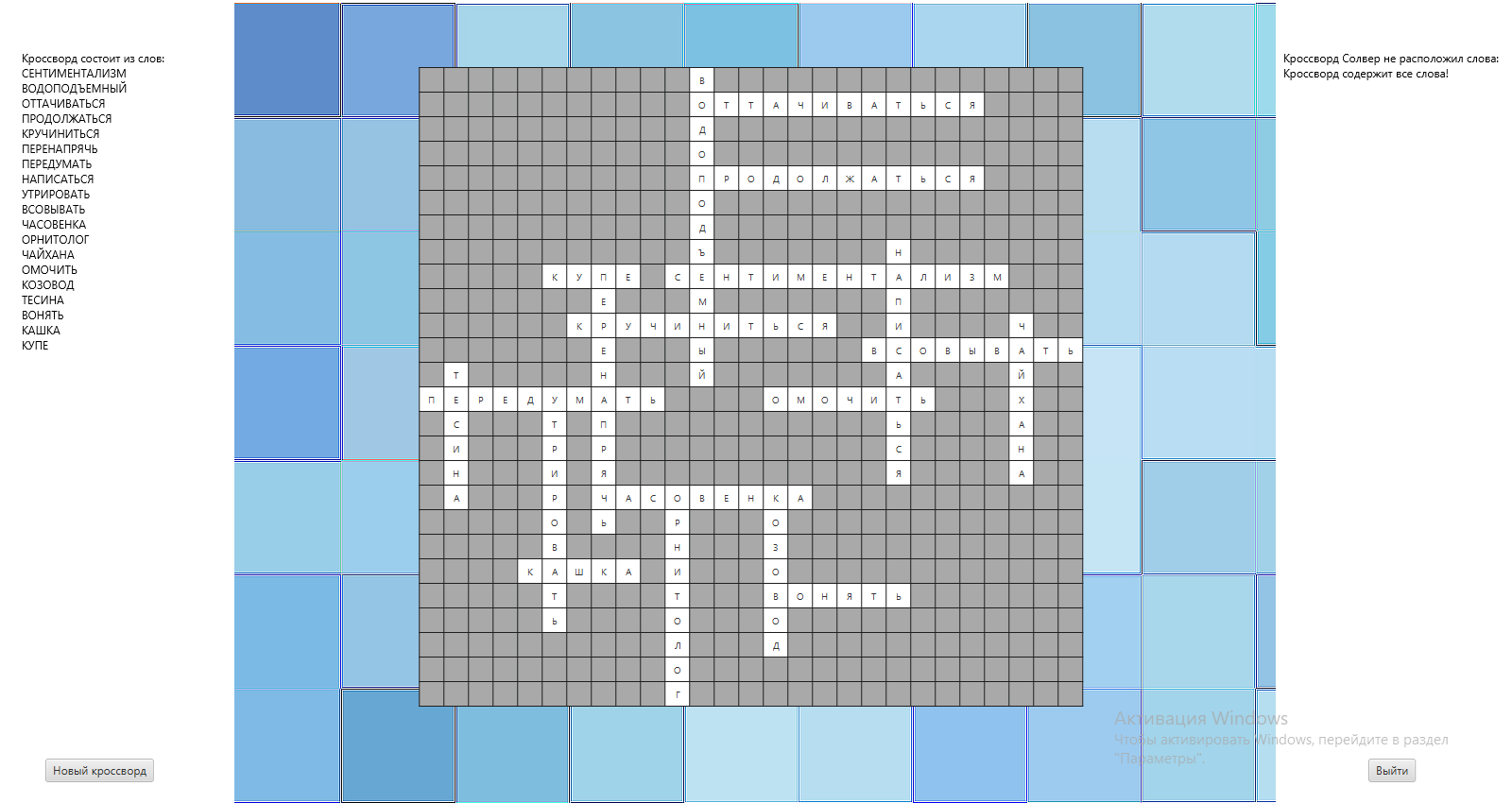


Рис. 8 Экран построенного кроссворда

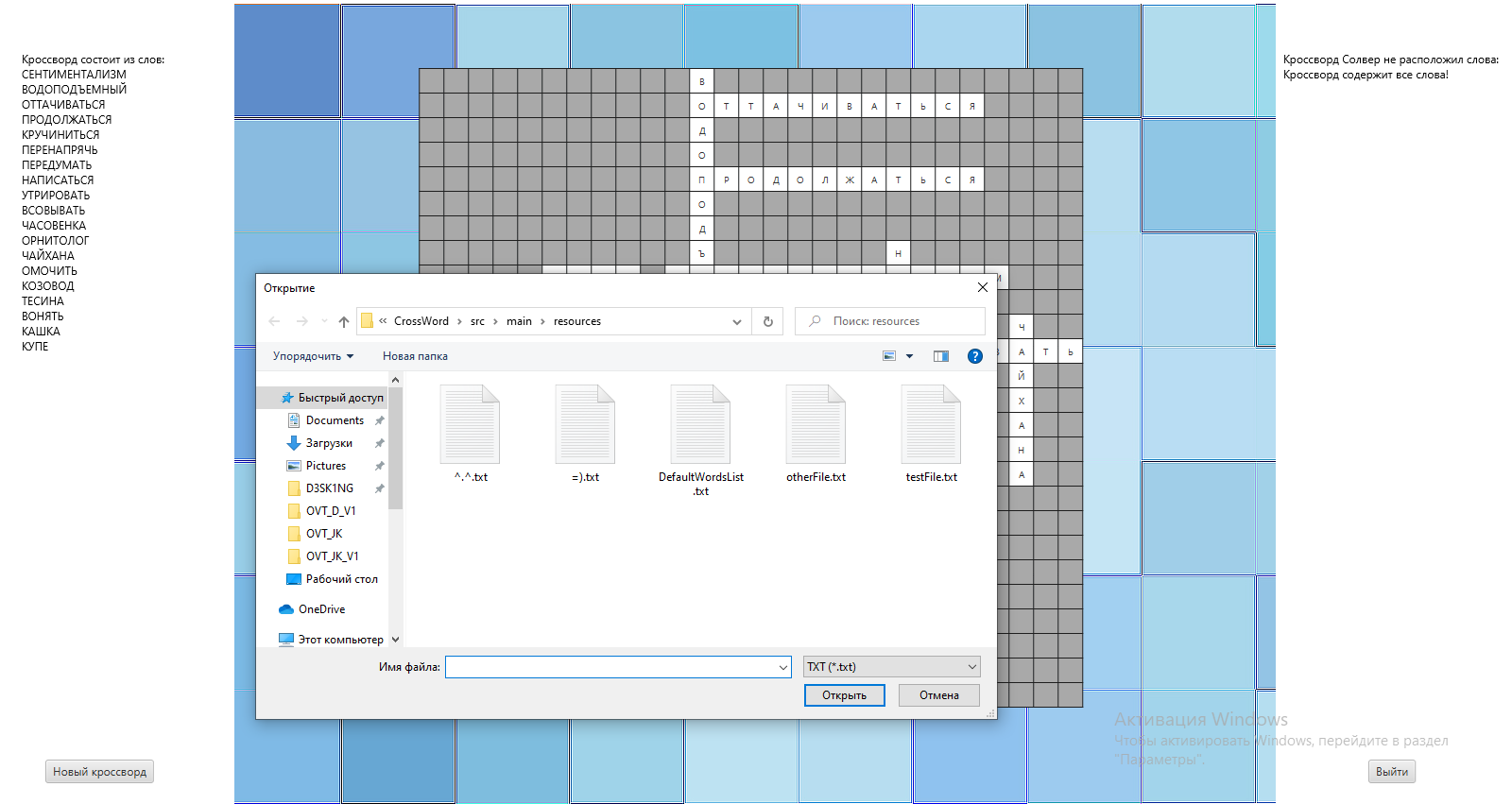


Рис. 9 Диалоговое окно выбора нового файла для построения

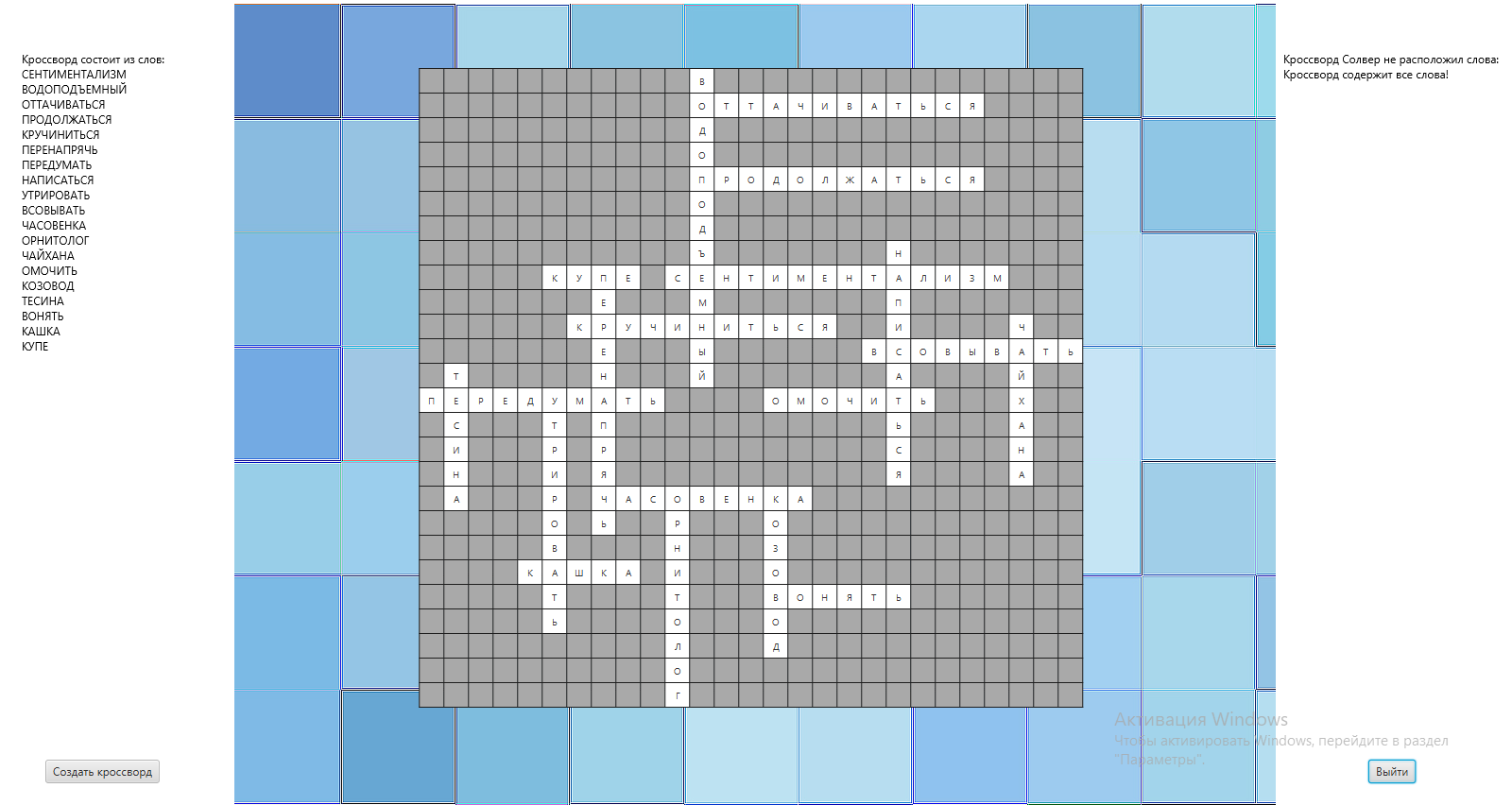


Рис. 10 Новый кроссворд 1

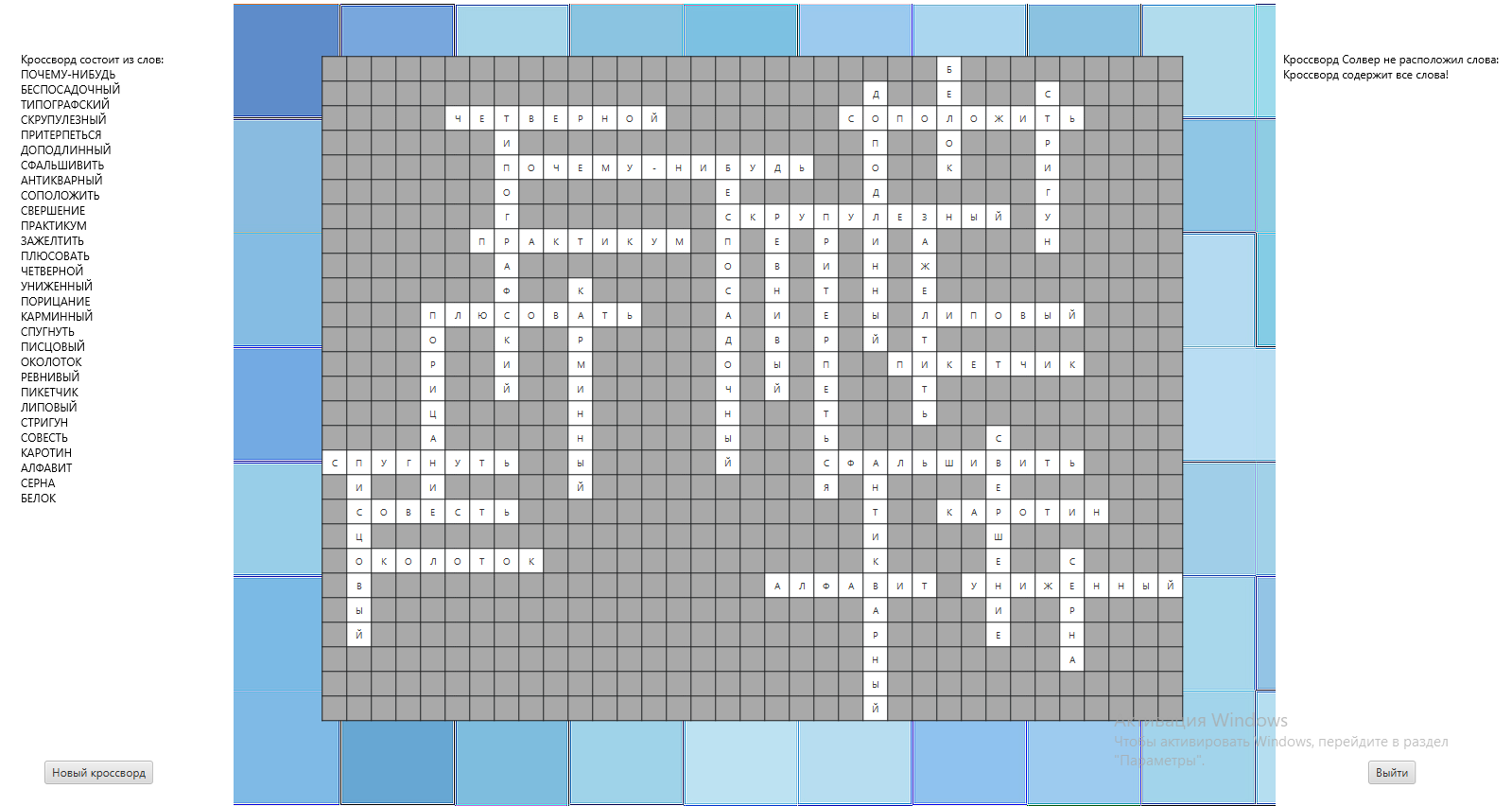


Рис. 11 Новый кроссворд 2

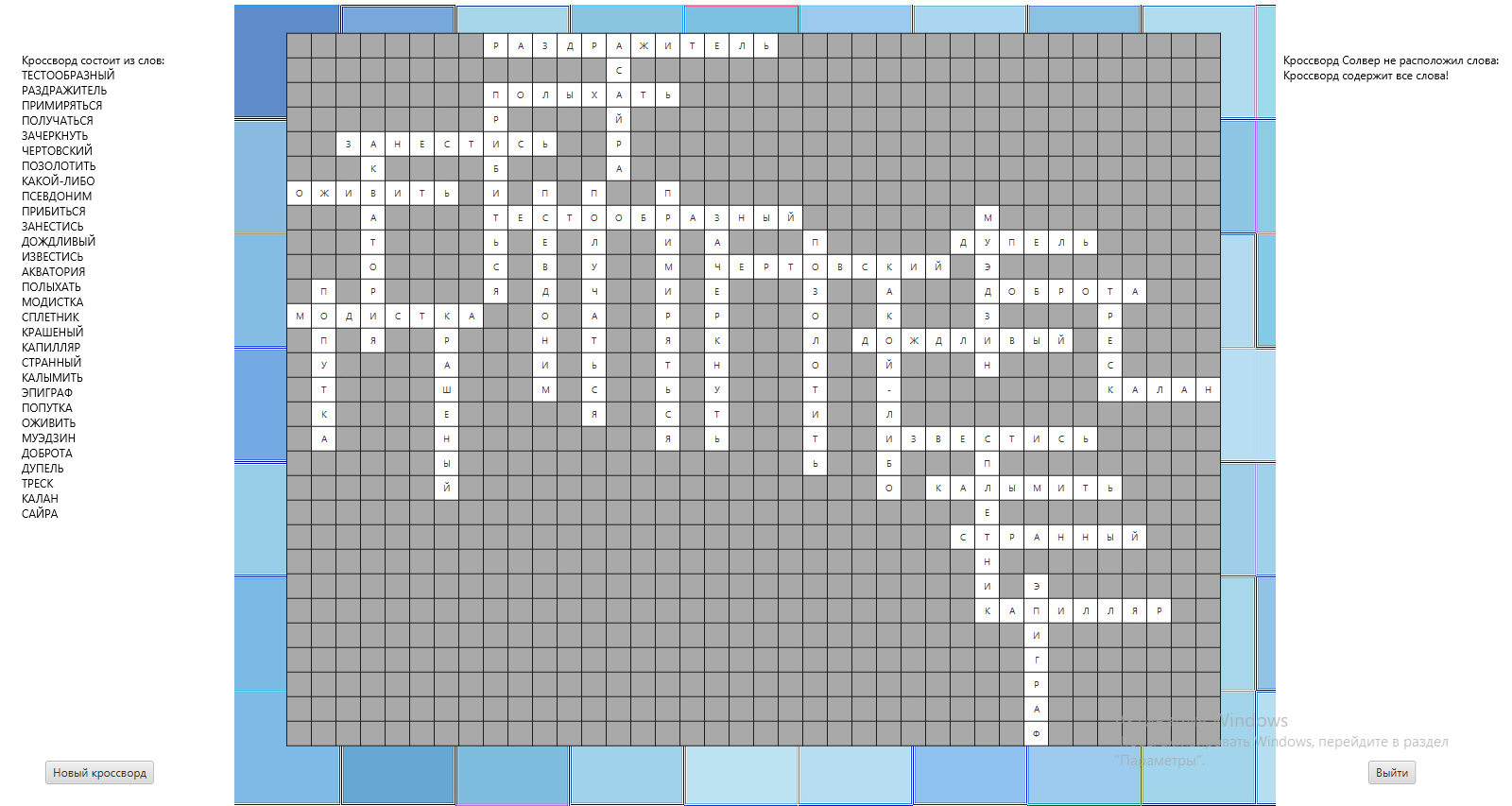


Рис. 11 Новый кроссворд 3