Министерство образования и науки Российской Федерации  
  
федеральное Государственное АВТОНОМНОЕ образовательное учреждение высшего образования  
«Самарский государственный аэрокосмический  
университет имени академика С. П. Королева  
(национальный исследовательский университет)» (СГАУ)

Факультет Информатики   
Кафедра Программных систем

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

по теме «Автоматизированная система

составления и разгадывания классического кроссворда»

Студенты Блюдина К.А.

Бугаков В.В.

Саяпин А.М.

Стефанюк В.В.

Руководитель Зеленко Л.С.

Самара 2014 г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ   
«САМАРСКИЙ ГОСУДАРСТВЕННЫЙ АЭРОКОСМИЧЕСКИЙ УНИВЕРСИТЕТ   
ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА   
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)» (СГАУ)

Факультет информатики

Кафедра программных систем

ЗАДАНИЕ

на курсовой проект по дисциплине

«Программная инженерия»

студентам группы № 6412 Б 300  
Бугакову В.В.  
Саяпину А.М.  
Стефанюк В.В.  
Блюдиной К.А.

1. **Тема проекта:** «Автоматизированная система составления и разгадывания классического кроссворда»
2. **Исходные данные к проекту:** см. приложение к заданию
3. **Перечень вопросов, подлежащих разработке в курсовом проекте:**
   1. Произвести анализ предметной области: изучить основные принципы составления кроссвордов, изучить методы и алгоритмы генерации кроссвордов
   2. Выполнить обзор существующих систем-аналогов
   3. Разработать информационно-логический проект по методологии UML
   4. Разработать и реализовать программное и информационное обеспечение, провести тестирование и отладку .
   5. Оформить документацию курсового проекта
   6. Подготовить презентацию по разработанной системе
4. **Перечень графических разработок**
   1. Диаграммы UML
   2. Схемы основных алгоритмов
5. **Календарный план выполнения работ**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | Содержание работы по этапам | Объем этапа в % к общему объему проекта | Срок  окончания | Фактическое выполнение |
| 1 | Оформление технического задания и его утверждение | 5 | 11.09.14 |  |
| 2 | Анализ и описание предметной области | 10 | 25.09.14 |  |
| 3 | Проектирование системы | 30 | 20.10.14 |  |
| 3.1 | Разработка структурной схемы системы | 5 | 29.09.14 |  |
| 3.2 | Разработка функциональной спецификации  системы | 10 | 06.10.14 |  |
| 3.3 | Разработка информационно-логического проекта системы и его предъявление руководителю | 15 | 20.10.14 |  |
| 4 | Реализация проекта, разработка контрольных примеров. Предъявление реализации руководителю | 45 | 01.12.14 |  |
| 5 | Корректировка проекта и оформление документации проекта. Защита проекта с представлением презентации. | 10 | 18.12.14 |  |

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_ / Бугаков В.В. / \_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_ / Саяпин А.М. / \_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_ / Стефанюк В.В. / \_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_ / Блюдина К.А. / \_\_\_\_\_\_\_\_\_\_\_

**ПРИЛОЖЕНИЕ  
к заданию на курсовой проект**студентам группы № 6412 Б 300  
Бугакову В.В.  
Саяпину А.М.  
Стефанюк В.В.  
Блюдиной К.А.

Тема проекта: **«Автоматизированная система составления и разгадывания классического кроссворда»**

**Исходные данные к проекту:**

1. **Характеристика объекта автоматизации:**

## объект автоматизации: классический кроссворд;

## виды автоматизируемой деятельности:

* + процесс составления/генерирования кроссворда;
  + процесс разгадывания кроссворда;
  + процесс работы со словарем понятий;
  + процесс визуализации работы с кроссвордом;

## минимальная длина одного слова – 3 символа;

## максимальная длина одного слова – 15 символов;

1. минимальное значение длины ***–*** 10;
2. максимальное значение длины ***–*** 50;
3. минимальное значение ширины ***–*** 10;
4. максимальное значение ширины ***–*** 50;

## язык записи понятий – русский.

1. **Требования к информационному обеспечению:**
2. информационное обеспечение разрабатывается на основе следующих документов и массивов данных:
   * Описание структуры кроссворда [Электронный ресурс] – ru.wikipedia.org/wiki/Кроссворд (дата обращения 09.09.2014 г.);
   * Структуры словаря понятий (понятие и его определение располагаются в одной строке, разделены пробелом);
3. словари понятий хранятся в текстовых файлах формата \*.txt;
4. кроссворды хранятся в файлах, структура файла определяется в процессе проектирования;
5. предусмотреть контроль целостности создаваемого кроссворда.
6. **Требования к техническому обеспечению:**
7. тип ЭВМ – IBM PC совместимый;
8. монитор с разрешающей способностью не ниже 800 х 600;
9. манипулятор – мышь;
10. технические характеристики определяются в процессе выполнения курсового проекта.
11. **Требования к программному обеспечению:**
12. тип операционной системы – Windows 98 и выше, Windows 7, Windows Vista, Mac OS;
13. язык программирования – JavaScript;
14. среда программирования – Webstorm;
15. среда проектирования – StarUML 2.0.0
16. **Общие требования к проектируемой системе:**

***5.1 Функции, реализуемые системой:***

1. настройка параметров кроссворда:
2. выбор оформления;
3. выбор размера поля;
4. загрузка словаря понятий;
5. автоматическое составление кроссворда с проверкой целостности его структуры;
6. ручное составление кроссворда кроссворда:
7. добавление слова;
8. удаление слова;
9. изменение ориентации слова;
10. перемещение слова по сетке;
11. сохранение кроссворда в файл заданной структуры;
12. загрузка кроссворда из файла;
13. работа со словарями понятий:
14. добавление понятия;
15. удаление понятия;
16. изменение понятия;
17. сортировка словаря по выбранному критерию;
18. загрузка словаря из файла;
19. сохранение словаря из файла;
20. создание нового словаря понятий;
21. разгадывание кроссворда с организацией системы подсказок:
22. загрузка кроссворда из файла;
23. вписывание буквы;
24. удаление буквы;
25. визуализация процессов работы с кроссвордом;
26. выдача справочной информации о системе.

***5.2 Технические требования к системе:***

1. режим работы – диалоговый;
2. время автоматической генерации кроссворда не более 1 минуты;
3. система должна удовлетворять санитарным правилам и нормам  
   СанПин 2.2.2./2.4.2198-07;
4. условия работы средств вычислительной техники (содержание вредных веществ, пыли и подвижность воздуха) должны соответствовать ГОСТ 12.1.005, 12.01.007.
5. температура окружающего воздуха – 15-25°С;
6. влажность 45-75%.

Руководитель   
курсового проекта \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Зеленко Л.С. / \_\_\_\_\_\_\_\_\_\_\_\_

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_ / Бугаков В.В. / \_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_ / Саяпин А.М. / \_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_ / Стефанюк В.В. / \_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_ / Блюдина К.А. / \_\_\_\_\_\_\_\_\_\_\_\_

СОДЕРЖАНИЕ

[Введение 10](#_Toc409563661)

[1 Описание предметной области 12](#_Toc409563662)

[1.1 История возникновения кроссворда 12](#_Toc409563663)

[1.2 Классификация кроссвордов 13](#_Toc409563664)

[1.2.1 Классический кроссворд 14](#_Toc409563665)

[1.2.2 Кроссворд скандинавского типа 15](#_Toc409563666)

[1.2.3 Ключворд 16](#_Toc409563667)

[1.3 Обзор существующих систем-аналогов 17](#_Toc409563668)

[1.3.1 Система Cross 17](#_Toc409563669)

[1.3.2 Система Bestcrosswords 18](#_Toc409563670)

[1.4 Анализ предметной области 19](#_Toc409563671)

[1.5 Постановка задачи 20](#_Toc409563672)

[2 Проектирование системы 23](#_Toc409563673)

[2.1 Структурная схема системы 23](#_Toc409563674)

[2.2 Спецификация системы 24](#_Toc409563675)

[2.2.1 Функциональная спецификация 25](#_Toc409563676)

[2.2.2 Перечень исключительных ситуаций 31](#_Toc409563677)

[2.2.3 Спецификация качества 31](#_Toc409563678)

[2.3 Проектирование интерфейса пользователя 32](#_Toc409563679)

[2.4 Информационно-логический проект системы 36](#_Toc409563680)

[2.4.1 Язык UML 36](#_Toc409563681)

[2.4.2 Диаграмма вариантов использования 37](#_Toc409563682)

[2.4.3 Диаграмма классов 47](#_Toc409563683)

[2.4.4 Диаграмма состояний 48](#_Toc409563684)

[2.4.5 Диаграмма деятельности 49](#_Toc409563685)

[2.4.6 Диаграмма последовательности 55](#_Toc409563686)

[2.5 Выбор и обоснование комплекса программных средств 61](#_Toc409563687)

[2.5.1 Выбор языка программирования и среды разработки 61](#_Toc409563688)

[2.5.2 Выбор операционной системы 62](#_Toc409563689)

[3 Реализация системы 64](#_Toc409563690)

[3.1 Разработка и описание интерфейса пользователя 64](#_Toc409563691)

[3.1.1 Разработка и описание пользовательского меню 65](#_Toc409563692)

[3.1.2 Описание контрольного примера 67](#_Toc409563693)

[3.2 Выбор и обоснование комплекса технических средств 68](#_Toc409563694)

[3.2.1 Расчет объема занимаемой памяти 68](#_Toc409563695)

[3.2.2 Минимальные требования, предъявляемые к системе 69](#_Toc409563696)

[3.3 Диаграммы реализации 69](#_Toc409563697)

[3.3.1 Диаграмма компонентов 69](#_Toc409563698)

[3.3.2 Диаграмма развертывания 71](#_Toc409563699)

[Заключение 72](#_Toc409563700)

[Список использованной литературы 73](#_Toc409563701)

[Приложение А Руководство пользователя 75](#_Toc409563702)

[Приложение Б Листинг модулей программы 86](#_Toc409563706)

**РЕФЕРАТ**

Пояснительная записка 109 с, 53 рисунка, 4 таблицы, 18 источников,   
2 приложения.

Графическая документация: 27 слайдов презентации Power Point.

КЛАССИЧЕСКИЙ КРОССВОРДА, СИСТЕМЫ АНАЛОГИ, UML, RAD, СЛОВАРЬ ПОНЯТИЙ, РАЗГАДЫВАНИЕ

Во время курсового проекта разработаны алгоритмы и соответствующая им автоматизированная система составления и разгадывания классического кроссворда. Слова (понятие и его определение) хранятся в текстовом файле и могут дополняться вручную (с использованием текстового редактора) или внутри программы, при этом ограничений на длину словаря не существует. Программа позволяет сформировать кроссворд, учитывая ограничения на параметры. В системе имеется возможность сохранения кроссвордов в файл с целью последующего их разгадывания.

Программа написана на языке JavaScript в среде WebStorm и функционирует в операционной системе Windows’7 и Mac OS. Проектирование велось с использованием среды проектирования Star UML 2.0.0.

# введение

Сейчас существует множество бесплатных программ по созданию кроссвордов, сканвордов, японских кроссвордов. То, на что при ручном труде занимает более суток, на компьютере можно сделать за 10-15 минут.

В настоящее время в каждом городе выходит множество газет, почти каждая из них публикует кроссворды. Но не у каждой редакции, даже при наличии компьютеров, есть возможность их составлять самим.

Индустрия развлечений с каждым годом все больше развивается, требуя привлечения новых технологий, расширения списка возможностей существующих систем. В связи с этим появляются все более новые и совершенные системы автоматизированного составления кроссвордов различной сложности с большим количеством понятий. Рассмотрим основные плюсы современных электронных кроссвордов: во-первых, это оперативность, во-вторых – удобство. Дело в том, что в Интернете гораздо удобнее разгадывать кроссворды, чем на страницах печатных изданий. Буквы вписываются с клавиатуры, что, естественно, более благоприятно для визуального восприятия по сравнению с ручным написанием, в случае ошибки неправильно вписанную букву легко удалить, в подавляющей массе онлайн-кроссвордов можно легко воспользоваться подсказкой, которая покажет вам неизвестное слово [1].

Во время выполнения курсового проекта необходимо разработать автоматизированную систему составления и разгадывания классического кроссворда, которая позволит составлять кроссворд, разгадывать его, а также работать со словарем понятий.

Во время разработки система будет использовать технологию RAD (Rapid Application Development – быстрая разработка приложений), которая служит для создания средств разработки программных Особое внимание уделяется быстроте и удобству программирования, созданию технологического процесса, позволяющего программисту максимально быстро создавать компьютерные программы. RAD – это жизненный цикл процесса проектирования, созданный для достижения более высокой скорости разработки и качества ПО, чем это возможно при традиционном подходе к проектированию [2].

Во время проектирования будет использоваться методология UML (Unified Modeling Language – унифицированный язык моделирования), которая служит для графического описания объектного моделирования в области разработки программного обеспечения. UML является языком широкого профиля, это – открытый стандарт, использующий графические обозначения для создания абстрактной модели системы, называемой UML-моделью. UML был создан для определения, визуализации, проектирования и документирования, в основном, программных систем. UML не является языком программирования, но на основании UML-моделей возможна генерация кода [3].

# 1 описание предметной области

Кроссворд (англ. Crossword – пересечение слов) – «крестословица», «плетенки», «пирамиды», «дорожки», «магические квадраты»; головоломка, представляющая собой переплетение рядов клеточек, которые заполняются словами по заданным значениям. Обычно значения слов задаются описательно под этой фигурой, сначала значения слов, которые должны получиться по горизонтали, затем - по вертикали [4].

## 1.1 История возникновения кроссворда

Исследователям встречались находки, похожие на кроссворд, датированные еще 1-4 вв. н.э. В частности, во время раскопок, производимых в Помпеях, была обнаружена головоломка, удивительно напоминающая современный кроссворд, которую ученые датировали 79 годом н.э. При этом существуют различные версии изобретения кроссвордов. Среди стран, претендующих на звание родины кроссвордов, Италия, Великобритания, Соединенные Штаты Америки.

По одной из версий, прототипы современных кроссвордов появились ещё в XIX веке. Самый первый дошедший до нас кроссворд был опубликован в 1875 году в сентябрьском номере журнала «Святой Николас» в Нью-Йорке. При этом первый кроссворд, соответствующий современным представлениям о кроссворде, был создан Артуром Уинном и опубликован в воскресном номере газеты «New York World» 21 декабря 1913 года. Кроссворды стали популярны в середине 1920-х годов.

В приложении «Наш мир» к берлинской газете «Руль» в феврале 1925 года впервые употребляется термин «крестословица», придуманный писателем Владимиром Набоковым для кроссвордов, которые он составлял для этого издания.

Самый первый в России кроссворд был напечатан в Ленинграде 2 декабря 1925 года в литературном журнале «Резец» № 48.

Широкую популярность приобрели кроссворды, публиковавшиеся в течение многих десятилетий в журнале «Огонек».

В постсоветское время в центральных газетах появились «фирменные» авторские кроссворды (например, призовые кроссворды Виктора Боборико в газете "Поле чудес" или традиционный «кроссворд от Олега Васильева» в «КП»).

В конце 1990-х появились специализированные «кроссвордные» газеты. Кроме классических кроссвордов и их разновидностей, упомянутых ниже, в печати появились и стали весьма популярны «скандинавские» кроссворды, а также цифровые головоломки. Сначала «рисование по числам», переименованное в «японский кроссворд», затем «судоку», «какуро» и многочисленные их вариации.

На 2013 год зарегистрировано более 400 печатных изданий, публикующих кроссворды и головоломки (как словесные, так и цифровые) различной степени сложности.

Кроссворд продолжает развиваться как по форме, так и по содержанию. Существует множество разновидностей этой игры. В разных странах есть свои любимые варианты кроссворда, причем они могут использоваться не только как полезное развлечение, но и в учебных целях. Во многих странах проводятся конкурсы по решению и составлению кроссвордов, существуют клубы любителей кроссвордов (в России - Международный клуб русских кроссвордов «Крестословица» в Ст.-Петербурге).

## 1.2 Классификация кроссвордов

Существует несколько десятков разновидностей и модификаций кроссвордов. Кроме того, существуют "кроссворды-гибриды", которые сочетают в себе качества двух и более групп. Например, реверсивный сканворд.

### 1.2.1 Классический кроссворд

Разгадывание слов производится по данным определениям. Определения для загаданных слов кроссворда идут отдельно от сетки, а ответы на эти определения вписываются в сетку кроссворда по горизонтали и по вертикали, при этом так же используется нумерация слов. Пример классический кроссворда изображен на рисунке 1.

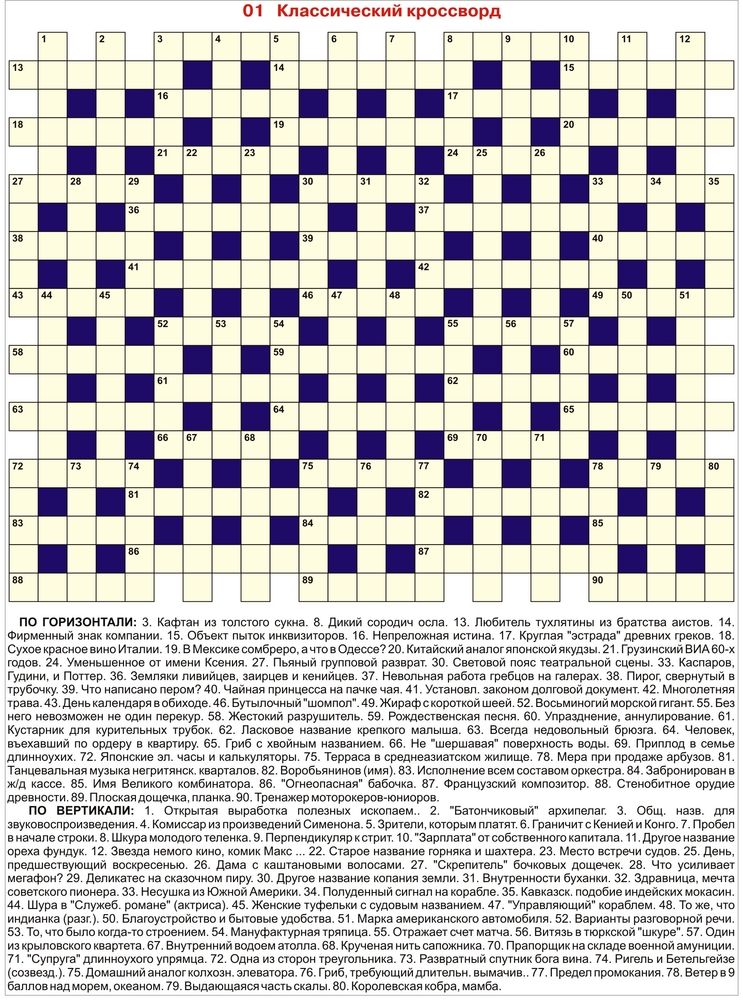
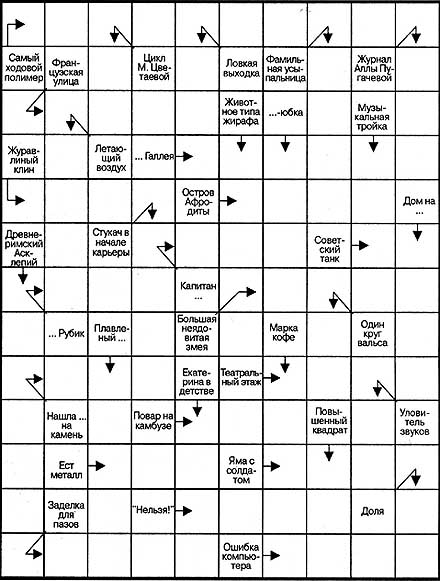


Рисунок 1 – Пример классического кроссворда

Сетка кроссворда может быть довольно плотной, но зачастую по плотности она уступает сетке сканвордов. В отличии, например, от сканвордов, сетка кроссворда может быть симметричной. В кроссворде в качестве вопросов могут использоваться фотографии или картинки вписанные в сетку кроссворда. Вопросы у кроссворда могут быть длинными, так как они не вписаны в сетку кроссворда и тем самым их длина ничем не ограничена.

### Кроссворд скандинавского типа

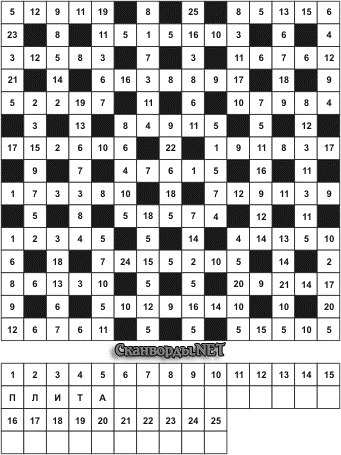
В сетке размещены короткие определения. В качестве определения в сканворде могут использоваться картинки или фото, что придаёт сканворду красивый внешний вид и делает его более интересным и увлекательным при разгадывании. Пример кроссворда скандинавского типа изображен на рисунке 2.

  
Рисунок 2 – Пример кроссворда скандинавского типа

Зачастую слова в сетке сканворда имеют довольно высокую плотность пересечения, что позволяет узнать слово, ответ, на определение которого неизвестен разгадывающему. Таким образом, можно полностью разгадать сканворд и при этом не знать всех ответов на загаданные в сканворде определения.

### 1.2.3 Ключворд

В ключворде каждая буква имеет свой определённый числовой код (ключ) и соответственно заменена этим числом, при этом одинаковым буквам соответствуют одинаковые числа. Кейворд (ключворд) - это своего рода буквенно-числовой кроссворд, в котором нет определений. Пример ключворда изображен на рисунке 3.

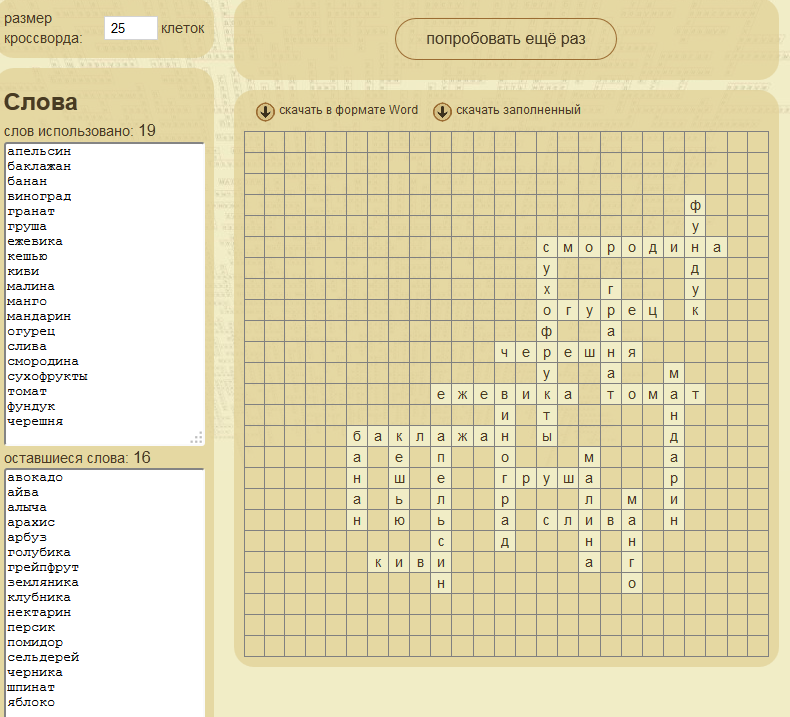
  
Рисунок 3 – Пример ключворда

Изначально сетка ключворда заполнена числами. Таким образом, для разгадывания ключворда необходимо определить какое число какой букве соответствует. Под сеткой ключворда стоит сетка с числами используемыми в ключворде, а под ними пустые ячейки для вписывания определившихся букв. В качестве подсказки в этой сетке есть несколько уже известных букв которым соответствуют определённые числа.

## 1.3 Обзор существующих систем-аналогов

### 1.3.1 Система Cross

Сross.highcat.org ***–*** сайт по составлению кроссвордов только из заданных слов. Сетка слов выстраивается таким образом, чтобы в кроссворд попало как можно больше слов. Создатели Дмитрий Жигулин и Александр Сычков [5]. Пример кроссворда, созданного в системе cross, изображен на рисунке 4.

  
Рисунок 4 – Пример классического кроссворда в автоматизированной системе составления кроссвордов Cross

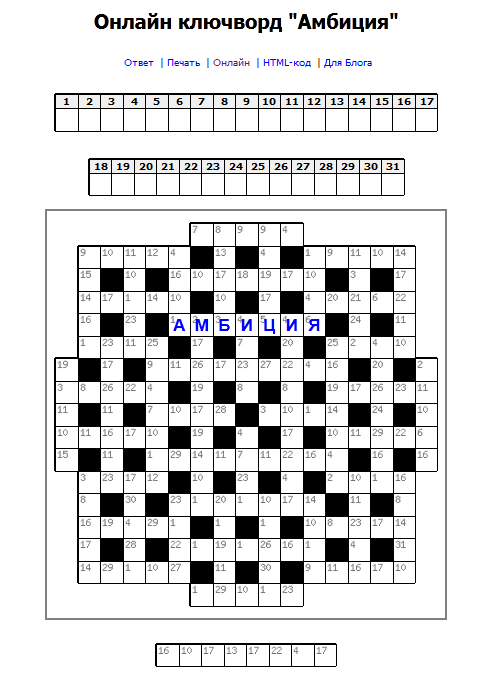
Изначально пользователь задает размер кроссворда в клетках. Система заполняет его словами, хранящимися в словаре, и строит из них кроссворд максимального размера. Система cross производит только составление кроссворда без списка вопросов и возможности его разгадывания. Есть возможность скачать составленный пустой и заполненный кроссворд в формате \*.docx.

### 1.3.2 Система Bestcrosswords

Bestcrosswords.ru ***–*** это самые разнообразные кроссворды, включая японские, скандинавские (сканворды) и ключворды. Кроссворды представлены в двух вариантах: печатный и онлайн. Версия для печати представляет собой статическую картинку с сеткой кроссворда и определениями. Онлайн вариант кроссворда реализован таким образом, что позволит решать кроссворд прямо на странице сайта. При этом доступны операции по сохранению решения пользователя и другие.

Все кроссворды собраны в каталоге, который разделен по типам кроссвордов: классические кроссворды, японские кроссворды, цветные японские кроссворды, скандинавские кроссворды и ключворды. Кроссворды в каждом из разделов могут быть отсортированы по различным критериям в прямом или обратном порядке. В каталоге представлены работы известных составителей кроссвордов, таких как Олег Васильев и других, а также собственные работы [6]. Пример классического кроссворда в системе Bestcrosswords изображен на рисунке 5, пример ключворда изображен на рисунке 7.

  
Рисунок 5 – Пример классический кроссворда в системе Bestcrosswords

  
Рисунок 6 – Пример ключворда в системе Bestcrosswords

## 1.4 Анализ предметной области

Диаграмма объектов представляет статическую составляющую взаимодействующих между собой объектов, она должна включить в себя только те объекты предметной области, которые потом преобразуются в диаграмму классов. Связи между объектами показывают отношения между ними, при необходимости в диаграмме можно привести и атрибуты(свойства) объектов [7].

Классический кроссворд представляет собой совокупность двух составляющих: сетки, на которой будут располагаться слова, и задания, которое по существу представляет собой набор определений, которые разъясняют смысл слов (понятий). Набор понятий и определений хранятся во внешнем файле заданного формата ***–*** словаре понятий. На рисунке 7 изображена диаграмма объектов предметной области, изображающая работу автоматической системы составления классического кроссворда.

Файл  
КК

Состоит из

Классический кроссворд

Сетка

Задание

Состоит из

Слово

Располагается на

Определение

Разъясняет смысл

Хранится в

Словарь понятий

Хранится в

Рисунок 7 – Диаграмма объектов предметной области

## 1.5 Постановка задачи

Перед авторами поставлена задача разработать автоматизированную систему составления и разгадывания классического кроссворда по технологии RAD и методологии UML. Данная система позволит осуществлять составление кроссворда, его разгадывание и работу со словарем понятий.

Для достижения поставленной цели необходимо решить следующие задачи:

1. *Составление кроссворда* должно проводиться в двух режимах: автоматическом (генерирование) и ручном. В любом случае пользователь должен предварительно задать параметры кроссворда: определить его максимальную длину и ширину (не менее 10 и не более 50 символов), выбор оформления, подключить словарь понятий (он должен храниться во внешнем текстовом файле заранее заданной структуры). При этом система должна провести проверку правильности этой структуры и, в случае несоответствия, выдать предупредительное сообщение и обеспечить повторный ввод параметров. В системе должен осуществляться контроль типов и диапазонов значений параметров. В режиме ручного составления кроссворда (или его редактирования) пользователю должны предоставляться следующие возможности: удаление слова, добавление нового слова, изменение ориентации слова и перемещение слова по сетке. В случае необходимости пользователь должен иметь возможность сохранить полученный кроссворд в файл (с целью дальнейшего его разгадывания или редактирования), структура файла должна быть определена в ходе проектирования. Необходимо также предусмотреть контроль целостности создаваемого кроссворда (отсутствие одиночных слов без пересечений).

2. *Разгадывание кроссворда* должно производиться побуквенно, удаление уже существующего понятия – целиком. В кроссворде должны использоваться только строчные буквы русского алфавита.

3. *Работа со словарями понятий* должна предоставляет пользователю следующие возможности: добавление понятия, удаление понятия, изменение понятия, сортировка словаря по выбранному критерию, загрузка словаря из файла, сохранение словаря из файла, создание нового словаря понятий. Понятие и его определение должны располагаются в одной строке и разделяться пробелом. Словари понятий должны храниться в текстовых файлах формата \*.dic.

В системе также должна быть обеспечена возможность получения справочной информации как о самой системе, так и предоставляемых ею возможностях.

Таким образом, система должна выполнять следующие функции:

1. настройка параметров кроссворда:
2. выбор оформления;
3. выбор размера поля;
4. загрузка словаря понятий;
5. автоматическое составление кроссворда с проверкой целостности его структуры;
6. ручное составление кроссворда кроссворда:
7. добавление слова;
8. удаление слова;
9. изменение ориентации слова;
10. перемещение слова по сетке;
11. сохранение кроссворда в файл заданной структуры;
12. загрузка кроссворда из файла;
13. работа со словарями понятий:
14. добавление понятия;
15. удаление понятия;
16. изменение понятия;
17. сортировка словаря по выбранному критерию;
18. загрузка словаря из файла;
19. сохранение словаря из файла;
20. создание нового словаря понятий;
21. разгадывание кроссворда:
22. загрузка кроссворда из файла;
23. вписывание буквы;
24. удаление буквы;
25. визуализация процессов работы с кроссвордом;
26. выдача справочной информации о системе.

# проектирование системы

Система – множество элементов, находящихся в отношениях и связях друг с другом, которое образует определённую целостность, единство. Отличительным свойством системы является ее целостность: комплекс объектов, рассматриваемых в качестве системы, должен обладать общими свойствами и поведением [8].

## 2.1 Структурная схема системы

Система должна представлять собой совокупность элементов, находящихся между собой в определенной зависимости и составляющих некоторое единство, направленное на достижение определенной цели. Система может являться элементом другой системы более высокого порядка и включать в себя системы более низкого порядка. То есть систему можно рассматривать как набор подсистем, организованных для достижения определенной цели и описанных с помощью набора моделей, а подсистему – как группу элементов, часть которых составляет спецификацию поведения, представленного другими ее составляющими. Структурная схема разрабатываемой системы и описание подсистем изображено на рисунке 8.

Рисунок 8 – Структурная схема системы

Подсистема управления

Справочная подсистема

Подсистема разгадывания кроссворда

Файловая подсистема

Подсистема визуализации

Подсистема составления кроссворда

Подсистема настройки параметров

Подсистема ручного составления

Подсистема генерирова  
ния

Подсистема работы со словарем

В состав системы входят следующие подсистемы:

1. подсистема управления, которая отвечает за взаимодействие подсистем между собой и представлена в виде иерархического меню;
2. подсистема составления кроссворда, в состав которой входят:
   * подсистема настройки параметров, которая отвечает за ввод (выбор) значений параметров кроссворда и проверку корректности этих значений;
   * подсистема ручного составления, которая отвечает за ручной ввод слова, его удаление, изменение ориентации выбранного слова и его перемещение по сетке;
   * подсистема генерирования, которая отвечает за автоматическое составление кроссворда из максимального числа слов, хранящихся в словаре понятий;
3. подсистема разгадывания, которая отвечает за загрузку кроссворда из файла определенного типа, вписывание букв и их удаление;
4. файловая подсистема, которая отвечает за хранение кроссворда в файле заранее определенного типа;
5. подсистема работы со словарем, которая отвечает за взаимодействие пользователя с файлами словаря: добавление новых понятий, удаление и изменение существующих понятий, сортировка словаря по выбранному критерию, загрузку словаря из файла и создание нового словаря понятий;
6. подсистема визуализации, которая отвечает за корректное отображение кроссворда у пользователя и дает возможность выбора оформления;
7. справочная подсистема, которая содержит сведения о системе (руководство пользователю) и ее об ее разработчиках.

## 2.2 Спецификация системы

Требования – это свойства, которыми должно обладать ПО для адекватного определения функций, условий и ограничений выполнения ПО, а также объемов данных, технического обеспечения и среды функционирования [9].

Программные требования (Software Requirements) – свойства программного обеспечения, которые должны быть надлежащим образом представлены в нём для решения конкретных практических задач. Данная область знаний касается вопросов извлечения (сбора), анализа, специфицирования и утверждения требований к разрабатываемой ПС.

Спецификация требований к ПО (SRS) – процесс формализованного описания функциональных и нефункциональных требований, требований к характеристикам качества в соответствии со стандартом качества ISO/IEC 9126-94, которые будут отрабатываться на этапах жизненного цикла ПО. В спецификации требований отражается:

− структура ПО;

− требования к функциям, качеству и документации;

− задается в общих чертах архитектура системы и ПО, алгоритмы, логика управления и структуры данных.

### 2.2.1 Функциональная спецификация

Функциональные требования задают то, что система должна делать; нефункциональные – с соблюдением «каких условий» (например, скорость отклика при выполнении заданной операции). При разработке этих требований в первую очередь необходимо учитывать потребности пользователя (заказчика). Пользовательские требования (User Requirements) – описывают цели/задачи пользователей системы, которые должны достигаться/выполняться пользователями при помощи создаваемой программной системы. Часто пользовательские требования представляют в виде сценариев (вариантов использования) [7].

Функциональная спецификация должна в полном объёме отображать информационные связи проектируемой системы как с внешним миром, так и между подсистемами. Перечень функций, выполняемых автоматизированной системой составления и разгадывания классического кроссворда содержится в таблице 1.

Таблица 1 – Перечень функций, выполняемых системой

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название  подсистемы | Название функции | Информационная среда | | | |
| Входные данные | | Выходные данные | |
| Назначение (наименование) | Тип, ограничения | Назначение (наименование) | Тип, ограничения |
|  |  |  |  |  |  |
| 1 Справочная | 1.1 Выдать сведения о разработчиках | Сведения о разработчиках системы (ФИО, номер группы) | Текст | Визуальное отображение информации | ‑ |
| 1.2 Выдать сведения о системе | Файл справки | Текстовый (\*.html) |
| Код ошибки | Целое |
| 2 Настройки параметров | 2.1 Загрузить словарь | Имя файла | Строка,\*.dict | Список понятий | Динамический массив строк |
| Код ошибки | Целое |
| 2.2 Задать/изменить высоту кроссворда | Диапазон значений  минимальное максимальное | Целое  10  50 | Сетка заданного размера | Объект класса “Кроссворд” |
| 2.3 Задать/изменить ширину кроссворда | Диапазон значений  минимальное максимальное | Целое  10  50 |
| 2.4 Задать/изменить высоту и ширину кроссворда | Диапазон значений  минимальное максимальное | Целые  10х10  50х50 |

Продолжение таблицы 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 3 Ручного составления | 3.1 Взять и перетащить понятие | Понятие | Строка | Кроссворд | Объект класса “Кроссворд” |
| 3.2 Сменить ориентацию | Понятие | Строка | Текущая ориентация | Логический (инверсия) |
| Текущая ориентация | Логический |
| 3.3 Выделить слово | Список понятий | Динамический массив строк | Номер текущего понятия | Целое |
| 3.4 Добавить слово на сетку | Координаты слова | Целое/целое | Кроссворд | Объект класса “Кроссворд” |
| Длина слова | Целое |
| Кроссворд | Объект класса “Кроссворд” |
| 4 Генерирования | 4.1 Сгенерировать кроссворд | Список понятий | Динамический массив строк | Кроссворд | Объект класса “Кроссворд” |
| Текущие размеры по горизонтали и вертикали | Целое/целое |
| 5 Файловая | 5.1 Загрузить файл с кроссвордом | Имя файла | Строка, \*.puz | Кроссворд | Объект класса “Кроссворд” |
| Код ошибки | Целое |

Продолжение таблицы 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 5 Файловая | 5.2 Сохранить в файл кроссворд | crossword№ | Строка.puz | Файл | Бинарный |
| Кроссворд | Объект класса “Кроссворд” |
| 5.3 Загрузить словарь из файла | Имя файла | Строка, \*.dict | Словарь понятий | Динамический массив строк |
| Код ошибки | Целое |
| 5.4 Сохранить словарь понятий в файл | dictionary№. | Строка, \*.dict | Файл | \*.dict |
| Словарь понятий | Динамический массив строк |
| 5.5 Создать новый словарь понятий | dictionary№. | Строка.dict | Словарь понятий | Динамический массив строк |
| 6 Визуализации | 6.1 Выбрать/сменить оформление | Текущее оформление | Перечисление (светлое/темное) | Текущее оформление | Перечисление (инверсия текущего) |
| 6.2 Отобразить текущее состояние кроссворда | Кроссворд | Объект класса “Кроссворд” | Визуальное отображение кроссворда | - |
| Текущее оформление | Перечисление (светлое/темное) |

Продолжение таблицы 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 6 Визуализации | 6.3 Подсветить неверно введенное слово | Кроссворд | Объект класса “Кроссворд” | Слово на сетке кроссворда с соответствующей «подсветкой» клеток | - |
|  | Координаты слова | Целое/целое |
| 7 Работы со словарем | 7.1 Добавить/изменить новое понятие и определение | Понятие | Строка | Список понятий и их определений | Динамический массив строк |
| Определение | Строка |
| 7.2 Ввести определение | Набор допустимых символов | Русский алфавит | Определение | Строка |
| 7.3 Ввести понятие | Понятие | Строка |
| 7.4 Выбрать понятие из списка | Список понятий | Динамический массив строк | Номер | Целое в пределах размера словаря |
| 7.5 Удалить существующее понятие | Номер | Целое | Список понятий и их определений | Динамический массив строк |
| 7.6 Изменить существующее понятие | Понятие | Строка | Понятие | Строка |
| 7.7 Изменить существующее определение | Определение | Строка | Определение | Строка |
| 7.8 Отсортировать словарь по выбранному критерию | Параметр сортировки | По алфавиту | Список понятий и их определений | Динамический массив строк |

Продолжение таблицы 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 7 Работы со словарем | 7.9 Выбрать критерий сортировки | Список критериев | По длине  По алфавиту | Текущий критерий | Перечислимый |
| 8 Разгадывания кроссворда | 8.1 Вписать букву в клетку кроссворда | Буква | Символ | Кроссворд | Объект класса “Кроссворд |
| Координаты клетки | Целое | Код ошибки | Целое |
| 8.2 Удалить букву из кроссворда | Кроссворд | Объект класса “Кроссворд | Кроссворд | Объект класса “Кроссворд |
| Координаты клетки | Целое/целое |
| 8.3 Выбрать клетку кроссворда | Кроссворд | Объект класса “Кроссворд | Координаты клетки | Целое/целое |
| 8.4 Проверить слово | Координаты клетки | Целое/целое | Признак правильности проверки | Логический |
| Эталонный кроссворд | Объект класса “Кроссворд |

### 2.2.2 Перечень исключительных ситуаций

Исключительная ситуация – это ситуация, при которой система не может выполнить возложенных на нее функций или которая может привести к денормализации работы системы.

Таблица 2 – Перечень исключительных ситуаций

|  |  |  |
| --- | --- | --- |
| Название  подсистемы | Название исключительной ситуации | Реакция системы |
| 1 Справочная | 1.1 Не возможно открыть файл справки | Выдача сообщения «Файл справки поврежден» |
| 1.2 Не возможно найти файл справки | Выдача сообщения «Отсутствует файл справки» |
| 2 Файловая | 2.1 Попытка открытия файла с несобственным форматом | Выдача сообщения «Файл поврежден или недопустимого формата» |
| 2.2 Файл с заданным именем не существует | Выдача сообщения «Файл отсутствует» |
| 3 Настройки параметров | 3.1 Попытка открытия файла с несобственным форматом | Выдача сообщения «Файл поврежден или недопустимого формата» |
| 4 Работы со словарем | 4.1 Попытка открытия файла с несобственным форматом | Выдача сообщения «Файл поврежден» |
| 5 Разгадывания кроссворда | 5.1 Ввод некорректного символа | Выдача сообщения «Могут использоваться только буквы русского алфавита» |

### 2.2.3 Спецификация качества

Разработка спецификации качества сводится к построению своеобразной модели качества разрабатываемой ПС [11]. В этой модели должен быть перечень всех тех достаточно элементарных свойств, которые требуется обеспечить в разрабатываемом ПС и которые в совокупности образуют приемлемое для пользователя качество ПС. При этом каждое из этих свойств должно быть в достаточной степени конкретизировано с учетом определения требований к разрабатываемому ПС и возможности оценки его наличия у разработанного ПС или необходимой степени обладания им этим ПС.

Для конкретизации качества ПС по каждому из критериев используется стандартизованный набор достаточно простых свойств ПС однозначно интерпретируемых разработчиками.

В спецификации качества перечисляются основные требования по показателям качества:

* описывается уровень надежности ПС;
* формулируются требования по быстродействию;
* требования к разработке интерфейса и т.п.

В нашем случае система должна соответствовать следующим требованиям:

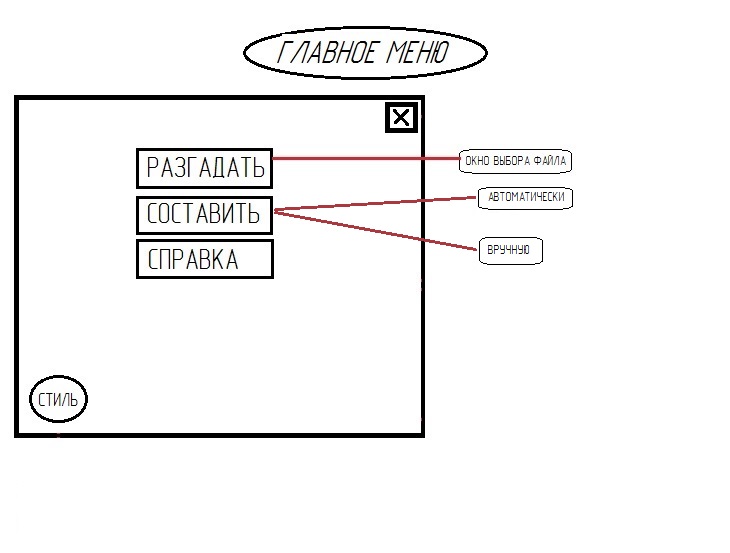
* ограниченность времени автоматической генерации кроссворда;
* предотвращение исключительных ситуаций, перечисленных в пункте 2.2.2;
* интерфейс должен соответствовать требованиям системы.

## 2.3 Проектирование интерфейса пользователя

Интерфейс (от англ. interface – поверхность раздела, перегородка) – совокупность средств и методов взаимодействия между элементами системы.

Интерфейс является основной и наиболее важной частью программы. Поскольку пользователь управляет программой и данными именно через интерфейс, мы уделяем большое внимание понятности и простоте интерфейсов в разработанных нами программах [10].

Для выполнения начальной фазы разработки необходимо погрузиться целиком в задачи пользователей и создать бумажный прототип навигационной модели. Навигационная модель показывает, как необходимо распределять функции или задачи между окнами вашей программы, она определяет, как пользователи смогут перемещаться как между различными задачами, так и внутри отдельной задачи. На рисунке 9 приведена навигационная модель системы автоматизированного составления и разгадывания классического кроссворда.



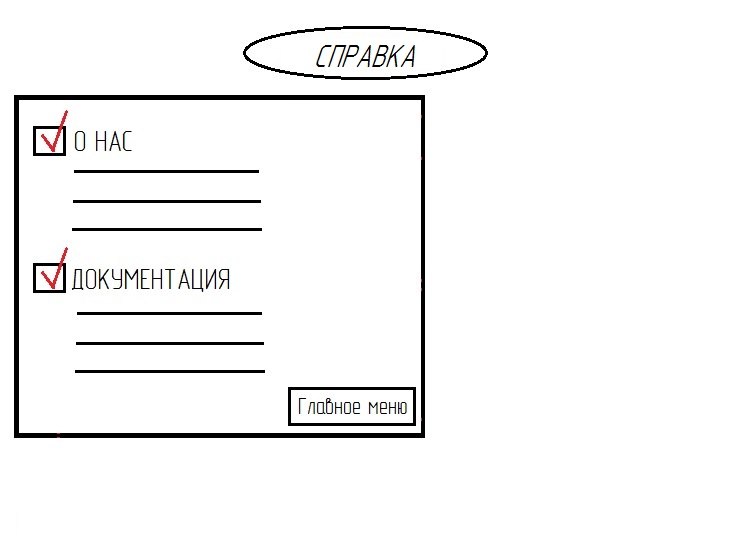
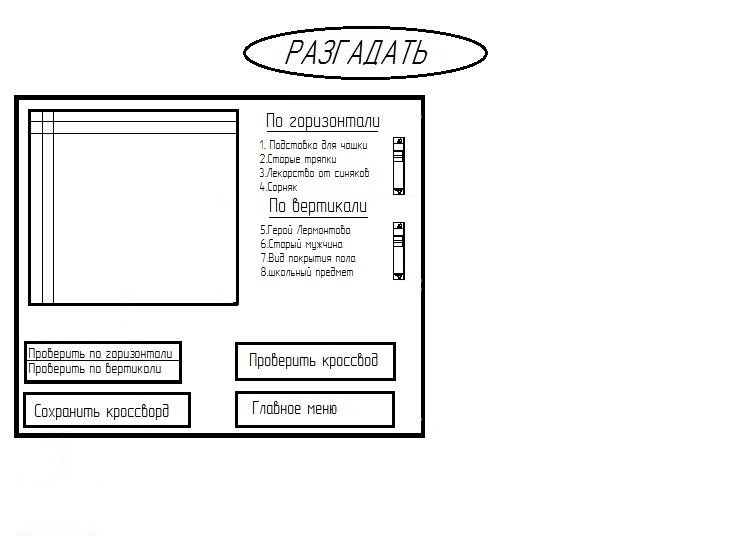
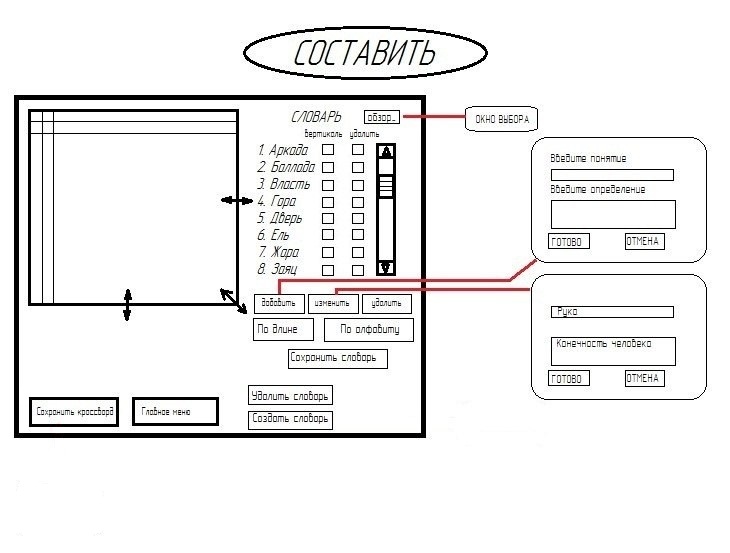
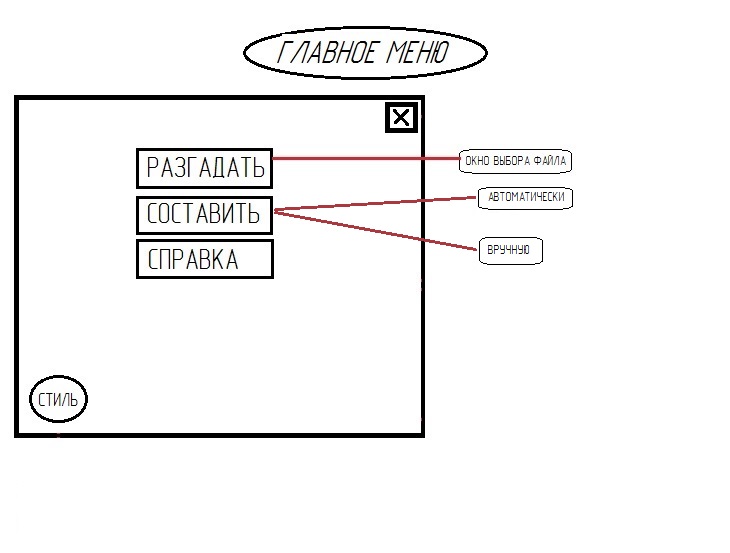


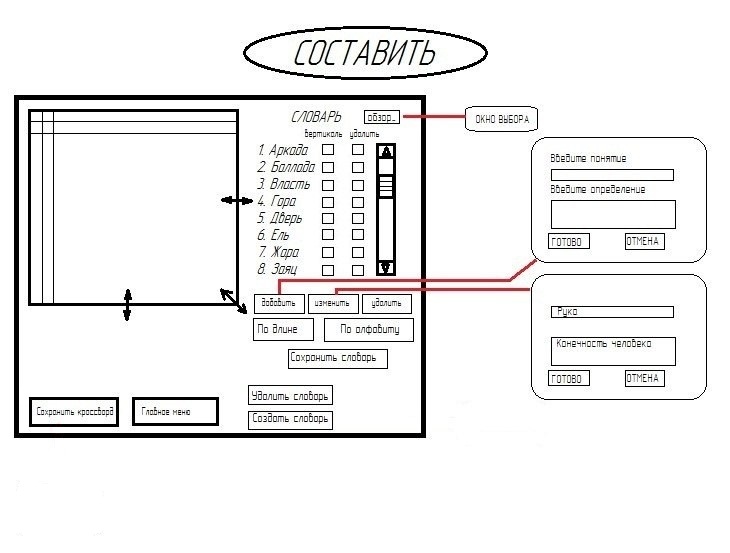
Рисунок 9 – Общее взаимодействие прототипов форм

При запуске программы будет открываться главное окно программы, представленное на рисунке 10. В нем будет находиться меню, состоящее из 3 кнопок-компонентов:

* разгадать;
* составить;
* справка.

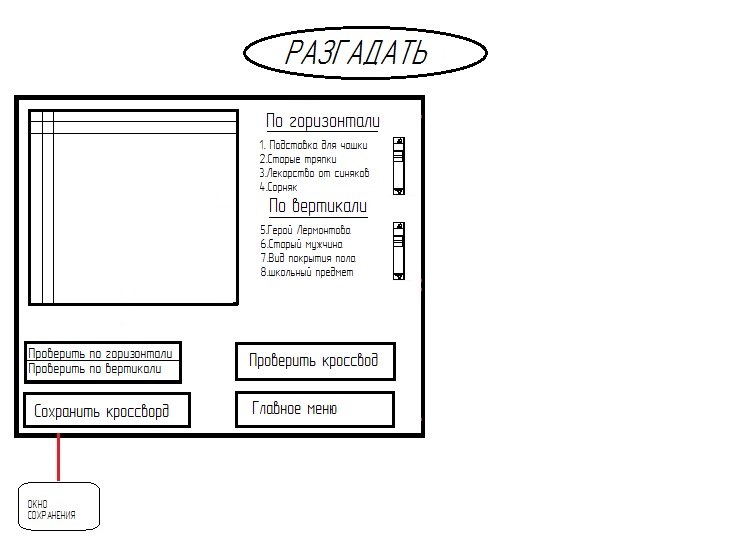
  
Рисунок 10 – Прототип интерфейса (начальная страница)

При нажатии на кнопку «X» основная форма будет закрываться. При нажатии кнопки «Составить» будет выводиться всплывающее меню, предлагающее пользователю автоматическое либо ручное составление. При автоматическом составлении будет загружаться последний созданный кроссворд, при ручном – будет открываться форма составления кроссворда, изображенная на рисунке 11.

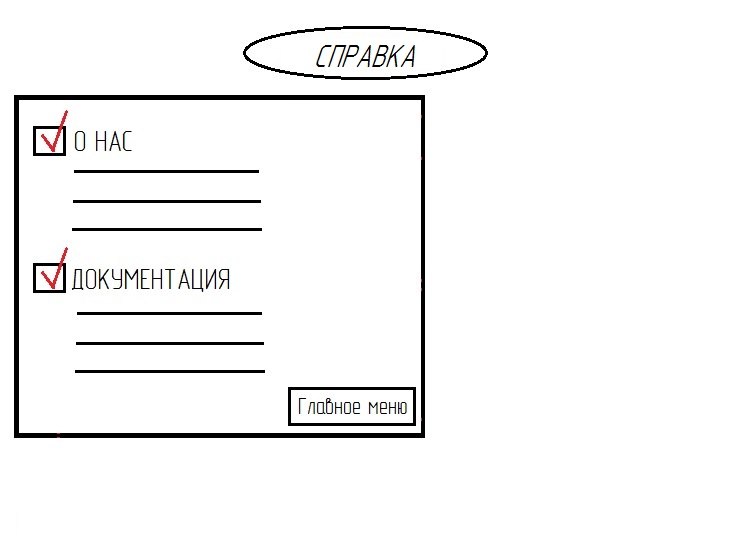
  
Рисунок 11 – Прототип интерфейса (составление кроссворда)

Пользователю будет предложено загрузить словарь, либо создать собственный. Также будут существовать возможности удалить словарь, отсортировать список понятий по длине и ширине, перетащить выбранное слово на сетку кроссворда и, при необходимости, выбрать его ориентацию, добавить, изменить и удалить понятие, сохранить словарь, сохранить кроссворд и перейти в главное меню по нажатию кнопки «Главное меню».

При нажатии на кнопку «Разгадать» будет открываться форма разгадывания кроссворда изображенная на рисунке 12, с возможностью проверить слова по горизонтали, вертикали и кроссворд полностью. Также будет можно сохранить кроссворд и вернуться в главное меню по нажатию кнопки «Главное меню».

  
  
Рисунок 12 – Прототип интерфейса (разгадывание кроссворда)

При нажатии на кнопку «Справка» будет открываться справочное окно, содержащее информацию об авторах и документацию. Данное окно изображено на рисунке 13. Вернуться в главное меню можно будет по нажатию кнопки «Главное меню».

  
  
Рисунок 13 – Интерфейс (справка)

## 2.4 Информационно-логический проект системы

Модель – это упрощенное представление реальности, по существу – это «чертеж» системы: в нее может входить как детальный план, так и более абстрактное представление системы «с высоты птичьего полета» [12]. Хорошая модель всегда включает элементы, которые существенно влияют на результат, и не включает те, которые малозначимы на данном уровне абстракции.

Моделирование позволяет решить четыре различные задачи:

* + 1. Визуализировать систему в ее текущем или желательном для нас состоянии;
    2. Описать структуру или поведение системы;
    3. Получить шаблон, позволяющий сконструировать систему;
    4. Документировать принимаемые решения, используя полученные модели.

### 2.4.1 Язык UML

Унифицированный язык моделирования (Unified Modeling Language – UML) – это стандартный инструмент для разработки «чертежей» программного обеспечения. Его можно использовать для визуализации, спецификации, конструирования и документирования артефактов программных систем. UML подходит для моделирования любых систем – от информационных систем масштаба предприятия до распределенных Web-приложений и даже встроенных систем реального времени [13].

В нотации языка UML определены следующие виды канонических диаграмм:

1. вариантов использования (use case diagram);
2. классов (class diagram);
3. кооперации (collaboration diagram);
4. последовательности (sequence diagram);
5. состояний (statechart diagram);
6. деятельности (activity diagram);
7. компонентов (component diagram);
8. развертывания (deployment diagram).

Перечень этих диаграмм и их названия являются каноническими в том смысле, что представляют собой неотъемлемую часть графической нотации языка UML. Более того, процесс ООАП неразрывно связан с процессом построения этих диаграмм. При этом совокупность построенных таким образом диаграмм является самодостаточной в том смысле, что в них содержится вся информация, которая необходима для реализации проекта сложной системы. Каждая из этих диаграмм детализирует и конкретизирует различные представления о модели сложной системы в терминах языка UML.

### 2.4.2 Диаграмма вариантов использования

Создание диаграммы вариантов использования имеет следующие цели:

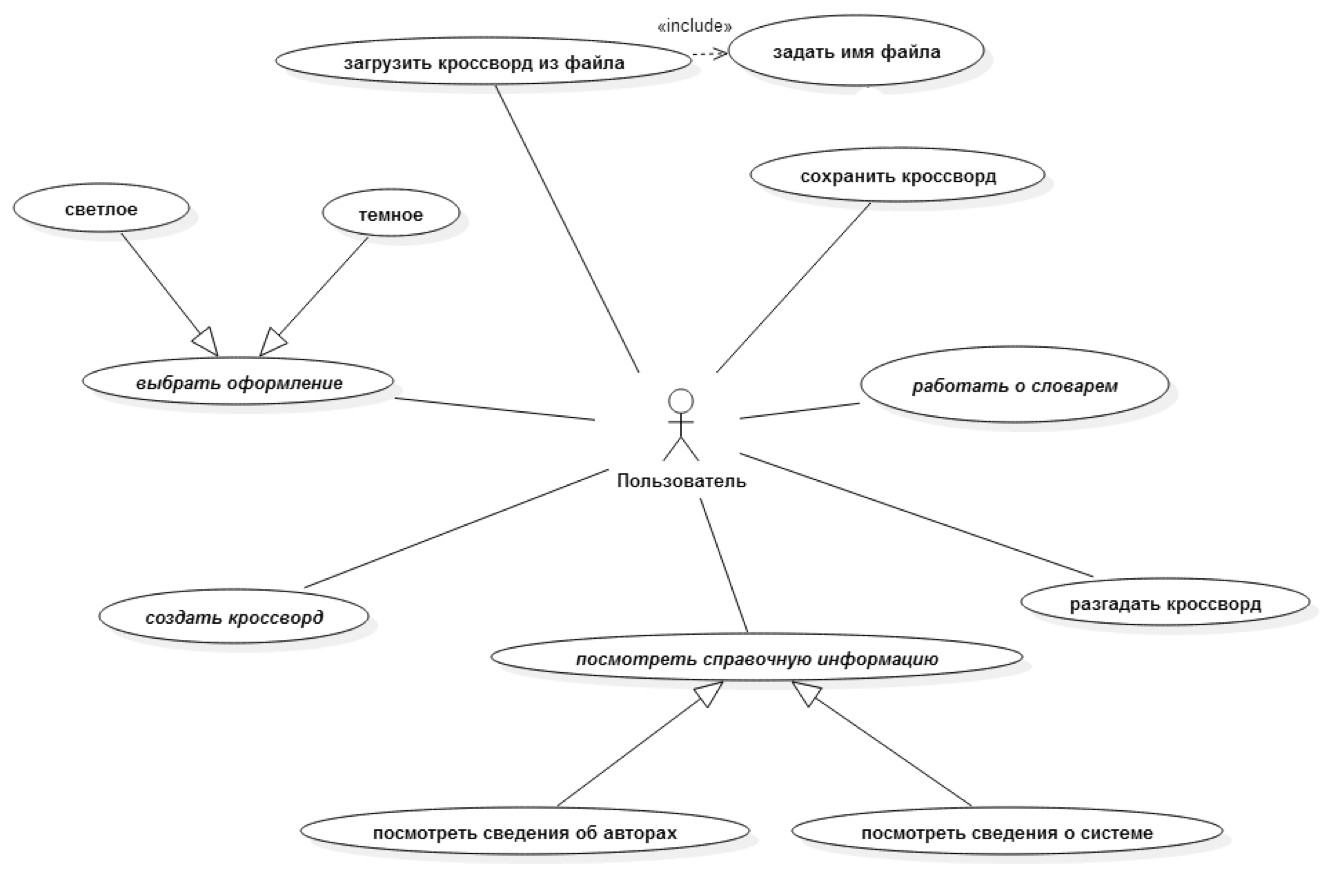
* Определить общие границы и контекст моделируемой предметной области на начальных этапах проектирования системы.
* Сформулировать общие требования к функциональному поведению проектируемой системы.
* Разработать исходную концептуальную модель системы для ее последующей детализации в форме логических и физических моделей.
* Подготовить исходную документацию для взаимодействия разработчиков системы с ее заказчиками и пользователями.

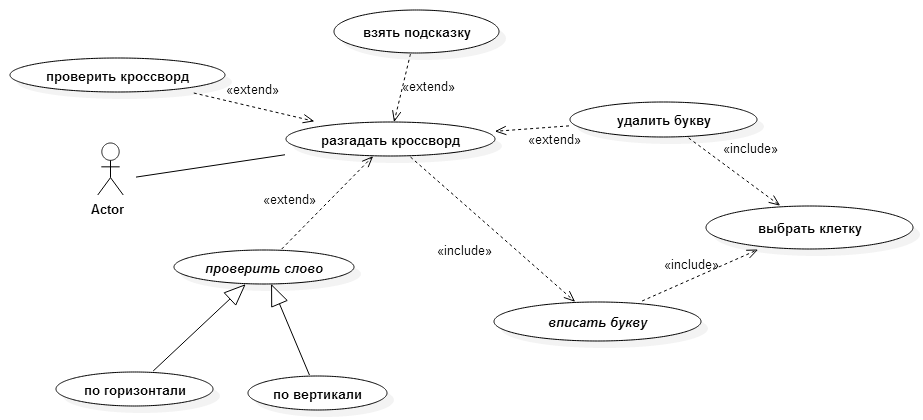
Актер (actor) – согласованное множество ролей, которые играют внешние сущности по отношению к вариантам использования при взаимодействии с ними (это может быть любой объект, субъект или система, взаимодействующая с моделируемой бизнес-системой извне, т.е. человек, техническое устройство, программа и т.п.) [14].

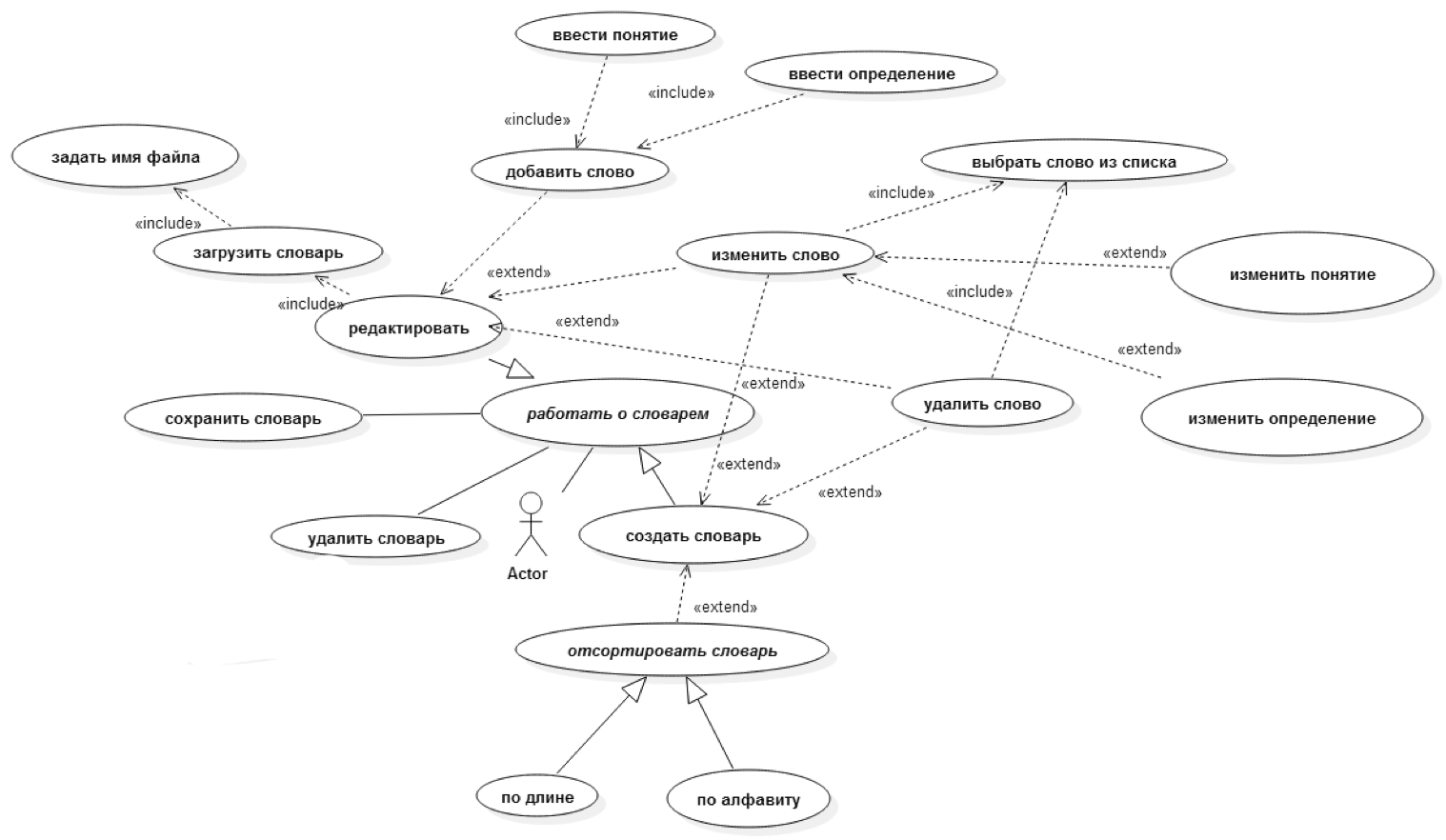
Вариант использования – внешняя спецификация последовательности действий, которые система или другая сущность могут выполнять в процессе взаимодействия с актерами (он определяет набор действий, совершаемый системой при диалоге с актером).

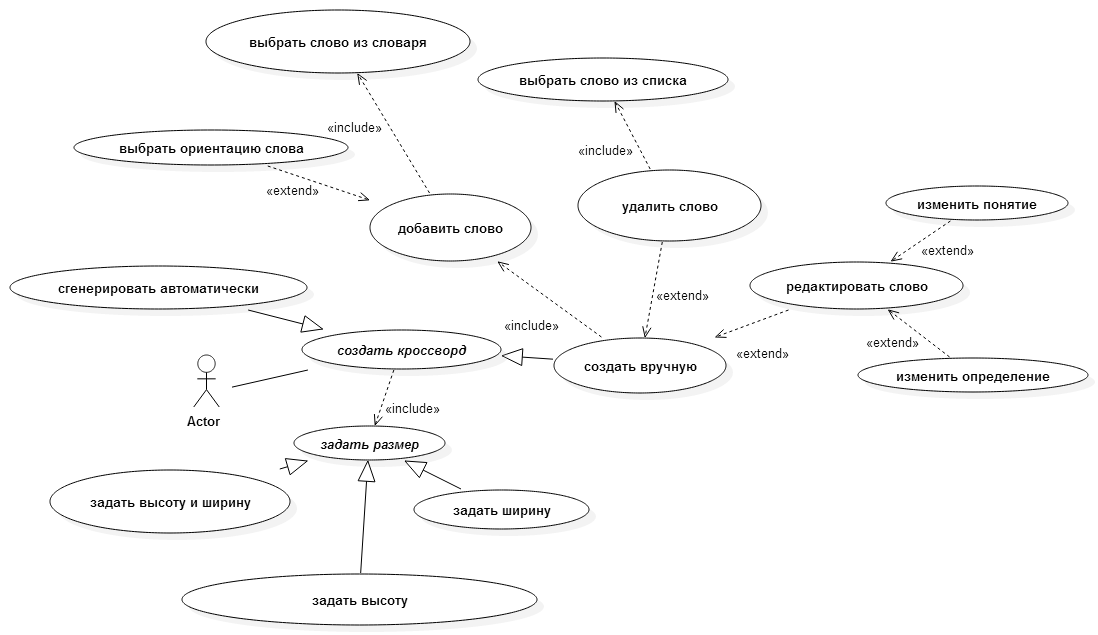
Цель спецификации варианта использования заключается в том, чтобы зафиксировать некоторый аспект или фрагмент поведения проектируемой системы без указания особенностей реализации данной функциональности. В этом смысле каждый вариант использования соответствует отдельному сервису, который предоставляет моделируемая система по запросу актера, т. е. определяет один из способов применения системы. Сервис, который инициализируется по запросу актера, должен представлять собой законченную последовательность действий. Это означает, что после того как система закончит обработку запроса актера, она должна возвратиться в исходное состояние, в котором снова готова к выполнению следующих запросов.

На рисунках 14-17 приведены диаграммы вариантов использования для разрабатываемой системы.

  
Рисунок 14 – Общая диаграмма вариантов использования системы

  
Рисунок 15 – Диаграмма вариантов использования системы (разгадать кроссворд)

Рисунок 16 – Диаграмма вариантов использования системы (работать со словарем)

  
Рисунок 17 – Диаграмма вариантов использования системы (создать кроссворд)

Сценарий – определенная последовательность действий, которая описывает действия актеров и поведение моделируемой системы в форме обычного текста [15].

В контексте языка UML сценарий используется для дополнительной иллюстрации взаимодействия актеров и вариантов использования.

Рассмотрим сценарии для вариантов использования «Сохранить кроссворд», «Добавить слово», «Вписать букву» и «Проверить слово» для автоматизированной системы составления и разгадывания классического кроссворда.

**Вариант использования: Добавить слово**

*Краткое описание*. Даёт возможность любому пользователю добавить понятие в словарь. Пользователь щелчком левой кнопки мыши по кнопке «Добавить» открывает окно добавления понятия и его описания. В этом окне пользователь имеет возможность ввести понятие и его определение. После ввода информации пользователь может их сохранить щелчком правой кнопки мыши по кнопке «Готово» или отменить действие щелчком правой кнопки мыши по кнопке «Отмена»

*Актант*. Пользователь.

*Предусловия*. Компьютер пользователя включён, на экране – главное окно программы в режиме «Составить кроссворд».

*Основной поток событий.*

1) На экране форма составления кроссворда с кнопками «Обзор», «Добавить», «Изменить», «Удалить», отсортировать «По длине», «По алфавиту», «Сохранить словарь», «Удалить словарь», «Создать словарь», «Главное меню» и «Сохранить кроссворд».

2) Щёлкнута кнопка «Добавить».

3) Система открывает окно ввода понятия и его определения, пользователь вводит их и нажимает кнопку «Готово».

А1: Щелкнута кнопка «Отмена».

4) Система закрывает окно ввода понятия и определения и возвращает на экран форму составления кроссворда. Вариант использования завершается успешно.

*Альтернативы*. Нет

А1: Щёлкнута кнопка «Отмена».

А1.1. Система закрывает форму ввода понятия и определения и выводит на экран окно составления кроссворда. Вариант использования завершается.

*Постусловия*. При успешном завершении на экране расположена форма составления кроссворда.

**Вариант использования: Сохранить кроссворд**

*Краткое описание*. Даёт возможность любому пользователю сохранить составленный кроссворд. Пользователь щелчком левой кнопки мыши по кнопке «Составить» открывает окно составления кроссворда. Здесь присутствуют возможности открыть уже имеющийся словарь и работать с ним, либо создать новый словарь, а также добавить слово на сетку кроссворда. После внесения изменений пользователь может их сохранить щелчком правой кнопки мыши по кнопке «Сохранить кроссворд.»

*Актант*. Пользователь.

*Предусловия*. Компьютер пользователя включён, на экране – главное окно программы в режиме «Составить кроссворд».

*Основной поток событий.*

1. На экране форма составления кроссворда с кнопками «Обзор», «Добавить», «Изменить», «Удалить», отсортировать «По длине», «По алфавиту», «Сохранить словарь», «Удалить словарь», «Создать словарь», «Главное меню» и «Сохранить кроссворд».
2. Щёлкнута кнопка «Сохранить кроссворд».
3. Система сохраняет файл и выводит на экран форму составления кроссворда. Вариант использования завершается успешно.

Альтернативы. Нет.

Постусловия. При успешном завершении на экране –форма составления кроссворда.

**Вариант использования: Вписать букву**

*Краткое описание*. Даёт возможность любому пользователю вписать букву в слово в кроссворде. Пользователь на главном экране приложения щелчком левой кнопки мыши по кнопке «Разгадать» открывает окно выбора файла кроссворда. После выбора файла кроссворда открывается окно разгадывания кроссворда. Здесь присутствуют сетка самого кроссворда, определения понятий, а так же функциональные кнопки такие как: «Проверить по горизонтали», «Проверить по вертикали», «Проверить кроссворд», «Сохранить кроссворд», «Главное меню». Во время разгадывания кроссворда пользователь может сохранить его текущее состояние в файл, нажав на кнопку «Сохранить кроссворд», и вернуться к нему позже.

*Актант*. Пользователь.

*Предусловия*. Компьютер пользователя включён, на экране – главное окно программы в режиме «Разгадать кроссворда».

*Основной поток событий.*

1. На экране появляется форма разгадывания кроссворда, на котором расположены «Список заданий по горизонтали», «Список заданий по вертикали» с полосами прокрутки справа от них, кнопки «Проверить по горизонтали», «Проверить по вертикали», «Проверить кроссворд», «Сохранить кроссворд», «Главное меню».
2. Пользователь щелкает по ячейке на сетке кроссворда. Выделяются ячейки, относящиеся к выбранному слову. Если ячейка находится на пересечении слов, то по умолчанию выбирается слово с горизонтальной ориентацией.
3. Пользователь вводит букву в выбранную ячейку.

А1: Выбрана другая ячейка.

Альтернативы.

А1: Выбрана другая ячейка.

А1.1. Переход к пункту 2 основного потока событий.

Постусловия. При успешном завершении на экране – форма разгадывания кроссворда.

**Вариант использования: Проверить слово**

*Краткое описание*. Даёт возможность любому пользователю проверить верно ли отгадано заполненное слово. Пользователь на главном экране приложения щелчком левой кнопки мыши по кнопке «Разгадать» открывает окно выбора файла кроссворда. После выбора файла кроссворда открывается окно разгадывания кроссворда. Здесь присутствуют сетка самого кроссворда, определения понятий, а так же функциональные кнопки такие как: «Проверить по горизонтали», «Проверить по вертикали», «Проверить кроссворд», «Сохранить кроссворд», «Главное меню». Во время разгадывания кроссворда пользователь может сохранить его текущее состояние в файл, нажав на кнопку «Сохранить кроссворд», и вернуться к нему позже.

*Актант*. Пользователь.

*Предусловия*. Компьютер пользователя включён, на экране – главное окно программы в режиме «Разгадать кроссворда».

*Основной поток событий.*

1. Пользователь щелкает по ячейке на сетке кроссворда. Выделяются ячейки, относящиеся к выбранному слову. Если ячейка находится на пересечении слов, то по умолчанию выбирается слово с горизонтальной ориентацией.
2. В зависимости от ориентации выбранного слова становится активной либо кнопка «Проверить по горизонтали», либо кнопка «Проверить по вертикали».
3. Щелкнута активная кнопка «Проверить по горизонтали» / «Проверить по вертикали».
4. Фон ячеек выбранного слова подсвечивается зеленым цветом (слово разгадано верно) или красным (слово разгадано неверно или заполнено не полностью).

А1: Выбрана другая ячейка.

Альтернативы.

А1: Выбрана другая ячейка.

А1.1. Переход к пункту 2 основного потока событий.

Постусловия. При успешном завершении на экране – форма разгадывания кроссворда.

### 2.4.3 Диаграмма классов

Диаграммы классов – это тип диаграмм, которые создаются при моделировании объектно-ориентированных систем, они показывают набор классов, интерфейсов и коопераций, а также их связи. На практике диаграммы классов применяют для моделирования статического представления системы, они служат основой для целой группы взаимосвязанных диаграмм – диаграмм компонентов и диаграмм размещения [7].

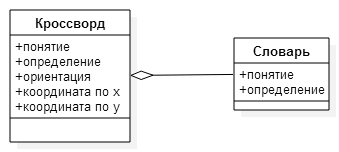
Агрегация – простая ассоциация между двумя классами отражает структурное отношение между равноправными сущностями, когда оба класса находятся на одном концептуальном уровне и ни один не является более важным, чем другой.

Отношение зависимости используется тогда, когда изменение одного элемента модели может потребовать изменения другого, зависимого от него. Это наиболее общая форма отношений, все другие виды отношений можно считать частным случаем данного.

Отношение обобщения показывает, что некоторые объекты являются потомками базового (родительского) класса.

Отношение ассоциации показывает, что некоторые объекты образуют группу (ассоциацию). Наиболее простая ассоциация бинарная. Для данного вида отношения может быть указана мощность связи (например, один ко многим).

Разрабатываемая система состоит из двух равноправных классов: «Кроссворд» и «Словарь», в которых содержится вся необходимая информация. Диаграмма классов разрабатываемой системы представлена на рисунке 18.

  
Рисунок 18 – Диаграмма классов системы

В классе «Кроссворд» содержится понятие, определение, ориентация слова и координаты по X и по Y. В классе «Словарь» содержится понятие и определение. В таблицах 3 и 4 приведены описания соответствующих классов.

Таблица 3– Описание класса «Кроссворд»

|  |  |  |
| --- | --- | --- |
| Поля | | |
|  | Название | Тип |
| + | Понятие | Строковое |
| + | Определение | Строковое |
| + | Ориентация | Строковое |
| + | Координата по Х | Целое |
| + | Координата по У | Целое |

Таблица 4– Описание класса «Словарь»

|  |  |  |
| --- | --- | --- |
| Поля | | |
|  | Название | Тип |
| + | Понятие | Строковое |
| + | Определение | Строковое |

### 2.4.4 Диаграмма состояний

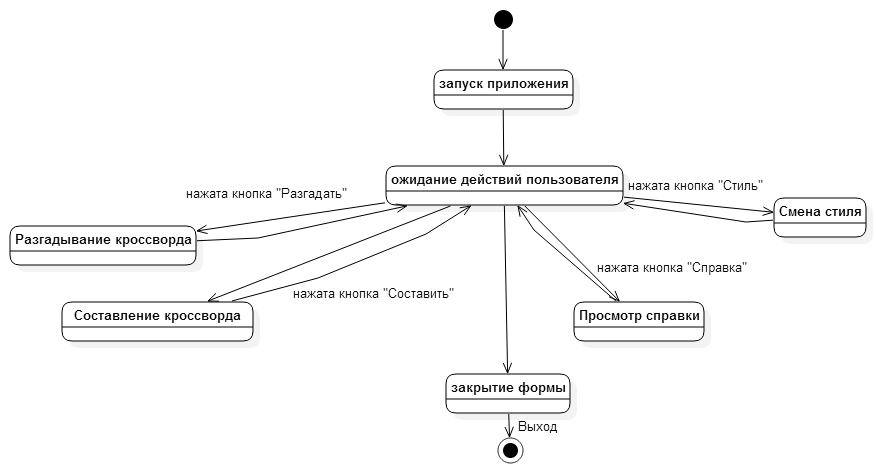
Главное предназначение этой диаграммы – описать возможные последовательности состояний и переходов, которые в совокупности характеризуют поведение элемента модели в течение его жизненного цикла. Диаграмма состояний представляет динамическое поведение сущностей, на основе спецификации их реакции на восприятие некоторых конкретных событий. Системы, которые реагируют на внешние действия от других систем или от пользователей, иногда называют реактивными. Если такие действия инициируются в произвольные случайные моменты времени, то говорят об асинхронном поведении модели [16]. На рисунках 18 и 19 приведены общая диаграмма состояние и диаграмма состояний для варианта использования «Разгадать».

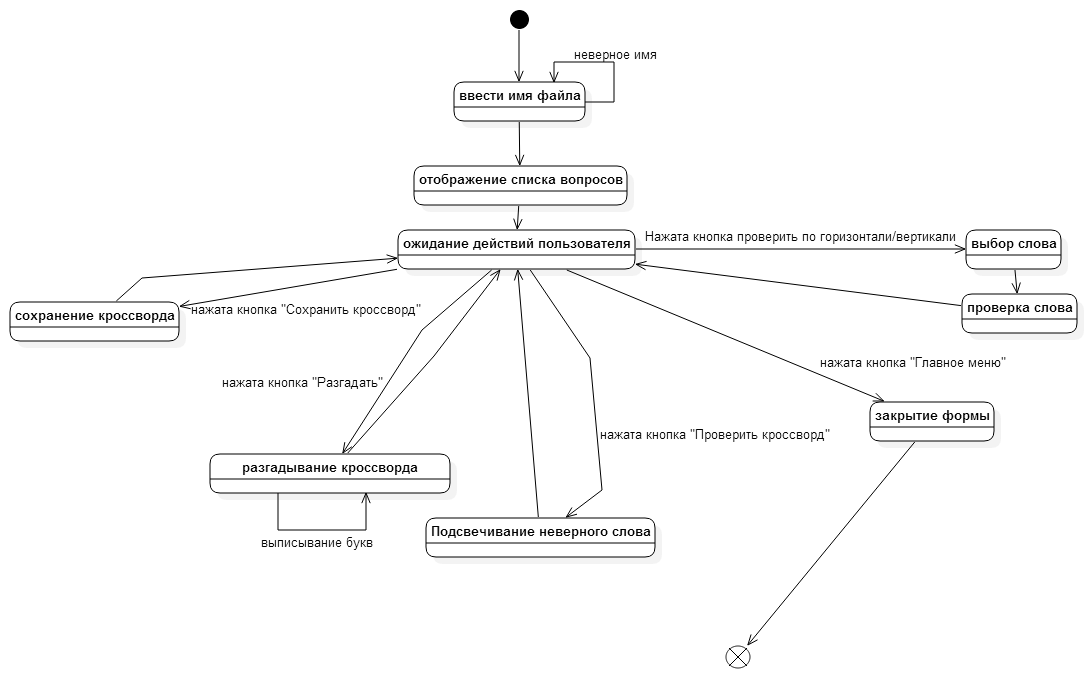
После запуска приложения отображается основное меню программы, система ждет действий пользователя. При нажатии кнопки «Разгадать» открывается окно разгадывания кроссворда, при нажатии кнопки «Составить» открывается окно составления кроссворда, при нажатии кнопки «Справка» отображается справочная информация о системе, при нажатии кнопки «Стиль» происходит смена цветового оформления приложения. По нажатию на «Х» форма закрывается.

При разгадывании происходит загрузка файла кроссворда, при неверном файле выдается ошибка. На форме отображается список вопросов и система ждет действий пользователя. При нажатии кнопки «Сохранить кроссворд» происходит сохранение файла, при нажатии «Разгадать» – разгадывание, путем вписывания букв, при нажатии «Проверить кроссворд» – подсветка неверно введённых слов, при нажатии «Проверить по горизонтали/вертикали» подсвечивается соответствующее неверное слово, по нажатию главное меню форма «Разгадать» закрывается и происходит выход в главное меню.

### 2.4.5 Диаграмма деятельности

Диаграммы деятельности – это один из пяти видов диаграмм, применяемых в UML для моделирования динамических аспектов поведения системы. Диаграмма деятельности – это, по существу, блок-схема, которая показывает, как поток управления переходит от одной деятельности к другой. Как правило, они применяются, чтобы промоделировать последовательные (а иногда и параллельные) шаги вычислительного процесса. С помощью диаграмм деятельности можно также моделировать жизнь объекта, когда он переходит из

  
Рисунок 19 – Общая диаграмма состояний системы

  
Рисунок 20 – Диаграмма состояний «Разгадывание»

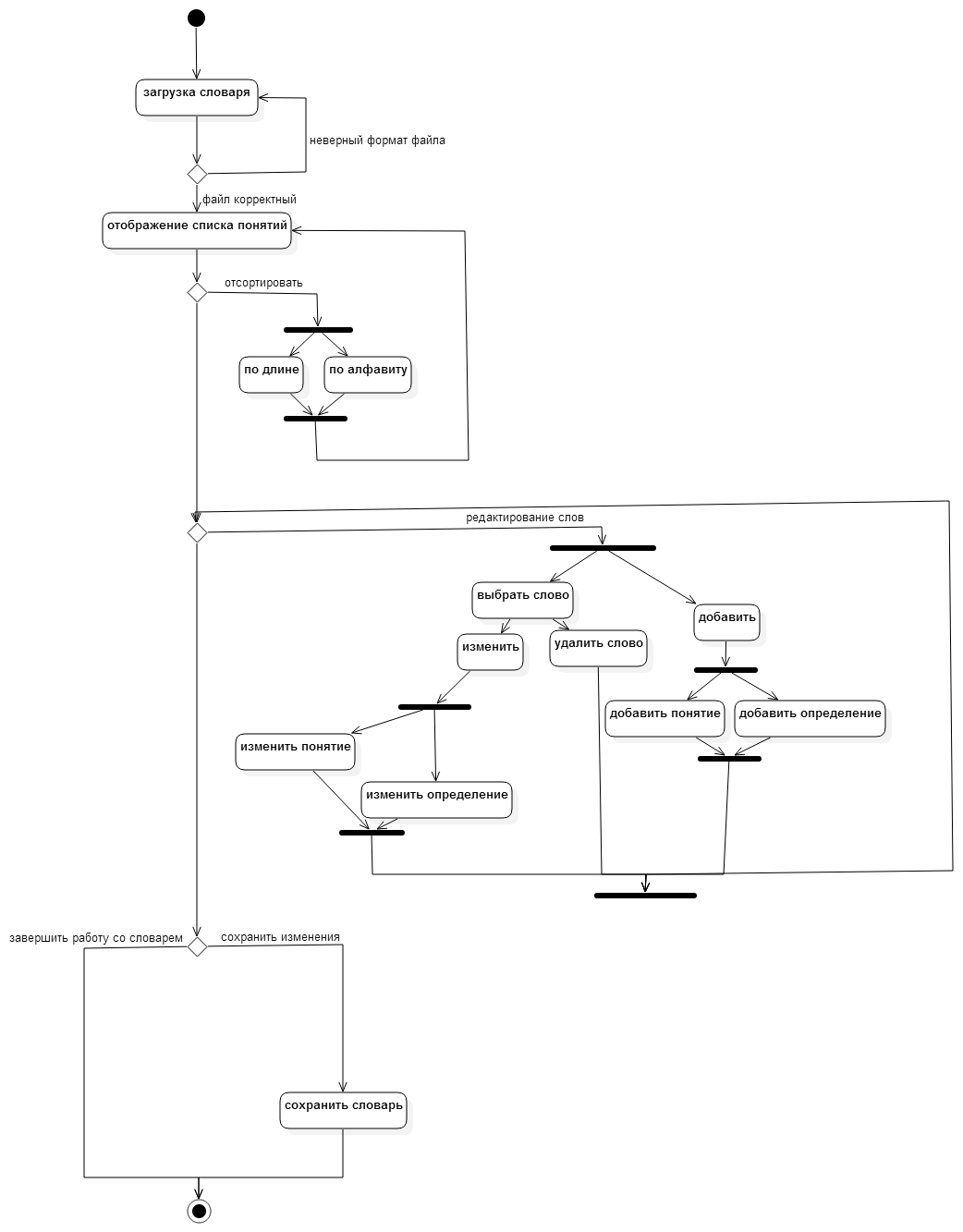
одного состояния в другое в разных точках потока управления. Диаграммы деятельности могут использоваться самостоятельно для визуализации, специфицирования, конструирования и документирования динамики совокупности объектов, но они пригодны также и для моделирования потока управления при выполнении некоторой операции [7].

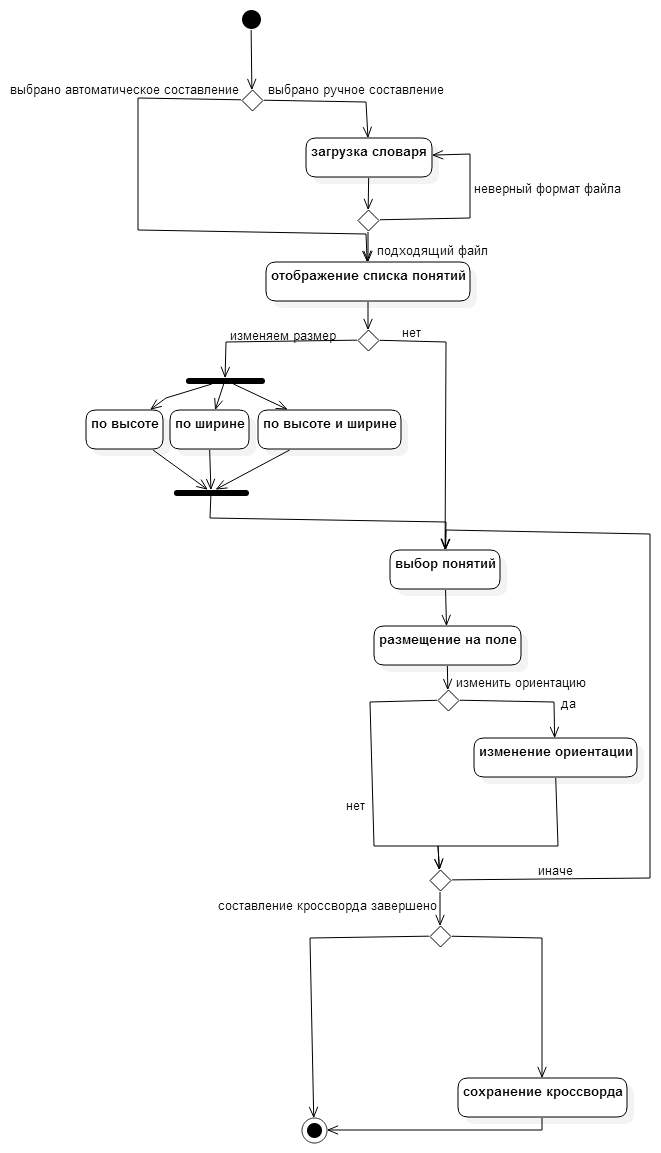
Состояние деятельности – состояние в графе деятельности, которое служит для представления процедурной последовательности действий, требующих определенного времени.

Можно построить несколько диаграмм деятельности для одной и той же системы, причем каждая из них будет фокусироваться на разных аспектах системы, показывать различные действия, выполняющиеся внутри ее. Читатель, конечно же, понял, что, когда мы говорим о динамике, мы подразумеваем поведение системы в целом или ее частей. Говоря более формально, диаграммы активности, в общем-то, не имеют монополии на описание поведенческих особенностей динамических частей системы. Для этой же цели могут использоваться еще диаграммы прецедентов, последовательности, кооперации и состояний. Почему же мы говорим именно о диаграмме активности? Нет, не только потому, что так называется эта лекция. Именно на диаграмме деятельности представлены переходы потока управления от одной деятельности к другой. На рисунках 20-21 приведены диаграммы деятельности «Работа со словарем» и «Создать кроссворд».

Работа со словарем заключается в загрузке словаря с проверкой корректности файла, отображении списка понятий с возможностью сортировки словаря по длине и алфавиту, редактировании слов, состоящем из добавления нового слова, изменения существующего и удаления уже имеющегося. Далее есть возможность завершить работу со словарем либо предварительно сохранить ранее внесенные изменения.

Создание кроссворда состоит из следующих этапов: выбор ручного (с загрузкой файла и проверкой его корректности) либо автоматического режима составления кроссворда, отображение списка понятий, возможность изменения размера кроссворда (по высоте/ширине/высоте и ширине), выбор понятий с размещением их на поле, возможность смены ориентации слова и завершение работы с кроссвордом с возможностью сохранить внесенные изменения.

Рисунок 21 – Диаграмма деятельности «Работа со словарем»

  
Рисунок 22 – Диаграмма деятельности «Создать кроссворд»

### 2.4.6 Диаграмма последовательности

Диаграмма последовательности – это диаграмма, чаще всего, описывающая один сценарий приложения. На диаграмме изображаются экземпляры объектов и сообщения, которыми они обмениваются в рамках одного прецедента (use case) [7].

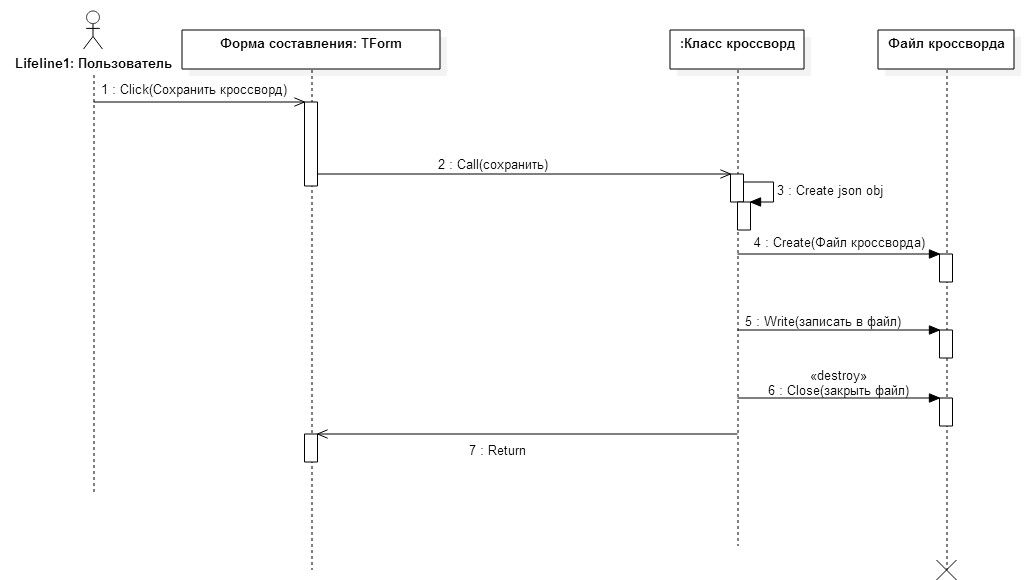
На диаграмме последовательности, каждый участник представлен вместе со своей линией жизни (lifeline), это вертикальная линия под объектом, вертикально упорядочивающая сообщения на странице. Все сообщения на диаграмме следует читать сверху вниз. Каждая линия жизни имеет полосу активности (прямоугольники), которая показывает интервал активности каждого участника при взаимодействии.

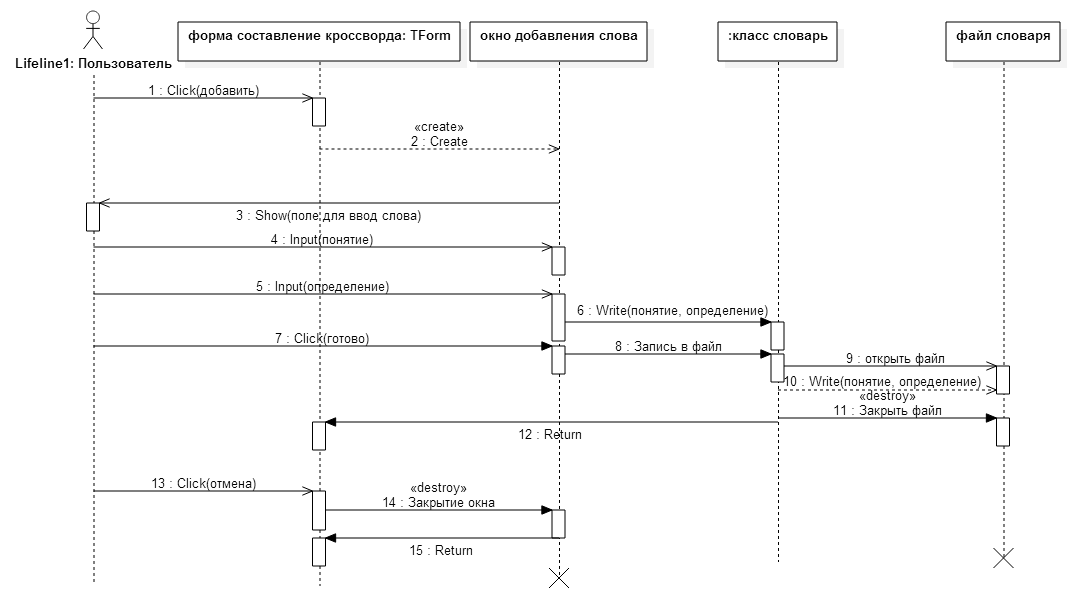
Сообщения можно разделить на 2 вида: синхронные (synchronous message) – требующие возврата ответа и асинхронные (asynchronous message) – ответ не требуется и вызывающий объект может продолжать работу. На диаграмме синхронные вызовы обозначаются закрашенными стрелочками. Асинхронные – не закрашенными или половинными стрелочками.

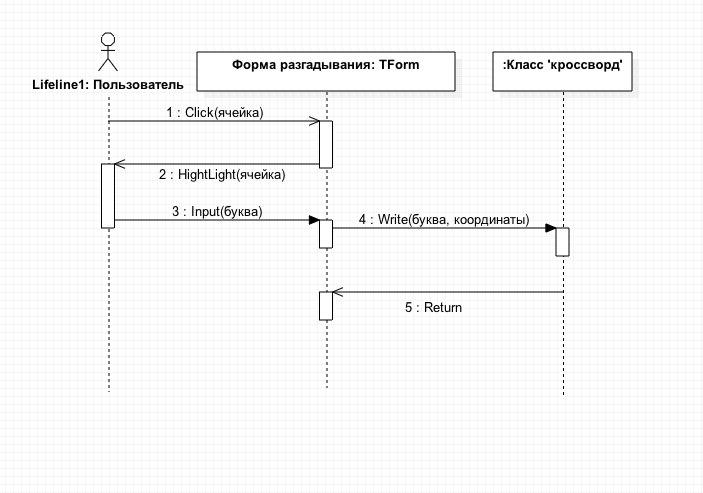
У первого сообщения, в нашей диаграмме, нет участника, пославшего его, поскольку оно приходит от неизвестного источника. Такое сообщение называется найденным сообщением (found message).

Обратной пунктирной стрелкой показан возврат ответа на сообщение message (т.к. сообщение message является синхронным). Лучше применять изображение возврата только в тех случаях, когда это поможет лучше понять устройство взаимодействия. Во всех остальных случаях, стоит опускать изображения возвратов, т.к. они будут вносить некоторую неразбериху. Просто, при использовании синхронного сообщения, стоит помнить, что у него всегда есть возврат.

Если в сообщении требуется передать параметры, то они указываются в скобках через запятую, с указанием типа параметра. На рисунках 22-25 приведены диаграммы последовательностей сценариев программы.

 Рисунок 23 – Диаграмма последовательности для варианта использования «Сохранить кроссворд»

 Рисунок 24 – Диаграмма последовательности для варианта использования «Добавить слово»

  
Рисунок 25 – Диаграмма последовательности для варианта использования «Вписать букву»

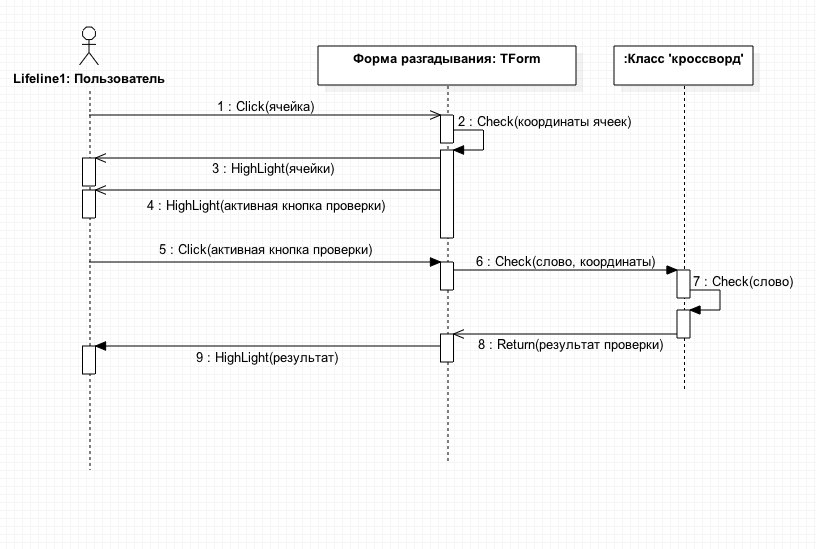
  
Рисунок 26 – Диаграмма последовательности для варианта использования «Проверить слово»

Диаграмма последовательности для варианта использования «Сохранить кроссворд»:

1. пользователь кликает кнопку «Сохранить кроссворд» на форме составления кроссворда;
2. вызывается метод сохранения;
3. в классе кроссворд создается json-объект;
4. создается файл кроссворда;
5. происходит запись в файл;
6. закрытие файла;
7. возврат на форму составления кроссворда.

Диаграмма последовательности для варианта использования «Добавить слово»:

1. пользователь кликает кнопку «Добавить»;
2. создается окно добавления слова и появляются поля для ввода понятия и определение;
3. ввод и сохранение понятия и определения;
4. пользователь кликает кнопку «Готово»;
5. происходит их запись в файл (открытие файла, запись слова, закрытие файла);
6. возврат на форму составления кроссворда.

Диаграмма последовательности для варианта использования «Вписать букву»:

1. открыта форма разгадывания;
2. пользователь кликает по интересующей ячейке;
3. данная ячейка выделяется;
4. пользователь вводит букву;
5. в объект класса кроссворд записывается буква и координаты;
6. возврат на форму разгадывания кроссворда.

Диаграмма последовательности для варианта использования «Проверить слово»:

1. открыта форма разгадывания;
2. пользователь кликает по интересующей ячейке;
3. проверка координат ячейки;
4. данная ячейка выделяется;
5. выделяется активная кнопка проверки;
6. пользователь кликает по ней;
7. происходит проверка выбранного слова по переданным координатам;
8. возврат на форму разгадывания кроссворда;
9. подсветка соответствующего результата.

## 2.5 Выбор и обоснование комплекса программных средств

### 2.5.1 Выбор языка программирования и среды разработки

Скриптовый язык JavaScript создан компанией Netscape в 1995 году. Его первое название было «LiveScript» (переводится как «живой сценарий»), но, так как уже в то время, язык программирования Java набирал популярность, то разработчики решили для удобства использования скриптов, сделать основные конструкции похожими, соответственно, сам язык переименовали в JavaScript. Если вас интересуют уроки по javascript, то советую сайт sitear.ru. Там находятся отличные материалы по этой теме.

Первое, что дает нам JavaScript, это возможность значительно снизить нагрузку на веб-сервер за счёт того, что основные сценарии для работы со страницей будут выполняться на стороне пользователя. Такие скрипты применяют для создания интерактивных страниц, для написания программ, например, легкой браузерной игры или калькулятора. Большинство сервисов и сайтов применяют JavaScript для определения браузера, операционной системы и так далее, для того, чтобы более корректно отобразить страницу на конкретном устройстве. Особенно это актуально для обозревателя Internet Explorer. Именно JavaScript позволяет взаимодействовать с cookie, хранить часть информации на стороне пользователя, и использовать ее при повторном посещении им сайта.

Еще один способ применения: проверка форм ввода информации на правильность без перезагрузки страницы. Теперь кратко о плюсах:

* В отличие от HTML5, поддерживается практически всеми браузерами.
* Прост в освоении и применении.
* Огромная библиотека готовых полезных скриптов.
* Позволяет разгрузить сервер, ведь выполняется на стороне пользователя.

WebStorm – интегрированная среда разработки на JavaScript, CSS & HTML от компании JetBrains, разработанная на основе платформы IntelliJ IDEA.

WebStorm обеспечивает автодополнение, анализ кода на лету, навигацию по коду, рефакторинг, отладку, и интеграцию с системами управления версиями. Важным преимуществом интегрированной среды разработки WebStorm является работа с проектами (в том числе, рефакторинг кода JavaScript, находящегося в разных файлах и папках проекта, а также вложенного в HTML). Поддерживается множественная вложенность (когда в документ на HTML вложен скрипт на Javascript, в который вложен другой код HTML, внутри которого вложен Javascript) – то есть в таких конструкциях поддерживается корректный рефакторинг.

### 2.5.2 Выбор операционной системы

Были рассмотрены две наиболее популярные среди пользователей операционные системы: Windows и Mac OS.

Mac OS – это операционная система, разработанная компанией Apple, и в настоящее время является второй наиболее часто используемой ОС после Windows. У нее менее 20% доли рынка. Mac OS в отличие от Windows основана на Unix. Таким образом, Mac OS считается частью семейства Unix ОС, как и Linux. К основным плюсам можно отнести скорость, надежность и стабильность работы, отсутствие вирусов, удобность интерфейса, предварительный (быстрый) просмотр и большое количество полезных программ.

Windows – это одна из самых известных операционных систем, разработанных Microsoft. В 9 из 10 домах и предприятий в настоящее время по крайней мере один компьютер с Windows. ОС Windows первоначально был основан на MS-DOS. Эти серии ОС Windows известны как 9-ые серии. Все последующие ОС Windows основаны на Windows NT. К основным плюсам относится большое количество дополнительных программ, множество игр и привычность, простота работы.

# 3 РЕАЛИЗАЦИЯ СИСТЕМЫ

## 3.1 Разработка и описание интерфейса пользователя

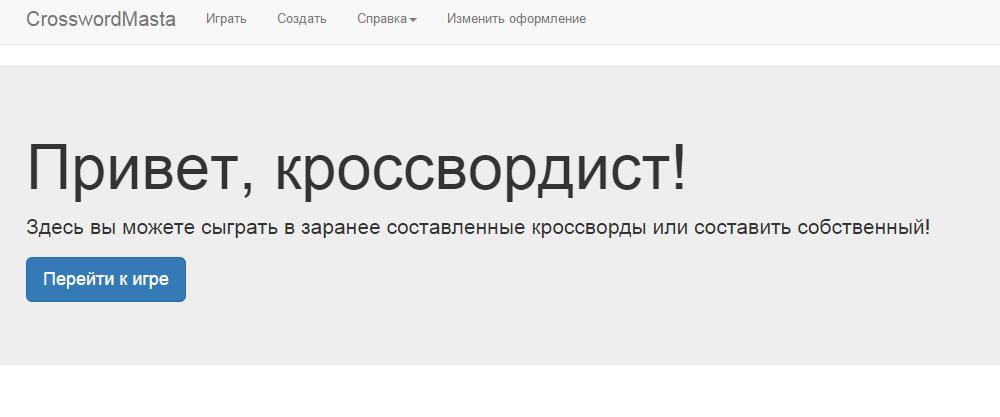
Интерфейс – система правил и средств, регламентирующая и обеспечивающая взаимодействие нескольких процессов или объектов [17].

Пользовательский интерфейс (ПИ) – система правил и средств, регламентирующая и обеспечивающая взаимодействие программы с пользователем. В понятие пользовательского интерфейса (ПИ) входит не только, и даже не столько, картинка на экране – трехмерная, анимированная или просто выполненная в модном дизайне, а способы взаимодействия пользователя с системой.

В целях предоставления пользователю удобного доступа к функциональности системы авторами был разработан интерфейс, максимально удобный для работы пользователя с системой. На рисунке 27 представлена главная форма приложения. В ней находится меню, состоящее из 3 кнопок-компонентов:

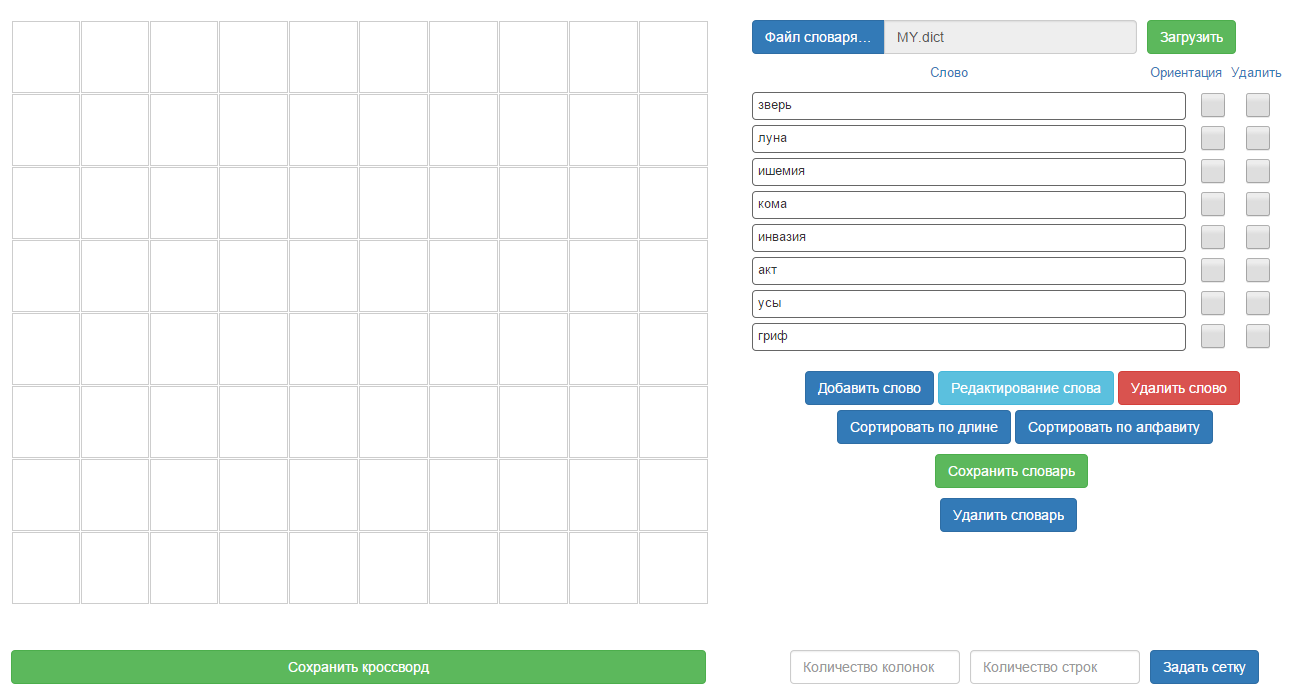
* разгадать;
* составить;
* справка.

Также есть возможность изменить стиль отображения приложения.

  
Рисунок 27 – Главная форма приложения

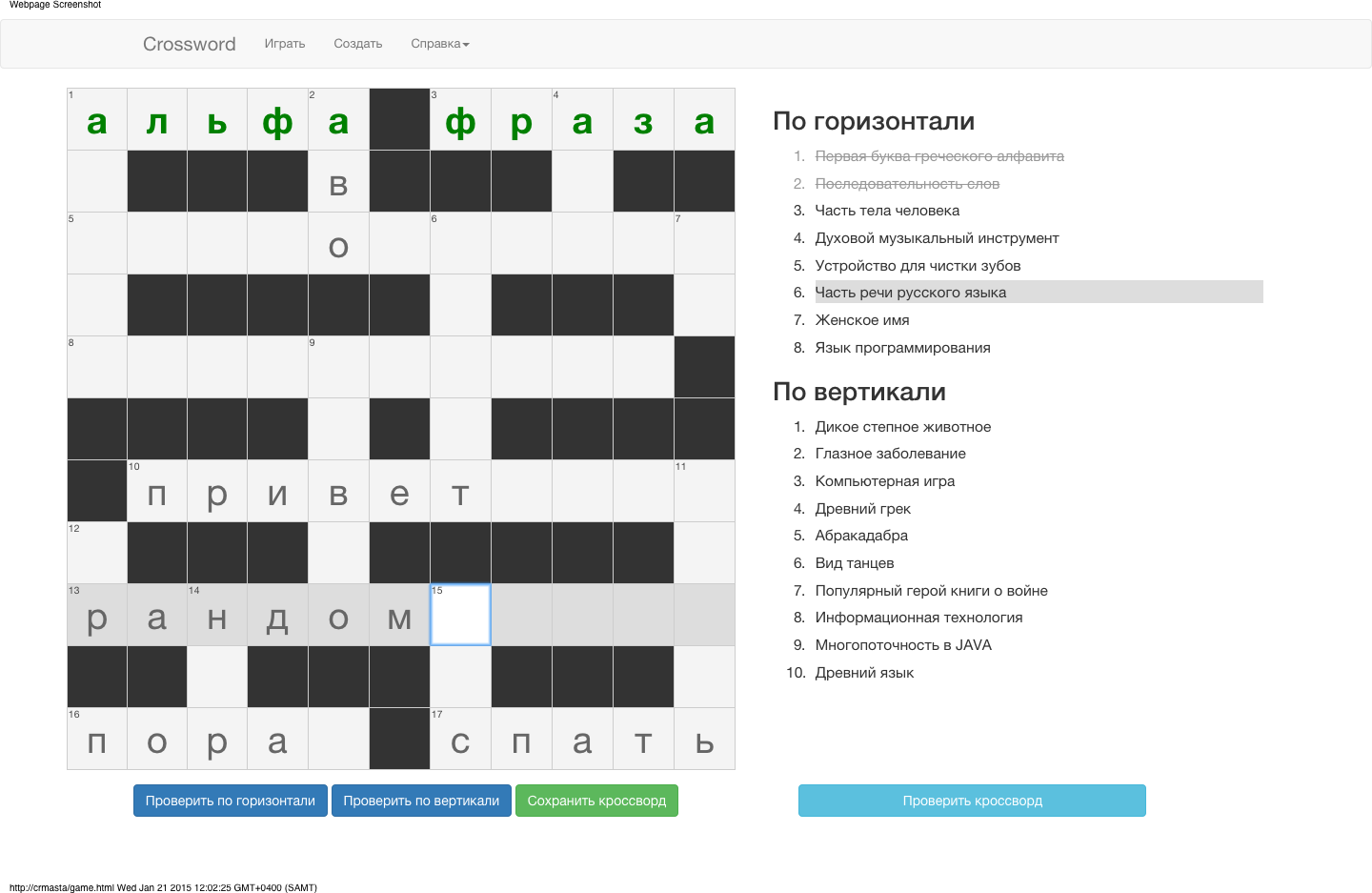
### 3.1.1 Разработка и описание пользовательского меню

При нажатии на кнопку «X» основная форма закрывается. При нажатии кнопки «Составить», выводится всплывающее меню, предлагающее пользователю автоматическое либо ручное составление. При автоматическом составлении загружается последний созданный кроссворд, при ручном открывается форма составления кроссворда, изображенная на рисунке 28.

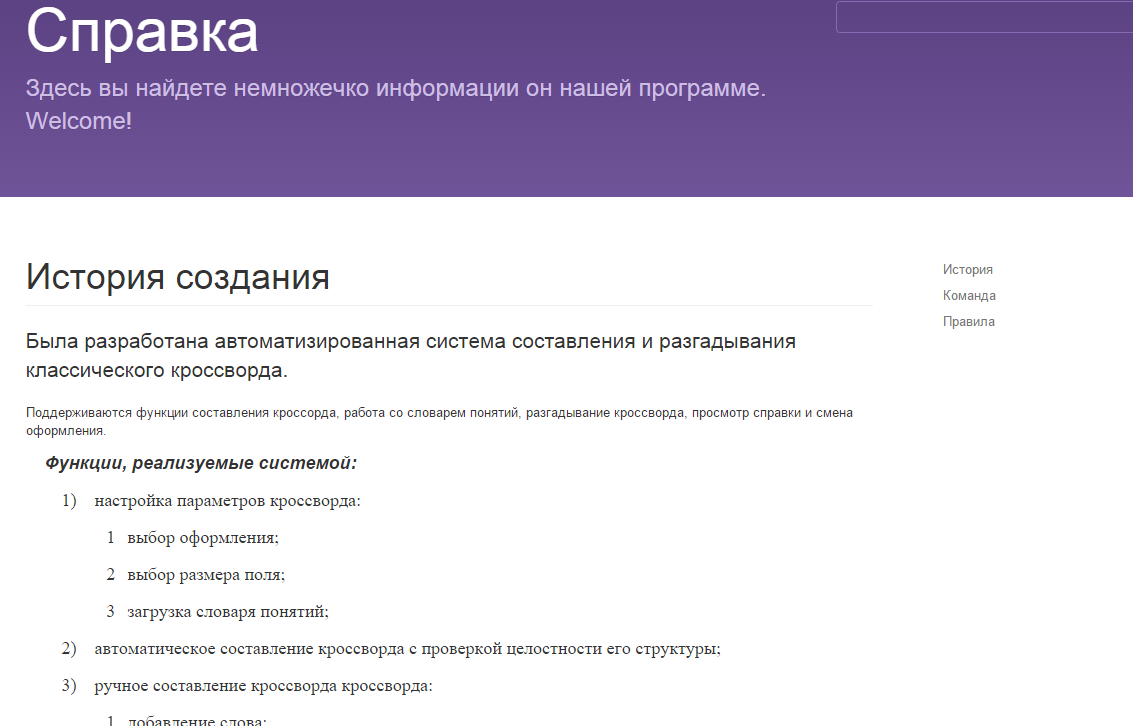
  
Рисунок 28 – Форма составление кроссворда

Пользователь может загрузить словарь, либо создать собственный. Также существует возможность удалить словарь, отсортировать список понятий по длине и ширине, перетащить выбранное слово на сетку кроссворда и, при необходимости, выбрать его ориентацию, добавить, изменить и удалить понятие, сохранить словарь, сохранить кроссворд и перейти в главное меню по нажатию кнопки «Главное меню».

При нажатии на кнопку «Разгадать» открывается форма разгадывания кроссворда изображенная на рисунке 29, с возможностью проверить слова по горизонтали, вертикали и кроссворд полностью. Также можно сохранить кроссворд и вернуться в главное меню по нажатию кнопки «Главное меню».

  
Рисунок 29 – Форма разгадывания кроссворда

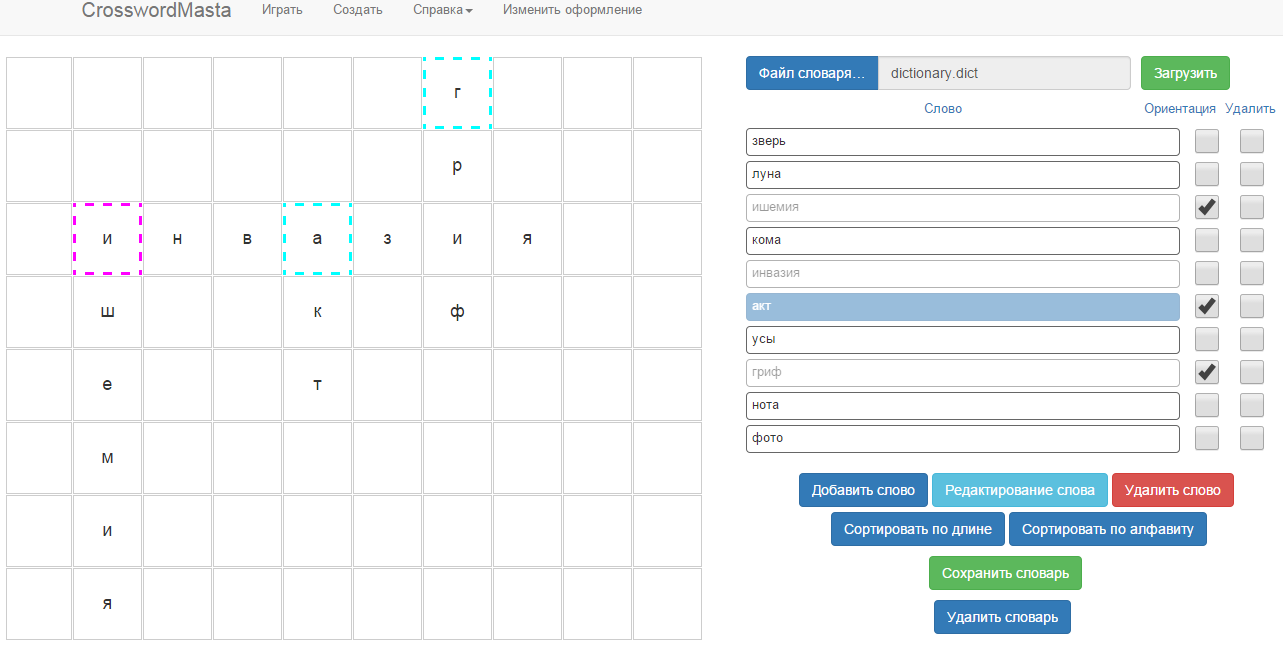
При нажатии на кнопку «Справка» открывается справочное окно, содержащее информацию об авторах и документацию. Данное окно изображено на рисунке 30. Вернуться в главное меню можно по нажатию кнопки «Главное меню».

  
Рисунок 30 – Форма справки

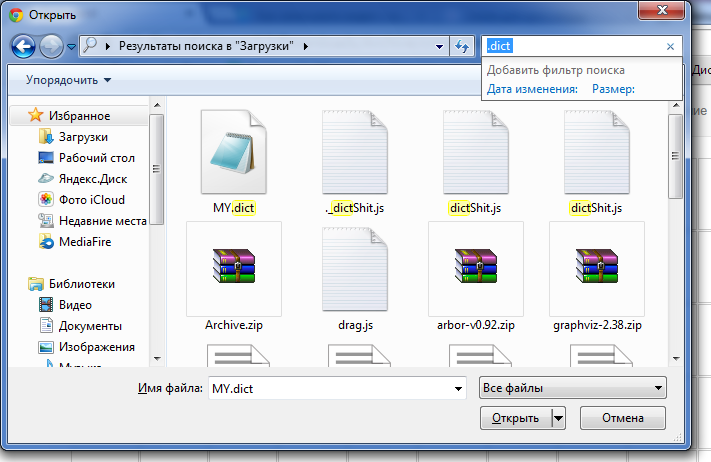
### 3.1.2 Описание контрольного примера

Рассмотрим в качестве контрольного примера ручное составление кроссворда и заполнение его словами из существующего словаря понятий.

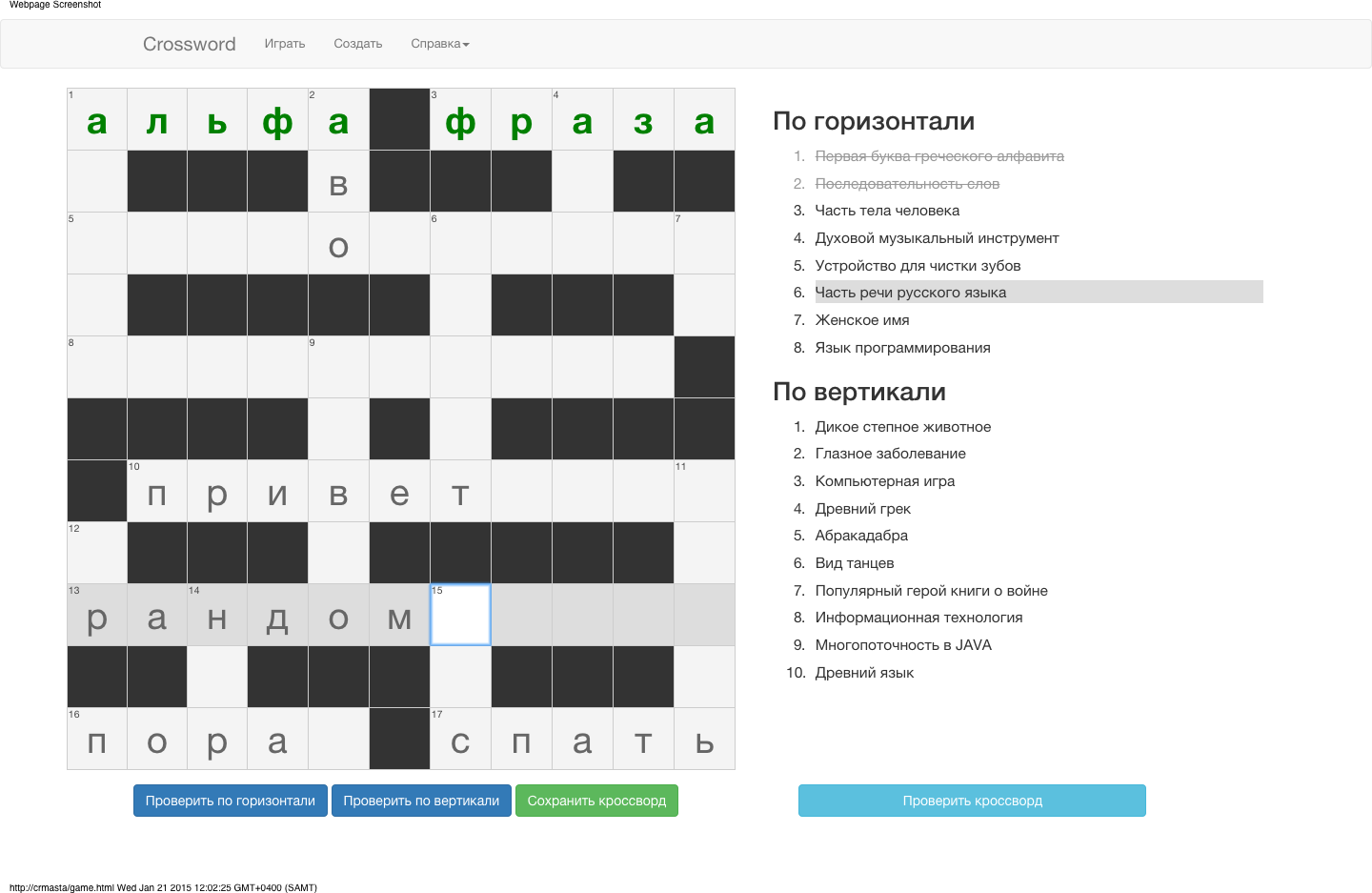
Для создания нового кроссворда необходима нажать на кнопку «Составить» в главном меню программы и выбрать пункт автоматически из всплывающего меню, после чего откроется форма составления кроссворда, изображенная на рисунке 31.

  
Рисунок 31 – Составление кроссворда

Здесь необходимо загрузить словарь понятий, нажав кнопку «Обзор». Перед Вами появится окно выбора кроссворда, изображенное на рисунке 32.

  
Рисунок 32 – Загрузка словаря

Далее можно добавить новое слово в словарь, заполнив поля понятие и определение. После этого пользователь должен нажать кнопку «Готово» для добавления слова в словарь. При желании, пользователь может изменить ориентацию слова, поставив галочку в пункте вертикаль, а также отсортировать загруженный словарь по длине и алфавиту. Слова из списка перетаскиваются на сетку, где и происходит составление кроссворда. В результате мы получаем кроссворд, изображенный на рисунке 33.

  
Рисунок 33 – Пример составленного кроссворда

Это кроссворд можно либо сохранить, нажав кнопку «Сохранить кроссворд», либо перейти на главную форму по нажатию кнопки «Главное меню». Завершить работу с приложением можно, нажав кнопку «Х».

## Выбор и обоснование комплекса технических средств

### 3.2.1 Расчет объема занимаемой памяти

Расчет объема занимаемой памяти включает в себя расчет необходимого объема оперативной памяти, а также памяти жесткого диска.

Необходимый объем внешней памяти рассчитывается по следующей формуле:

VЖД = VОС + VПР + VСПО,

где VОС – размер системных файлов операционной системы;

VПР – объем памяти, занимаемый файлами приложения;

VСПО – объем памяти, занимаемый всем необходимым сопутствующим программным обеспечением.

Операционная система Windows 7 требует VОС 9,43 Гб памяти. Файлы приложения занимают VПР ~2,28 Мб. Объем памяти, занимаемый всем необходимым сопутствующим программным обеспечением равен VСПО = 1 Гб.

Исходя из этих данных, рассчитаем требуемый объем памяти жесткого диска:

VЖД = 9,43 Гб + 2,28 Мб + 1 Гб = 10,4322 Гб.

Для расчета необходимого объема ОЗУ используется следующая формула [2]:

VОЗУ = VОС + VПР**.**

Операционная система занимает 1024 Мб, приложение – 401 Мб. Исходя из этих значений, рассчитаем требуемый объем оперативной памяти:

VОЗУ = 1024 Мб + 401 Мб = 1425 Мб.

### 3.2.2 Минимальные требования, предъявляемые к системе

Для функционирования системы необходимо:

1) персональный компьютер с ОС Windows XP или выше, либо Mac OS;

2) не менее 1425 Мб оперативной памяти;

3) 9,4398 Гб на жестком диске;

4) дисплей с разрешением не менее 1024 x 768 пикселей;

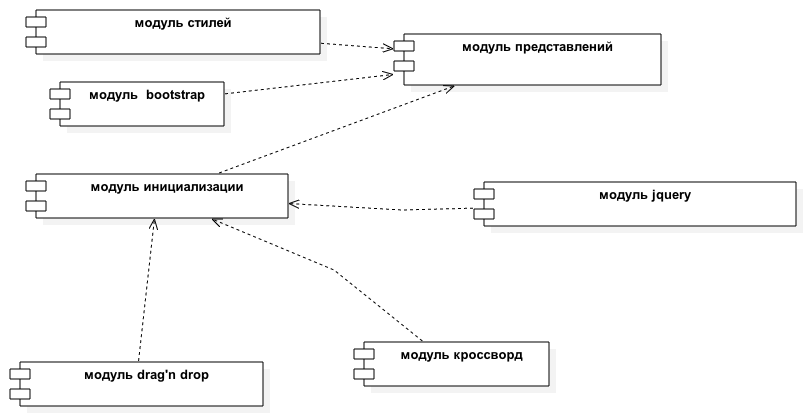
5) клавиатура или иное устройство ввода;

6) мышь или иное манипулирующее устройство.

## 3.3 Диаграммы реализации

### Диаграмма компонентов

Диаграмма компонентов описывает особенности физической реализации системы в момент перехода от логического представления к конкретной реализации системы. Диаграмма компонентов позволяет определить архитектуру разрабатываемой системы, установив зависимости между программными компонентами, в роли которых может выступать исходный, бинарный и исполняемый код. Основными графическими элементами диаграммы компонентов являются компоненты, интерфейсы и зависимости между ними [7]. На рисунке 34 приведена диаграмма компонентов разрабатываемой системы.

  
Рисунок 34 – Диаграмма компонентов

В программе представлены 7 связанных компонентов:

* модуль стилей, в котором хранятся все стили, используемые приложением;
* модуль bootstrap – настраиваемый HTML, CSS, и Javascript инструментарий для корректного отображения данных;
* модуль представления для манипулирования внешним видом сайта, а именно его контентом с различного типа данными;
* модуль инициализации, в котором происходит инициализация переменных;
* модуль jquery, включающий в себя множество полезных модулей, таких как модуль подсветки активного поля формы, модуль автоматического ресайза картинок и другие;
* модуль drag’n’drop, служащий для перетаскивания объекта;
* модуль кроссворд, в котором происходит работа с кроссвордом.

### Диаграмма развертывания

Диаграмма развертывания предназначена для визуализации элементов и компонентов программы, существующих лишь на этапе ее исполнения (runtime). При этом представляются только компоненты-экземпляры программы, являющиеся исполнимыми файлами или динамическими библиотеками. Те компоненты, которые не используются на этапе исполнения, на диаграмме развертывания не показываются [18].

Диаграмма развертывания содержит графические изображения процессоров, устройств, процессов и связей между ними. В отличие от диаграмм логического представления, диаграмма развертывания является единой для системы в целом, поскольку должна всецело отражать особенности ее реализации. Эта диаграмма, по сути, завершает процесс ООАП для конкретной программной системы и ее разработка, как правило, является последним этапом спецификации модели. Диаграмма развертывания разрабатываемой системы приведена на рисунке 35. В качестве операционной системы может быть выбрана Windows 7/8 либо Mac OS.

  
Рисунок 35– Диаграмма развертывания

# ЗАКЛЮЧЕНИЕ

В соответствии с техническим заданием был произведен анализ предметной области, реализовано программное и информационное обеспечение, подготовлены контрольные примеры и произведены тестирование и отладка автоматизированной системы составления и разгадывания классического кроссворда.

В первом разделе были проведены описание и анализ предметной области, обзор существующих систем аналогов, используемых для создания автоматизированных систем составления и разгадывания классического кроссворда. В конце раздела представлена постановка задачи – перечислены функций, которые должны быть реализованы в системе.

Во втором разделе происходит проектирование системы. Для него была разработана структурная схема автоматизированной системы составления и разгадывания классического кроссворда, описана спецификация требования, проведено прототипирование интерфейса пользователя системы и построены соответствующие проекту диаграммы. Также произведен выбор и обоснование комплекса программных средств.

В третьей части происходит реализация системы: разработка и описание интерфейса пользователя, описание контрольного примера. В данном разделе показано как работает автоматизированная система составления и разгадывания классического кроссворда.

Спроектированная система имеет удобный пользовательский интерфейс, легка и надежна в использовании, не требует серьезных аппаратных затрат. При выполнении работы использовалась методология объектно-ориентированного анализа и проектирования, кроме того применялась технология быстрой разработки приложений RAD и методология UML.

# СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Сайт кроссвордов и сканвордов "Шевельни мозговой извилиной!" [Электронный ресурс]. – http://crossword-best.ru/statji\_01.htm (дата обращения 20.09.2014 г.).
2. Методические указания – Создание UML диаграмм в Microsoft Visio 2010 [Электронный ресурс]. – http://www.twirpx.com/file/1375837/ (дата обращения 30.09.2014 г.).
3. Язык UML [Электронный ресурс]. – ru.wikipedia.org›UML (дата обращения 30.09.2014 г.).
4. Классический кроссворд [Электронный ресурс]. – ru.wikipedia.org›Кроссворд (дата обращения 09.09.2014 г.).
5. Cross: составь свой кроссворд онлайн [Электронный ресурс]. – http://cross.highcat.org/ru\_RU/ (дата обращения 01.10.2014 г.).
6. Онлайн кроссворды [Электрон ный ресурс]. – Bestcrosswords.ru (дата обращения 01.10.2014 г.).
7. Зеленко, Л.С. Методические указания к лабораторному практикуму по дисциплине «Программная инженерия» [Текст]/ Л.С. Зеленко. – Самара: СГАУ, 2012. – 51 с.
8. Объектно-ориентированный анализ и проектирование [ООА&П/OOA&D] [Электронный ресурс]. – http://ooad.asf.ru/students/lectures\_risp/lec025.aspx (дата обращения 10.10.2014 г.).
9. Академия Microsoft: Методы и средства инженерии программного обеспечения [Электронный ресурс]. – http://www.intuit.ru/studies/courses/  
   2190/237/lecture/6118 (дата обращения 13.10.2014 г.).

Интерфейс программ [Электронный ресурс]. – http://www.softandweb.ru/index.php/database-development/database-interface.html (дата обращения 20.11.2014 г.).

1. Спецификация качества программного средства [Электронный ресурс]. – http://life-prog.ru/1\_23466\_spetsifikatsiya-kachestva-programmnogo-sredstva.html (дата обращения 19.10.2014 г.).
2. Обзор методов моделирования промышленных процессов [Электронный ресурс] – http://www.rusnauka.com/1\_KAND\_2010/Economics/  
   8\_57811.doc.htm (дата обращения 19.10.2014 г.).
3. Язык UML Руководство пользователя [Электронный ресурс] – http://dit.isuct.ru/ivt/books/CASE/case11/ch2.htm (дата обращения 30.10.2014 г.).
4. Дополнительные обозначения языка UML для бизнес-моделирования [Электронный ресурс] – http://studopedia.net/2\_52764\_dopolnitelnie-oboznacheniya-yazika-UML-dlya-biznes-modelirovaniya.html (дата обращения 30.10.2014 г.).
5. Нотация и семантика языка UML [Электронный ресурс] – http://ointuit.ru/?p=326 (дата обращения 15.11.2014 г.).
6. Элементы графической нотации диаграммы состояний [Электронный ресурс] – http://www.intuit.ru/studies/courses/32/32/lecture/1016 (дата обращения 30.11.2014 г.).
7. Леоненков, А.В. Самоучитель UML [Электронный ресурс] – http://khpi-iip.mipk.kharkiv.edu/library/case/leon/gl6/gl6.html (дата обращения 25.12.2014 г.).
8. Пользовательский интерфейс [Электронный ресурс] – http://www.rae.ru/  
   monographs/141-4635 (дата обращения 10.12.2014 г.).
9. Проектирование программного обеспечения [Электронный ресурс] – http://habrahabr.ru/post/74330/ (дата обращения 25.12.2014 г.).

Приложение А  
Руководство пользователя

# А.1 Назначение системы

Данное приложение представляет собой программу, реализующую автоматизированную систему составления и разгадывания классического кроссворда.

Приложение предназначено для составления кроссворда с использованием собственного словаря понятий либо автоматической его генерации. Пользователь может производить разгадывание кроссворда и поиск ошибок.

# А.2 Условия работы системы

Для корректной работы системы необходимо наличие соответствующих программных и аппаратных средств.

Требования:

* персональный компьютер с ОС Windows XP или выше, либо Mac OS;
* не менее 1425 Мб оперативной памяти;
* 10,4322 Гб на жестком диске;
* дисплей с разрешением не менее 1024 x 768 пикселей;
* клавиатура или иное устройство ввода;
* мышь или иное манипулирующее устройство.

# А.3 Установка системы

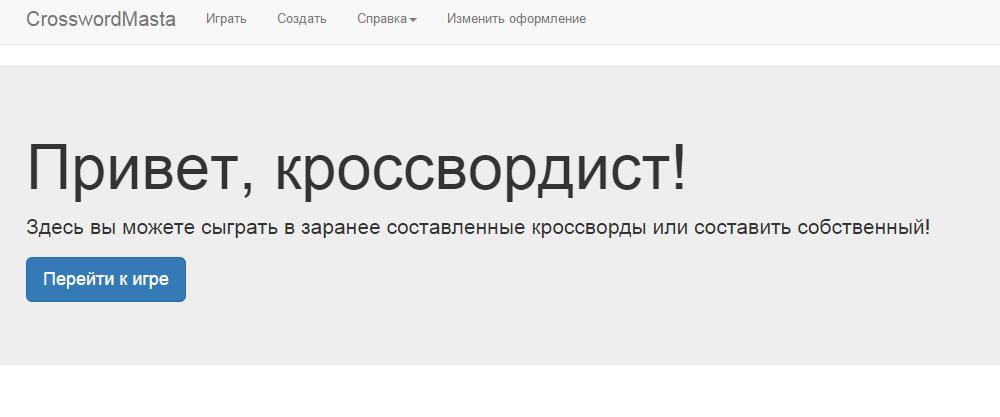
Запускаемым файлом системы является файл index.html. Его можно открыть в любом браузере, например Google Chrome, Safari или Internet Explorer.

А.4 Работа с системой

А.4.1 Конструирование кроссворда

После запуска приложения должно появиться главное окно, изображенное на рисунке А.1. В нем находится меню, состоящее из 3 кнопок-компонентов:

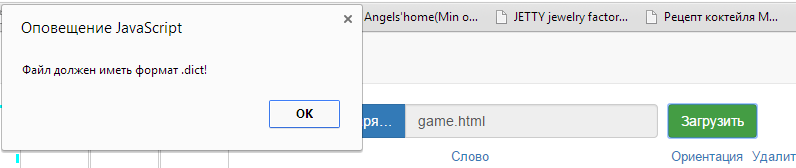
* составить;
* разгадать;
* справка.

  
Рисунок А.1 – Главная форма приложения

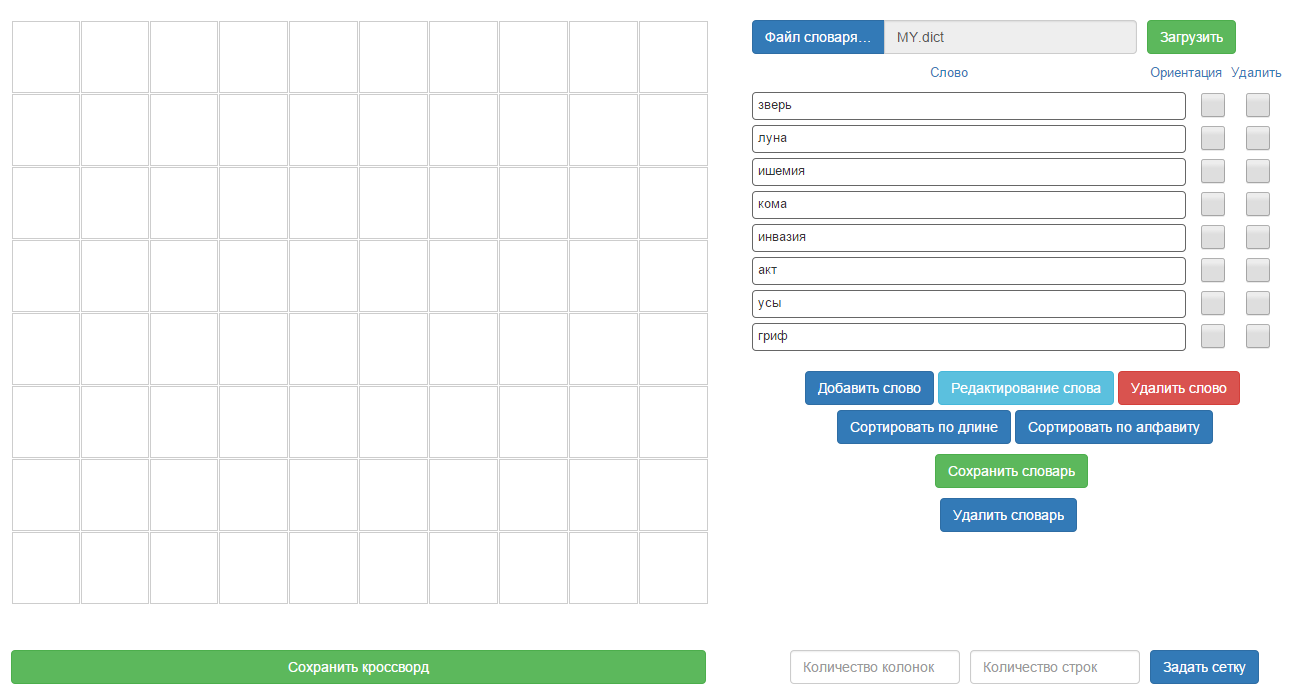
При нажатии на кнопку «X» основная форма закрывается.

А.4.2 Составление кроссворда

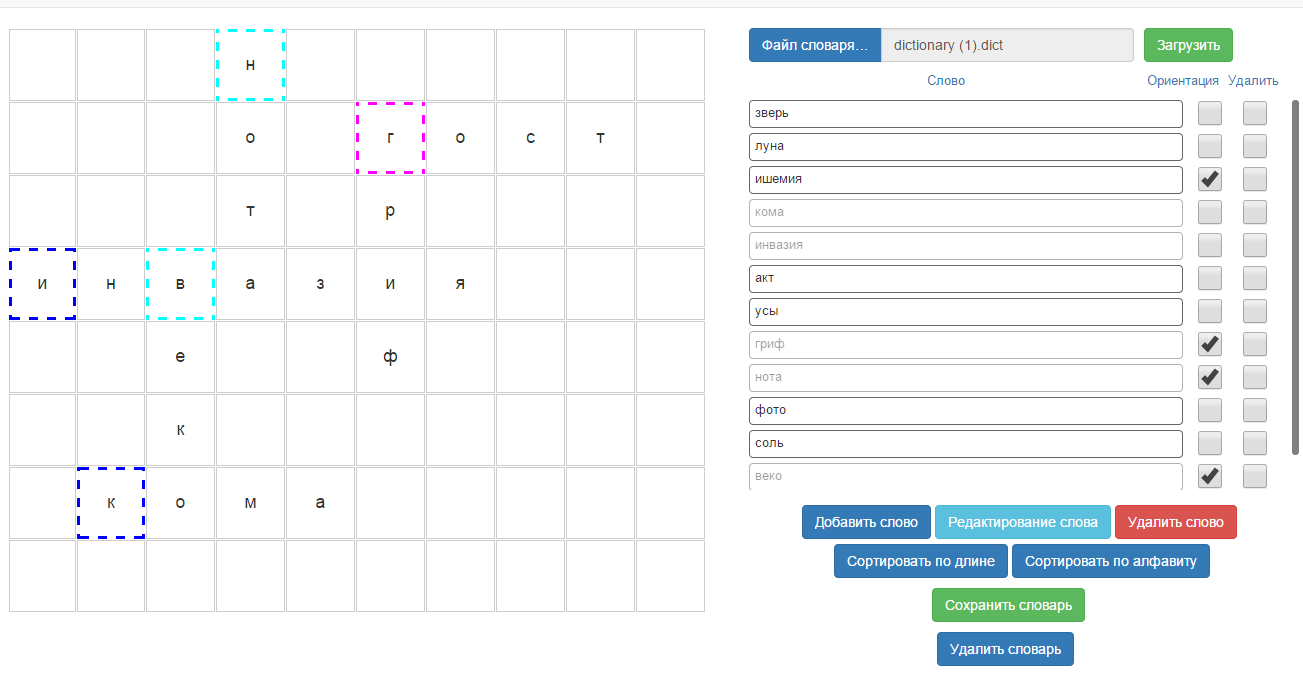
При нажатии кнопки «Составить» выводится всплывающее меню, предлагающее пользователю автоматическое либо ручное составление. При автоматическом составлении загружается последний созданный кроссворд, при ручном – открывается форма составления кроссворда. Рассмотрим более подробно ручное составление. Первоначально необходимо загрузить словарь понятий, нажав кнопку «Загрузить» и выбрав необходимый файл из списка. При выборе неверного формата файла появляется предупредительное сообщение, изображенное на рисунке А.2.

  
Рисунок А.2 – Сообщение об ошибке загрузки файла словаря

Форма составления кроссворда изображена на рисунке А.3.

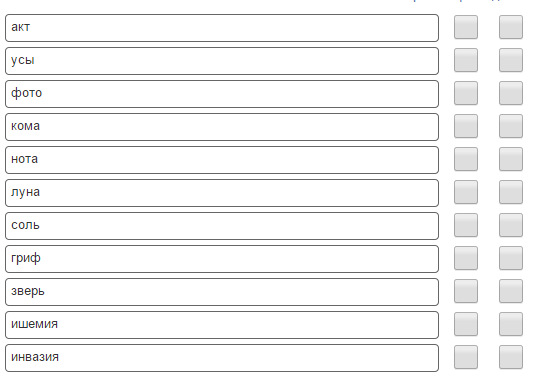
  
Рисунок А.3 – Форма составления кроссворда

Пример созданного нами кроссворда изображен на рисунке А.4. Ориентация слов указана в виде галочки в соответствующей ячейке.

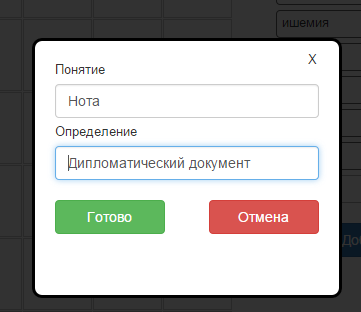
  
Рисунок А.4 – Пример кроссворда

Также пользователь может создать собственный словарь понятий, нажав на кнопку «Создать словарь». Помимо этого существуют возможности удалить словарь, отсортировать список понятий по длине и ширине, перетащить выбранное слово на сетку кроссворда и, при необходимости, выбрать его ориентацию, добавить, изменить и удалить понятие, сохранить словарь, сохранить кроссворд и перейти в главное меню по нажатию кнопки «Главное меню». Отсортированный по алфавиту и по длине словарь понятий изображен на рисунках А.5 и А.6.

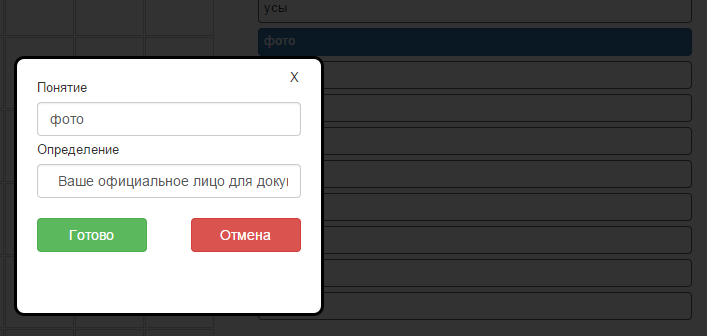
  
Рисунок А.5 – Сортировка словаря понятий по алфавиту

  
Рисунок А.6 – Сортировка словаря понятий по длине

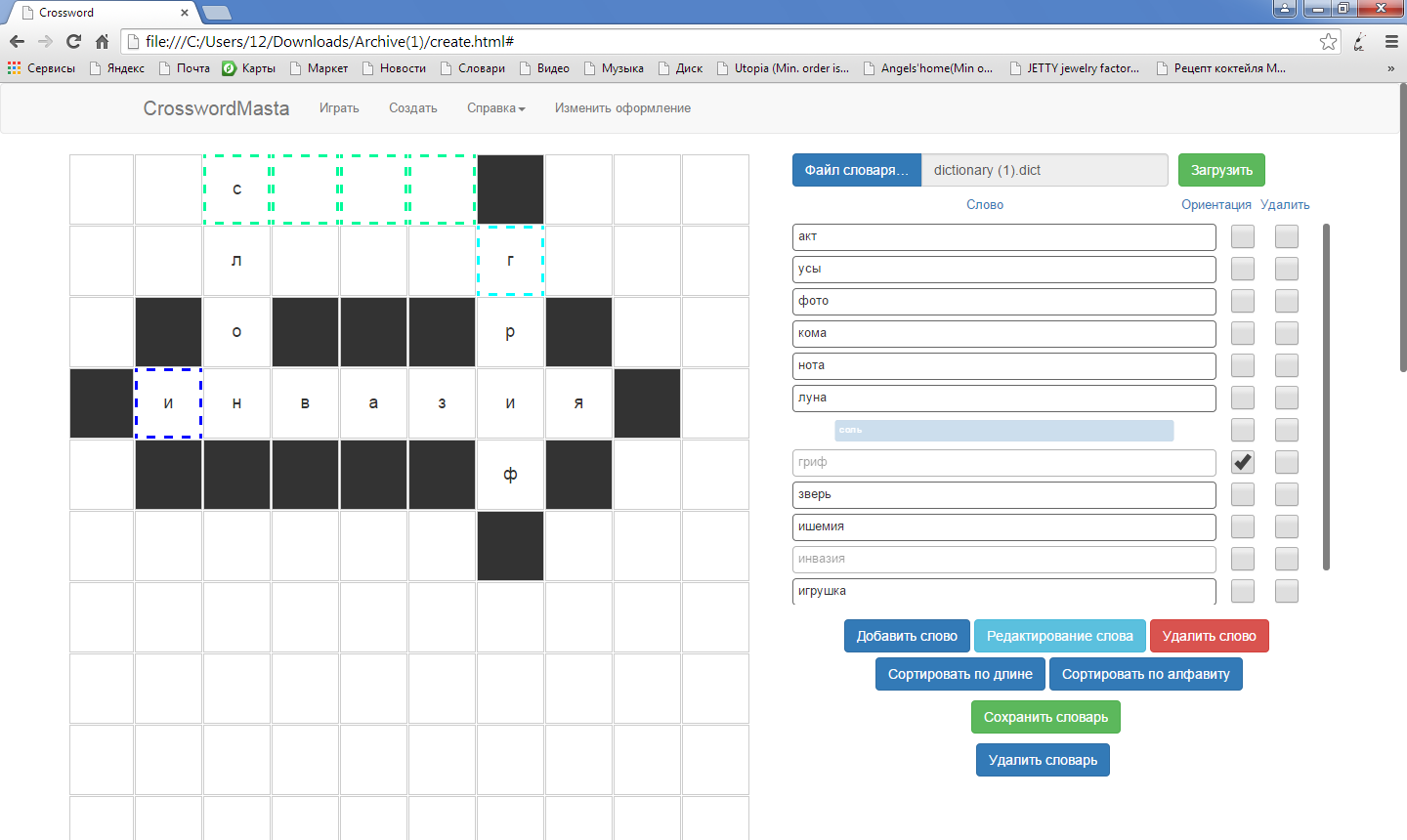
Добавление нового слова происходит по нажатию кнопки «Добавить слово». Появляется новое окно, в которое необходимо ввести понятие и определение и подтвердить добавление, нажав кнопку сохранить. Добавление слова в словарь понятий изображено на рисунке А.7.

  
Рисунок А.7 – Добавление слова в словарь понятий

Для редактирования слова необходимо выделить его в списке понятий и нажать кнопку «Редактировать». Перед пользователем появляется окно редактирования, изображенное не рисунке А.7, где можно изменить понятие или определение.

  
Рисунок А.8 – Редактирование слова

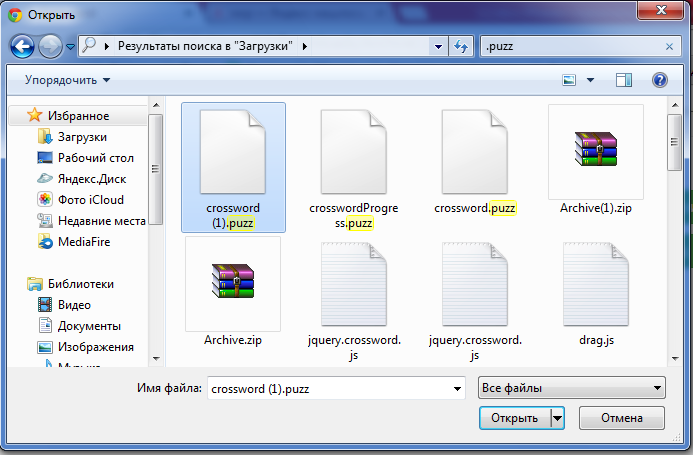
Перетаскивание слова на стеку происходит путем его нажатия по нему правой кнопкой мышки в списке понятий в и перетаскивания в необходимую клетку сетки кроссворда. Пример перетаскивания слова «соль» содержится на рисунке А.9.

  
Рисунок А.9 – Перетаскивание слова «соль»

Для сохранения кроссворда нужно нажать кнопку «Сохранить кроссворд». Для открытия главной формы приложения необходимо нажать конку «Главное меню».

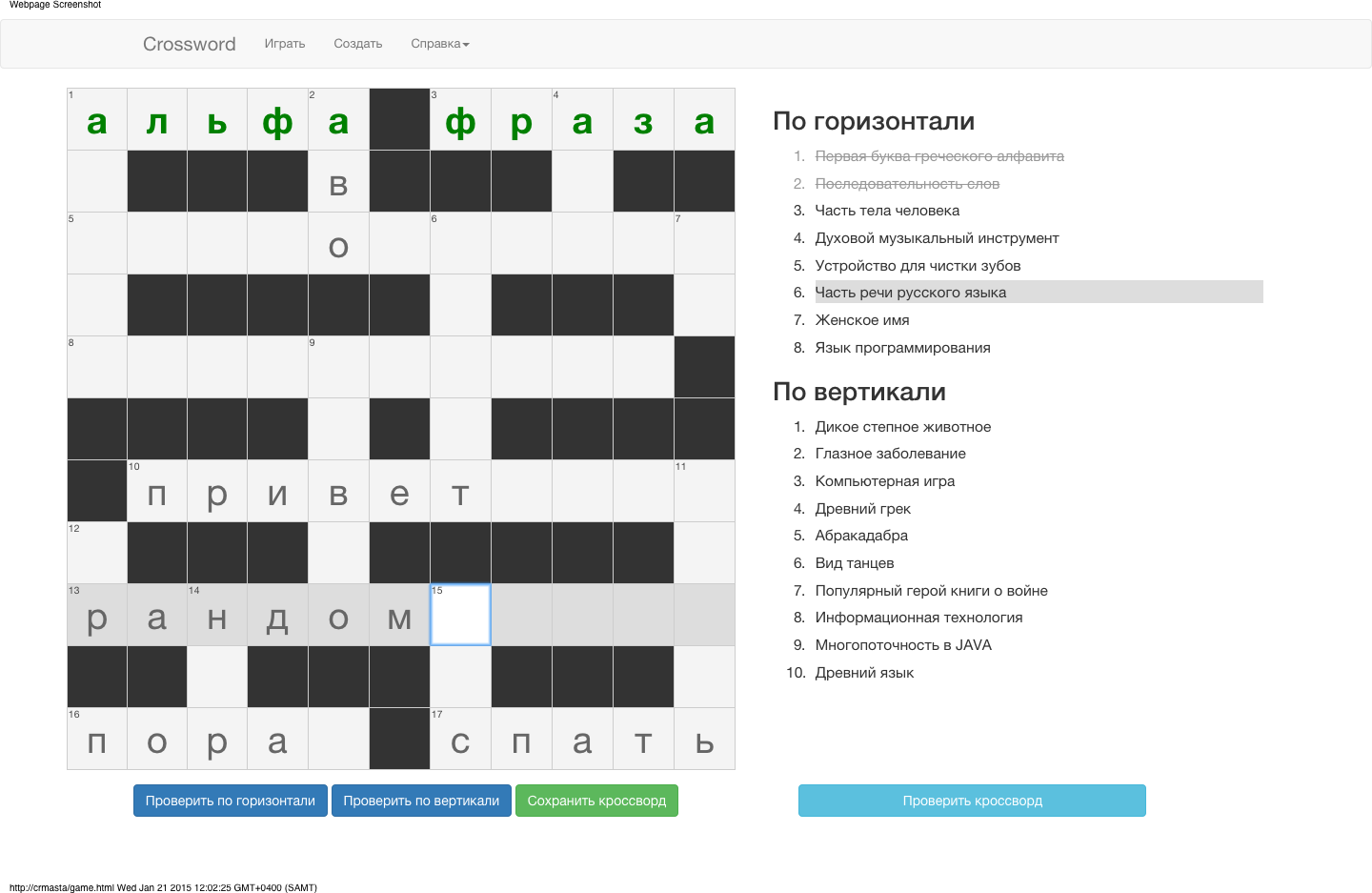
А.4.3 Разгадывание кроссворда

При нажатии на кнопку «Разгадать» открывается форма разгадывания кроссворда. Необходимо выбрать нужный файл и нажать кнопку «Загрузить». Загрузка файла кроссворда crossword (1).puzz изображена на рисунке А.10.

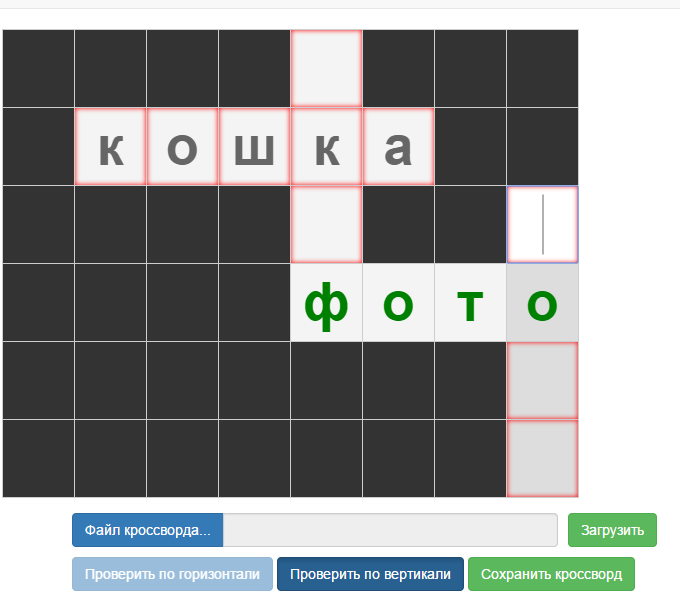
  
Рисунок А.10 – Загрузка файла кроссворда

Разгадывание кроссворда изображено на рисунке А.11, существует возможность проверить слова по горизонтали, вертикали и кроссворд полностью. Также можно сохранить кроссворд и вернуться в главное меню по нажатию кнопки «Главное меню».

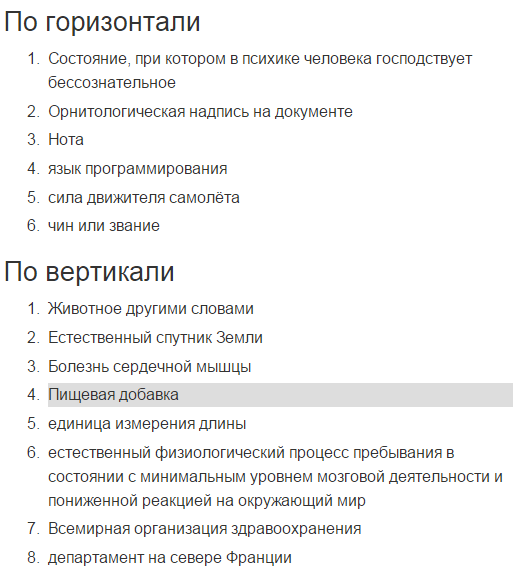
При разгадывании кроссворда приводятся определения слов, размещенных горизонтально либо вертикально на сетке. Чтобы вписать отгаданное слово, пользователю нужно выбрать соответствующие ячейки на сетке и напечатать туда известные буквы.

  
Рисунок А.11 – Разгадывание кроссворда

Проверить правильность разгаданного кроссворда можно нажав кнопку «Проверить кроссворд», также можно проверить отдельные слова, выбрав интересующую ячейку и нажав кнопку «Проверить по горизонтали/вертикали». Пример проверки правильности заполнения кроссворда изображен на рисунке А.12. Красным подсвечены слова, отгаданные неверно, зеленым – верно.

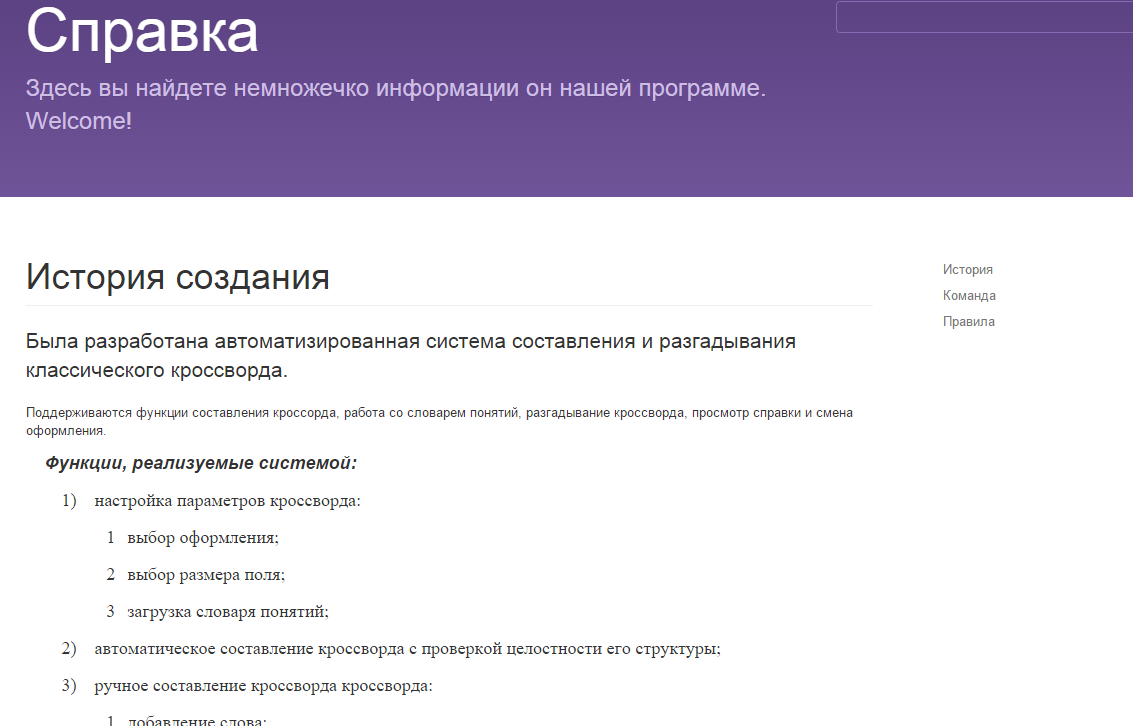
  
Рисунок А.12 – Проверка кроссворда

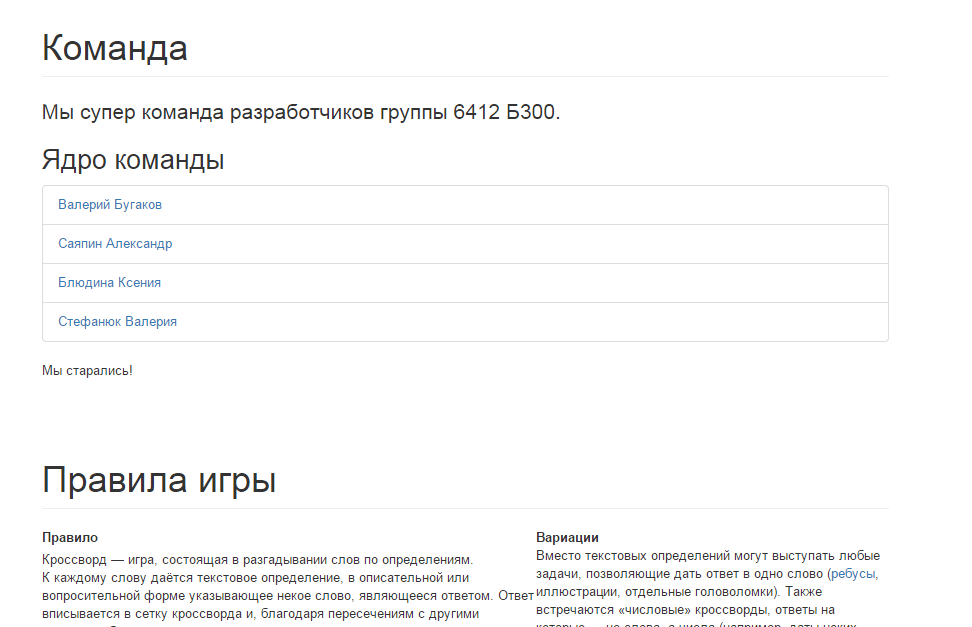
Отгадываемые понятия делятся на два раздела – по горизонтали и по вертикали. Отгадываемое понятие выделяется серым фоном. Пример изображен на рисунке А.13.

  
Рисунок А.13 – Список определений

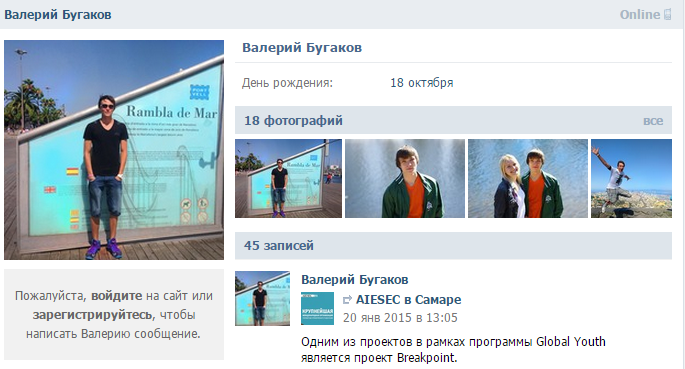
А.4.4 Справка по системе

При нажатии на кнопку «Справка» открывается справочное окно, содержащее информацию об авторах и документацию. Данное окно изображено на рисунках А.14 и А.15. Вернуться в главное меню можно по нажатию кнопки «Главное меню».

  
Рисунок А.14 – Справка

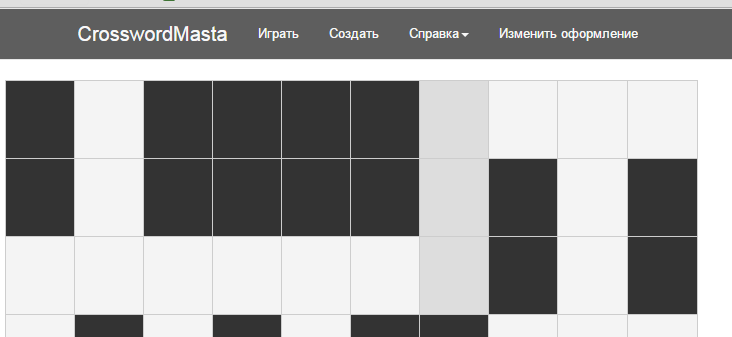
  
Рисунок А.15 – Информация о разработчиках

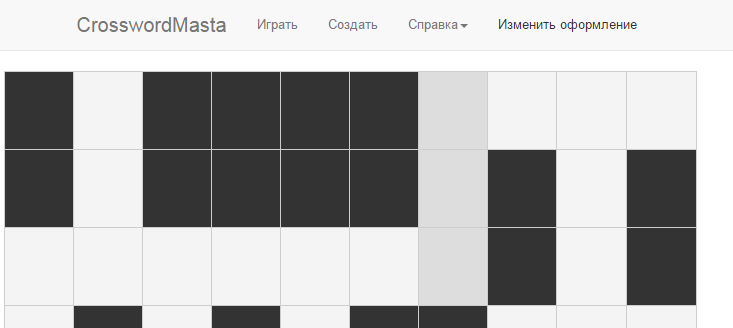
При необходимость можно связаться с разработчиком, нажав на необходимую ссылку в разделе «Ядро команды». Данное действие показано на рисунке А.16.

  
Рисунок А.16 – Страница разработчика

А.4.5 Смена оформления

При необходимости пользователь может сменить фон главного меню приложения нажав на кнопку «Сменить оформление». Темный и светлый варианты меню изображены на рисунках А.17 и А.18 соответственно.

  
Рисунок А.17 – Темное оформление

  
Рисунок А.18 – Светлое оформление

Приложение Б  
Листинг модулей программы

Файл bootstrap.min.js:

(function($) {

$(function() {

// provide crossword entries in an array of objects like the following example

// Position refers to the numerical order of an entry. Each position can have

// two entries: an across entry and a down entry

var puzzleData = [

{

clue: "Первая буква греческого алфавита",

answer: "альфа",

position: 1,

orientation: "across",

startx: 1,

starty: 1

},

{

clue: "Последовательность слов",

answer: "фраза",

position: 3,

orientation: "across",

startx: 7,

starty: 1

},

{

clue: "Created from a separation of charge",

answer: "capacitance",

position: 5,

orientation: "across",

startx: 1,

starty: 3

},

{

clue: "The speeds of engines without and accelaration",

answer: "idlespeeds",

position: 8,

orientation: "across",

startx: 1,

starty: 5

},

{

clue: "Complex resistances",

answer: "impedances",

position: 10,

orientation: "across",

startx: 2,

starty: 7

},

{

clue: "This device is used to step-up, step-down, and/or isolate",

answer: "transformer",

position: 13,

orientation: "across",

startx: 1,

starty: 9

},

{

clue: "Type of ray emitted frm the sun",

answer: "gamma",

position: 16,

orientation: "across",

startx: 1,

starty: 11

},

{

clue: "C programming language operator",

answer: "cysan",

position: 17,

orientation: "across",

startx: 7,

starty: 11

},

{

clue: "Defines the alpha-numeric characters that are typically associated with text used in programming",

answer: "ascii",

position: 1,

orientation: "down",

startx: 1,

starty: 1

},

{

clue: "Generally, if you go over 1kV per cm this happens",

answer: "arc",

position: 2,

orientation: "down",

startx: 5,

starty: 1

},

{

clue: "Control system strategy that tries to replicate the human through process (abbr.)",

answer: "ann",

position: 4,

orientation: "down",

startx: 9,

starty: 1

},

{

clue: "Greek variable that usually describes rotor positon",

answer: "theta",

position: 6,

orientation: "down",

startx: 7,

starty: 3

},

{

clue: "Electromagnetic (abbr.)",

answer: "em",

position: 7,

orientation: "down",

startx: 11,

starty: 3

},

{

clue: "No. 13 across does this to a voltage",

answer: "steps",

position: 9,

orientation: "down",

startx: 5,

starty: 5

},

{

clue: "Emits a lout wailing sound",

answer: "siren",

position: 11,

orientation: "down",

startx: 11,

starty: 7

},

{

clue: "Information technology (abbr.)",

answer: "it",

position: 12,

orientation: "down",

startx: 1,

starty: 8

},

{

clue: "Asynchronous transfer mode (abbr.)",

answer: "atm",

position: 14,

orientation: "down",

startx: 3,

starty: 9

},

{

clue: "Offset current control (abbr.)",

answer: "occ",

position: 15,

orientation: "down",

startx: 7,

starty: 9

}

];

var sampleArr = new Array(16);

$('#puzzle-wrapper').crossword(puzzleData, 'play');

$('#constructor-wrapper').createTable(10,8);

// $('#constructor-wrapper').crossword(sampleArr, 'create');

})

})(jQuery)

Файл create.js**:**

(function($) {

$(function() {

// provide crossword entries in an array of objects like the following example

// Position refers to the numerical order of an entry. Each position can have

// two entries: an across entry and a down entry

var puzzleData = [

{

clue: "Первая буква греческого алфавита",

answer: "альфа",

position: 1,

orientation: "across",

startx: 1,

starty: 1

},

{

clue: "Последовательность слов",

answer: "фраза",

position: 3,

orientation: "across",

startx: 7,

starty: 1

},

{

clue: "Created from a separation of charge",

answer: "capacitance",

position: 5,

orientation: "across",

startx: 1,

starty: 3

},

{

clue: "The speeds of engines without and accelaration",

answer: "idlespeeds",

position: 8,

orientation: "across",

startx: 1,

starty: 5

},

{

clue: "Complex resistances",

answer: "impedances",

position: 10,

orientation: "across",

startx: 2,

starty: 7

},

{

clue: "This device is used to step-up, step-down, and/or isolate",

answer: "transformer",

position: 13,

orientation: "across",

startx: 1,

starty: 9

},

{

clue: "Type of ray emitted frm the sun",

answer: "gamma",

position: 16,

orientation: "across",

startx: 1,

starty: 11

},

{

clue: "C programming language operator",

answer: "cysan",

position: 17,

orientation: "across",

startx: 7,

starty: 11

},

{

clue: "Defines the alpha-numeric characters that are typically associated with text used in programming",

answer: "ascii",

position: 1,

orientation: "down",

startx: 1,

starty: 1

},

{

clue: "Generally, if you go over 1kV per cm this happens",

answer: "arc",

position: 2,

orientation: "down",

startx: 5,

starty: 1

},

{

clue: "Control system strategy that tries to replicate the human through process (abbr.)",

answer: "ann",

position: 4,

orientation: "down",

startx: 9,

starty: 1

},

{

clue: "Greek variable that usually describes rotor positon",

answer: "theta",

position: 6,

orientation: "down",

startx: 7,

starty: 3

},

{

clue: "Electromagnetic (abbr.)",

answer: "em",

position: 7,

orientation: "down",

startx: 11,

starty: 3

},

{

clue: "No. 13 across does this to a voltage",

answer: "steps",

position: 9,

orientation: "down",

startx: 5,

starty: 5

},

{

clue: "Emits a lout wailing sound",

answer: "siren",

position: 11,

orientation: "down",

startx: 11,

starty: 7

},

{

clue: "Information technology (abbr.)",

answer: "it",

position: 12,

orientation: "down",

startx: 1,

starty: 8

},

{

clue: "Asynchronous transfer mode (abbr.)",

answer: "atm",

position: 14,

orientation: "down",

startx: 3,

starty: 9

},

{

clue: "Offset current control (abbr.)",

answer: "occ",

position: 15,

orientation: "down",

startx: 7,

starty: 9

}

];

var sampleArr = new Array(16);

$('#puzzle-wrapper').crossword(puzzleData, 'play');

$('#constructor-wrapper').createTable(10,8);

// $('#constructor-wrapper').crossword(sampleArr, 'create');

})

})(jQuery)

Файл dictShit.js:

(function($) {

$(function() {

// provide crossword entries in an array of objects like the following example

// Position refers to the numerical order of an entry. Each position can have

// two entries: an across entry and a down entry

var puzzleData = [

{

clue: "Первая буква греческого алфавита",

answer: "альфа",

position: 1,

orientation: "across",

startx: 1,

starty: 1

},

{

clue: "Последовательность слов",

answer: "фраза",

position: 3,

orientation: "across",

startx: 7,

starty: 1

},

{

clue: "Created from a separation of charge",

answer: "capacitance",

position: 5,

orientation: "across",

startx: 1,

starty: 3

},

{

clue: "The speeds of engines without and accelaration",

answer: "idlespeeds",

position: 8,

orientation: "across",

startx: 1,

starty: 5

},

{

clue: "Complex resistances",

answer: "impedances",

position: 10,

orientation: "across",

startx: 2,

starty: 7

},

{

clue: "This device is used to step-up, step-down, and/or isolate",

answer: "transformer",

position: 13,

orientation: "across",

startx: 1,

starty: 9

},

{

clue: "Type of ray emitted frm the sun",

answer: "gamma",

position: 16,

orientation: "across",

startx: 1,

starty: 11

},

{

clue: "C programming language operator",

answer: "cysan",

position: 17,

orientation: "across",

startx: 7,

starty: 11

},

{

clue: "Defines the alpha-numeric characters that are typically associated with text used in programming",

answer: "ascii",

position: 1,

orientation: "down",

startx: 1,

starty: 1

},

{

clue: "Generally, if you go over 1kV per cm this happens",

answer: "arc",

position: 2,

orientation: "down",

startx: 5,

starty: 1

},

{

clue: "Control system strategy that tries to replicate the human through process (abbr.)",

answer: "ann",

position: 4,

orientation: "down",

startx: 9,

starty: 1

},

{

clue: "Greek variable that usually describes rotor positon",

answer: "theta",

position: 6,

orientation: "down",

startx: 7,

starty: 3

},

{

clue: "Electromagnetic (abbr.)",

answer: "em",

position: 7,

orientation: "down",

startx: 11,

starty: 3

},

{

clue: "No. 13 across does this to a voltage",

answer: "steps",

position: 9,

orientation: "down",

startx: 5,

starty: 5

},

{

clue: "Emits a lout wailing sound",

answer: "siren",

position: 11,

orientation: "down",

startx: 11,

starty: 7

},

{

clue: "Information technology (abbr.)",

answer: "it",

position: 12,

orientation: "down",

startx: 1,

starty: 8

},

{

clue: "Asynchronous transfer mode (abbr.)",

answer: "atm",

position: 14,

orientation: "down",

startx: 3,

starty: 9

},

{

clue: "Offset current control (abbr.)",

answer: "occ",

position: 15,

orientation: "down",

startx: 7,

starty: 9

}

];

var sampleArr = new Array(16);

$('#puzzle-wrapper').crossword(puzzleData, 'play');

$('#constructor-wrapper').createTable(10,8);

// $('#constructor-wrapper').crossword(sampleArr, 'create');

})

})(jQuery)

Файл drag.js:

$(function() {

var dragSrcEl = null;

function handleDragStart(e) {

// Target (this) element is the source node.

// this.style.opacity = '0.4';

dragSrcEl = this;

e.dataTransfer.effectAllowed = 'move';

e.dataTransfer.setData('text/html', this.innerHTML);

$(this).addClass('moving');

}

function handleDragOver(e) {

if (e.preventDefault) {

e.preventDefault(); // Necessary. Allows us to drop.

}

// this.style.opacity = '1'; // this / e.target is the source node.

e.dataTransfer.dropEffect = 'move'; // See the section on the DataTransfer object.

dataString = e.dataTransfer.getData('text/html')

ori = wordsUtil.getDraggingOri(dataString);

symbArr = wordsUtil.getDraggingSymbols(dataString);

availableCells = getAvalableCells(this, ori);

for(var i = 0; i < symbArr.length; i++) {

$(availableCells[i]).addClass('help-highlight')

}

console.log('over')

return false;

}

function handleDragEnter(e) {

// this / e.target is the current hover target.

this.classList.add('over');

console.log('enter')

}

function handleDragLeave(e) {

this.classList.remove('over'); // this / e.target is previous target element.

// $('#puzzle td').removeClass('help-highlight')

dataString = e.dataTransfer.getData('text/html')

ori = wordsUtil.getDraggingOri(dataString);

symbArr = wordsUtil.getDraggingSymbols(dataString);

availableCells = getAvalableCells(this, ori);

for(var i = 0; i < symbArr.length; i++) {

$(availableCells[i]).removeClass('help-highlight')

}

console.log('leave')

}

function handleDrop(e) {

// this/e.target is current target element.

if (e.stopPropagation) {

e.stopPropagation(); // Stops some browsers from redirecting.

}

targetEl = this;

// Don't do anything if dropping the same column we're dragging.

if (dragSrcEl != this) {

if ($(this).parents('#constructor-wrapper').length > 0) {

dataString = e.dataTransfer.getData('text/html')

ori = wordsUtil.getDraggingOri(dataString);

symbArr = wordsUtil.getDraggingSymbols(dataString);

pasteWord(symbArr, targetEl, ori);

}

}

return false;

}

function handleDragEnd(e) {

// this/e.target is the source node.

setTimeout(function() { $('.error').removeClass('error'); }, 1000);

[].forEach.call(cols, function (col) {

col.classList.remove('over');

col.classList.remove('moving');

});

$('#puzzle td').removeClass('over moving help-highlight');

}

function pasteWord(symbArr, targetEl, ori) {

availableCells = getAvalableCells(targetEl, ori);

if (availableCells.length < symbArr.length) {

alert("U can't place it, biatch!");

} else {

if (checkCells(availableCells, symbArr)) {

addAttrs(availableCells[0], symbArr, ori)

$(dragSrcEl).attr('draggable', null)

for(var i = 0; i < symbArr.length; i++) {

if ($(availableCells[i]).html() != "") {

availableCells[i].setAttribute('data-cross', true)

$(availableCells[i]).data('cross', true)

}

$(availableCells[i]).html(symbArr[i]);

}

}

}

}

function addAttrs(cell, symbols, ori) {

if ( ori == 'hor' ) {

$(cell).attr('data-word-hor', symbols.join(''));

$(cell).data('word-hor', symbols.join(''));

} else {

$(cell).attr('data-word-vert', symbols.join(''));

$(cell).data('word-vert', symbols.join(''));

}

// $(cell).attr('data-word', symbols.join(''));

// $(cell).attr('data-ori', ori);

$(dragSrcEl).attr('data-placed', cell.getAttribute('data-coords'));

$(dragSrcEl).data('placed', cell.getAttribute('data-coords'));

$(cell).addClass('word-' + ori)

}

function checkCells(cells, symbols) {

check = true;

for(var i = 0; i < symbols.length; i++) {

if ($(cells[i]).html() != '') {

if ($(cells[i]).html() != symbols[i]) {

$(cells[i]).addClass('error');

check = false;

}

}

}

return check;

}

function getAvalableCells(targetEl, ori) {

if (ori == 'hor') {

arr = [targetEl];

$(targetEl).nextAll().each(function(index, el) {

arr.push(el);

});

return arr;

} else if (ori == 'vert') {

arr = [];

x = $(targetEl).data('coords').split(',')[0]

y = $(targetEl).data('coords').split(',')[1]

for (var i = y; i <= Y\_MAX; i++) {

arr.push($('#constructor-wrapper td[data-coords="' + x + ',' + i + '"]')[0]);

};

return arr;

}

}

function makeItDraggble(cols){

[].forEach.call(cols, function(col) {

col.setAttribute('draggable', 'true');

col.addEventListener('dragstart', handleDragStart, false);

// col.addEventListener('dragenter', handleDragEnter, false)

// col.addEventListener('dragover', handleDragOver, false);

// col.addEventListener('dragleave', handleDragLeave, false);

// col.addEventListener('drop', handleDrop, false);

col.addEventListener('dragend', handleDragEnd, false);

});

}

var cols = document.querySelectorAll('#columns .column');

makeItDraggble(cols)

var cells = $('#puzzle td');

[].forEach.call(cells, function(cell) {

cell.setAttribute('draggable', 'true');

cell.addEventListener('dragstart', handleDragStart, false);

cell.addEventListener('dragenter', handleDragEnter, false)

cell.addEventListener('dragover', handleDragOver, false);

cell.addEventListener('dragleave', handleDragLeave, false);

cell.addEventListener('drop', handleDrop, false);

cell.addEventListener('dragend', handleDragEnd, false);

});

$('#ololo').on('click', 'button', function(event) {

event.preventDefault();

var allSelect = document.getElementsByClassName('selectWord');

var allWord = document.getElementsByClassName('column');

for (var j = 0; j < allWord.length; j++){

var select = allSelect[j];

if (select.checked){

ololo = allWord[j];

ololo.parentNode.setAttribute('draggable', true);

x = $(ololo).closest('.column').data('placed').split(',')[0];

y = $(ololo).closest('.column').data('placed').split(',')[1];

targetCell = $('td[data-coords="' + x + ',' + y + '"]');

ori = $(ololo).prev().data('ori');

if ( ori == 'hor' ) {

word = targetCell.data('word-hor');

targetCell.removeClass('word-hor')

targetCell.data('word-hor', null);

targetCell.attr('data-word-hor', null);

}

if ( ori == 'vert' ) {

word = targetCell.data('word-vert');

targetCell.removeClass('word-vert')

targetCell.data('word-vert', null);

targetCell.attr('data-word-vert', null);

}

cells = getAvalableCells(targetCell, ori);

for (var i = 0; i < word.length; i++) {

if ($(cells[i]).data('cross') != true) {

$(cells[i]).html('');

targetCell.removeClass('word-starts')

} else {

$(cells[i]).attr('data-cross', null);

$(cells[i]).data('cross', null)

}

}

}

}

});

});

var wordsUtil = {

getDraggingSymbols: function(dataString) {

valString = wordsUtil.getDraggingEl(dataString).innerHTML;

return valString.split('');

},

getDraggingOri: function(dataString) {

return wordsUtil.getDraggingEl(dataString).getAttribute('data-ori');

},

getDraggingEl: function(dataString) {

if ( dataString.indexOf('meta http-equiv="Content-Type"') > -1 ) {

fullString = dataString.replace(dataString.substring(0, 66), "");

return $.parseHTML(fullString)[0];

} else {

return $.parseHTML(dataString)[0];

}

}

}

Файл jquery.crossword.js:

(function($){

$.fn.crossword = function(entryData, createOrPlay) {

/\*

"light" refers to a white box - or an input

DEV NOTES:

- activePosition and activeClueIndex are the primary vars that set the ui whenever there's an interaction

- 'Entry' is a puzzler term used to describe the group of letter inputs representing a word solution

- This puzzle isn't designed to securely hide answerers. A user can see answerers in the js source

- An xhr provision can be added later to hit an endpoint on keyup to check the answerer

- The ordering of the array of problems doesn't matter. The position & orientation properties is enough information

- Puzzle authors must provide a starting x,y coordinates for each entry

- Entry orientation must be provided in lieu of provided ending x,y coordinates (script could be adjust to use ending x,y coords)

- Answers are best provided in lower-case, and can NOT have spaces - will add support for that later

\*/

var puzz = {}; // put data array in object literal to namespace it into safety

puzz.data = entryData;

// initialize some variables

var tbl = ['<table id="puzzle">'],

puzzEl = this,

clues = $('#puzzle-clues'),

clueLiEls,

coords,

entryCount = puzz.data.length,

entries = [],

rows = [],

cols = [],

solved = [],

tabindex,

$actives,

activePosition = 0,

activeClueIndex = 0,

currOri,

targetInput,

mode = 'interacting',

solvedToggle = false,

z = 0;

var puzInit = {

init: function() {

currOri = 'across'; // app's init orientation could move to config object

// Reorder the problems array ascending by POSITION

puzz.data.sort(function(a,b) {

return a.position - b.position;

});

// Set keyup handlers for the 'entry' inputs that will be added presently

puzzEl.delegate('input', 'keyup', function(e){

mode = 'interacting';

// need to figure out orientation up front, before we attempt to highlight an entry

switch(e.which) {

case 39: // right

case 37: // left

currOri = 'across';

break;

case 38: // up

case 40: // down

currOri = 'down';

break;

default:

break;

}

if ( e.keyCode === 9) {

return false;

} else if (

e.keyCode === 37 ||

e.keyCode === 38 ||

e.keyCode === 39 ||

e.keyCode === 40 ||

e.keyCode === 8 ||

e.keyCode === 46 ) {

if (e.keyCode === 8 || e.keyCode === 46) {

// remove and delete buttons

currOri === 'across' ? nav.nextPrevNav(e, 37) : nav.nextPrevNav(e, 38);

} else {

nav.nextPrevNav(e);

}

e.preventDefault();

return false;

} else {

console.log('input keyup: ' + solvedToggle);

puzInit.checkAnswer(e);

}

e.preventDefault();

return false;

});

// tab navigation handler setup

puzzEl.delegate('input', 'keydown', function(e) {

// tab button

if ( e.keyCode === 9) {

mode = "setting ui";

if (solvedToggle) solvedToggle = false;

//puzInit.checkAnswer(e)

nav.updateByEntry(e);

} else {

return true;

}

e.preventDefault();

});

// tab navigation handler setup

puzzEl.delegate('input', 'click', function(e) {

mode = "setting ui";

if (solvedToggle) solvedToggle = false;

console.log('input click: '+solvedToggle);

nav.updateByEntry(e);

e.preventDefault();

});

// click/tab clues 'navigation' handler setup

clues.delegate('li', 'click', function(e) {

mode = 'setting ui';

if (!e.keyCode) {

nav.updateByNav(e);

}

e.preventDefault();

});

// highlight the letter in selected 'light' - better ux than making user highlight letter with second action

puzzEl.delegate('#puzzle', 'click', function(e) {

$(e.target).focus();

$(e.target).select();

});

// DELETE FOR BG

puzInit.calcCoords();

// Puzzle clues added to DOM in calcCoords(), so now immediately put mouse focus on first clue

clueLiEls = '#puzzle-clues li';

$('#' + currOri + ' li' ).eq(0).addClass('clues-active').focus();

// DELETE FOR BG

puzInit.buildTable();

puzInit.buildEntries();

},

/\*

- Given beginning coordinates, calculate all coordinates for entries, puts them into entries array

- Builds clue markup and puts screen focus on the first one

\*/

calcCoords: function() {

/\*

Calculate all puzzle entry coordinates, put into entries array

\*/

for (var i = 0, p = entryCount; i < p; ++i) {

// set up array of coordinates for each problem

entries.push(i);

entries[i] = [];

for (var x=0, j = puzz.data[i].answer.length; x < j; ++x) {

entries[i].push(x);

coords = puzz.data[i].orientation === 'across' ? "" + puzz.data[i].startx++ + "," + puzz.data[i].starty + "" : "" + puzz.data[i].startx + "," + puzz.data[i].starty++ + "" ;

entries[i][x] = coords;

}

// while we're in here, add clues to DOM!

$('#' + puzz.data[i].orientation).append('<li tabindex="1" data-position="' + i + '">' + puzz.data[i].clue + '</li>');

}

// Calculate rows/cols by finding max coords of each entry, then picking the highest

for (var i = 0, p = entryCount; i < p; ++i) {

for (var x=0; x < entries[i].length; x++) {

cols.push(entries[i][x].split(',')[0]);

rows.push(entries[i][x].split(',')[1]);

};

}

// Actual sizes of puzzle table

rows = Math.max.apply(Math, rows) + "";

cols = Math.max.apply(Math, cols) + "";

},

/\*

Build the table markup

- adds [data-coords] to each <td> cell

\*/

buildTable: function() {

for (var i = 1; i <= rows; ++i) {

tbl.push("<tr>");

for (var x = 1; x <= cols; ++x) {

tbl.push('<td data-coords="' + x + ',' + i + '"></td>');

};

tbl.push("</tr>");

};

tbl.push("</table>");

puzzEl.append(tbl.join('')); // array toStrign() without commas

},

/\*

Builds entries into table

- Adds entry class(es) to <td> cells

- Adds tabindexes to <inputs>

\*/

buildEntries: function() {

var puzzCells = '#puzzle td',

light,

$groupedLights,

hasOffset = false,

positionOffset = entryCount - puzz.data[puzz.data.length-1].position; // diff. between total ENTRIES and highest POSITIONS

for (var x = 1, p = entryCount; x <= p; ++x) {

// get array of letters

var letters = puzz.data[x-1].answer.split('');

for (var i = 0; i < entries[x-1].length; ++i) {

light = $(puzzCells +'[data-coords="' + entries[x-1][i] + '"]');

// check if POSITION property of the entry on current go-round is same as previous.

// If so, it means there's an across & down entry for the position.

// Therefore you need to subtract the offset when applying the entry class.

if(x > 1 ){

if (puzz.data[x-1].position === puzz.data[x-2].position) {

hasOffset = true;

};

}

if($(light).empty()){

$(light)

.addClass('entry-' + (hasOffset ? x - positionOffset : x) + ' position-' + (x-1) )

.append('<input maxlength="1" val="" type="text" tabindex="-1" />');

}

};

};

// Put entry number in first 'light' of each entry, skipping it if already present

for (var i=1, p = entryCount; i < p; ++i) {

$groupedLights = $('.entry-' + i);

if(!$('.entry-' + i +':eq(0) span').length){

$groupedLights.eq(0)

.append('<span>' + puzz.data[i].position + '</span>');

}

}

util.highlightEntry();

util.highlightClue();

$('.active').eq(0).focus();

$('.active').eq(0).select();

},

/\*

- Checks current entry input group value against answer

- If not complete, auto-selects next input for user

\*/

checkAnswer: function(e) {

var valToCheck, currVal;

util.getActivePositionFromClassGroup($(e.target));

valToCheck = puzz.data[activePosition].answer.toLowerCase();

currVal = $('.position-' + activePosition + ' input')

.map(function() {

return $(this)

.val()

.toLowerCase();

})

.get()

.join('');

//console.log(currVal + " " + valToCheck);

if(valToCheck === currVal){

$('.active')

.addClass('done')

.removeClass('active');

$('.clues-active').addClass('clue-done');

solved.push(valToCheck);

solvedToggle = true;

return;

}

currOri === 'across' ? nav.nextPrevNav(e, 39) : nav.nextPrevNav(e, 40);

//z++;

//console.log(z);

//console.log('checkAnswer() solvedToggle: '+solvedToggle);

}

}; // end puzInit object

var nav = {

nextPrevNav: function(e, override) {

var len = $actives.length,

struck = override ? override : e.which,

el = $(e.target),

p = el.parent(),

ps = el.parents(),

selector;

util.getActivePositionFromClassGroup(el);

util.highlightEntry();

util.highlightClue();

$('.current').removeClass('current');

selector = '.position-' + activePosition + ' input';

//console.log('nextPrevNav activePosition & struck: '+ activePosition + ' '+struck);

// move input focus/select to 'next' input

switch(struck) {

case 39:

p

.next()

.find('input')

.addClass('current')

.select();

break;

case 37:

p

.prev()

.find('input')

.addClass('current')

.select();

break;

case 40:

ps

.next('tr')

.find(selector)

.addClass('current')

.select();

break;

case 38:

ps

.prev('tr')

.find(selector)

.addClass('current')

.select();

break;

default:

break;

}

},

updateByNav: function(e) {

var target;

$('.clues-active').removeClass('clues-active');

$('.active').removeClass('active');

$('.current').removeClass('current');

currIndex = 0;

target = e.target;

activePosition = $(e.target).data('position');

util.highlightEntry();

util.highlightClue();

$('.active').eq(0).focus();

$('.active').eq(0).select();

$('.active').eq(0).addClass('current');

// store orientation for 'smart' auto-selecting next input

currOri = $('.clues-active').parent('ol').prop('id');

activeClueIndex = $(clueLiEls).index(e.target);

//console.log('updateByNav() activeClueIndex: '+activeClueIndex);

},

// Sets activePosition var and adds active class to current entry

updateByEntry: function(e, next) {

var classes, next, clue, e1Ori, e2Ori, e1Cell, e2Cell;

if(e.keyCode === 9 || next){

// handle tabbing through problems, which keys off clues and requires different handling

activeClueIndex = activeClueIndex === clueLiEls.length-1 ? 0 : ++activeClueIndex;

$('.clues-active').removeClass('.clues-active');

next = $(clueLiEls[activeClueIndex]);

currOri = next.parent().prop('id');

activePosition = $(next).data('position');

// skips over already-solved problems

util.getSkips(activeClueIndex);

activePosition = $(clueLiEls[activeClueIndex]).data('position');

} else {

activeClueIndex = activeClueIndex === clueLiEls.length-1 ? 0 : ++activeClueIndex;

util.getActivePositionFromClassGroup(e.target);

clue = $(clueLiEls + '[data-position=' + activePosition + ']');

activeClueIndex = $(clueLiEls).index(clue);

currOri = clue.parent().prop('id');

}

util.highlightEntry();

util.highlightClue();

//$actives.eq(0).addClass('current');

//console.log('nav.updateByEntry() reports activePosition as: '+activePosition);

}

}; // end nav object

var util = {

highlightEntry: function() {

// this routine needs to be smarter because it doesn't need to fire every time, only

// when activePosition changes

$actives = $('.active');

$actives.removeClass('active');

$actives = $('.position-' + activePosition + ' input').addClass('active');

$actives.eq(0).focus();

$actives.eq(0).select();

},

highlightClue: function() {

var clue;

$('.clues-active').removeClass('clues-active');

$(clueLiEls + '[data-position=' + activePosition + ']').addClass('clues-active');

if (mode === 'interacting') {

clue = $(clueLiEls + '[data-position=' + activePosition + ']');

activeClueIndex = $(clueLiEls).index(clue);

};

},

getClasses: function(light, type) {

if (!light.length) return false;

var classes = $(light).prop('class').split(' '),

classLen = classes.length,

positions = [];

// pluck out just the position classes

for(var i=0; i < classLen; ++i){

if (!classes[i].indexOf(type) ) {

positions.push(classes[i]);

}

}

return positions;

},

getActivePositionFromClassGroup: function(el){

classes = util.getClasses($(el).parent(), 'position');

if(classes.length > 1){

// get orientation for each reported position

e1Ori = $(clueLiEls + '[data-position=' + classes[0].split('-')[1] + ']').parent().prop('id');

e2Ori = $(clueLiEls + '[data-position=' + classes[1].split('-')[1] + ']').parent().prop('id');

// test if clicked input is first in series. If so, and it intersects with

// entry of opposite orientation, switch to select this one instead

e1Cell = $('.position-' + classes[0].split('-')[1] + ' input').index(el);

e2Cell = $('.position-' + classes[1].split('-')[1] + ' input').index(el);

if(mode === "setting ui"){

currOri = e1Cell === 0 ? e1Ori : e2Ori; // change orientation if cell clicked was first in a entry of opposite direction

}

if(e1Ori === currOri){

activePosition = classes[0].split('-')[1];

} else if(e2Ori === currOri){

activePosition = classes[1].split('-')[1];

} else {

activePosition = classes[0].split('-')[1];

}

} else {

activePosition = classes[0].split('-')[1];

}

console.log('getActivePositionFromClassGroup activePosition: ' + activePosition);

},

checkSolved: function(valToCheck) {

for (var i=0, s=solved.length; i < s; i++) {

if(valToCheck === solved[i]){

return true;

}

}

},

getSkips: function(position) {

if ($(clueLiEls[position]).hasClass('clue-done')){

activeClueIndex = position === clueLiEls.length-1 ? 0 : ++activeClueIndex;

util.getSkips(activeClueIndex);

} else {

return false;

}

}

}; // end util object

if (createOrPlay == "play") {

// append clues markup after puzzle wrapper div

// This should be moved into a configuration object

this.after('<div id="puzzle-clues"><h2>По горизонтали</h2><ol id="across"></ol><h2>По вертикали</h2><ol id="down"></ol></div>');

puzInit.init();

} else if (createOrPlay == "create") {

// create actions

// puzInit.init();

}

}

})(jQuery);

Файл script.js:

(function($) {

$(function() {

// provide crossword entries in an array of objects like the following example

// Position refers to the numerical order of an entry. Each position can have

// two entries: an across entry and a down entry

var puzzleData = [

{

clue: "Первая буква греческого алфавита",

answer: "альфа",

position: 1,

orientation: "across",

startx: 1,

starty: 1

},

{

clue: "Последовательность слов",

answer: "фраза",

position: 3,

orientation: "across",

startx: 7,

starty: 1

},

{

clue: "Created from a separation of charge",

answer: "capacitance",

position: 5,

orientation: "across",

startx: 1,

starty: 3

},

{

clue: "The speeds of engines without and accelaration",

answer: "idlespeeds",

position: 8,

orientation: "across",

startx: 1,

starty: 5

},

{

clue: "Complex resistances",

answer: "impedances",

position: 10,

orientation: "across",

startx: 2,

starty: 7

},

{

clue: "This device is used to step-up, step-down, and/or isolate",

answer: "transformer",

position: 13,

orientation: "across",

startx: 1,

starty: 9

},

{

clue: "Type of ray emitted frm the sun",

answer: "gamma",

position: 16,

orientation: "across",

startx: 1,

starty: 11

},

{

clue: "C programming language operator",

answer: "cysan",

position: 17,

orientation: "across",

startx: 7,

starty: 11

},

{

clue: "Defines the alpha-numeric characters that are typically associated with text used in programming",

answer: "ascii",

position: 1,

orientation: "down",

startx: 1,

starty: 1

},

{

clue: "Generally, if you go over 1kV per cm this happens",

answer: "arc",

position: 2,

orientation: "down",

startx: 5,

starty: 1

},

{

clue: "Control system strategy that tries to replicate the human through process (abbr.)",

answer: "ann",

position: 4,

orientation: "down",

startx: 9,

starty: 1

},

{

clue: "Greek variable that usually describes rotor positon",

answer: "theta",

position: 6,

orientation: "down",

startx: 7,

starty: 3

},

{

clue: "Electromagnetic (abbr.)",

answer: "em",

position: 7,

orientation: "down",

startx: 11,

starty: 3

},

{

clue: "No. 13 across does this to a voltage",

answer: "steps",

position: 9,

orientation: "down",

startx: 5,

starty: 5

},

{

clue: "Emits a lout wailing sound",

answer: "siren",

position: 11,

orientation: "down",

startx: 11,

starty: 7

},

{

clue: "Information technology (abbr.)",

answer: "it",

position: 12,

orientation: "down",

startx: 1,

starty: 8

},

{

clue: "Asynchronous transfer mode (abbr.)",

answer: "atm",

position: 14,

orientation: "down",

startx: 3,

starty: 9

},

{

clue: "Offset current control (abbr.)",

answer: "occ",

position: 15,

orientation: "down",

startx: 7,

starty: 9

}

];

var sampleArr = new Array(16);

$('#puzzle-wrapper').crossword(puzzleData, 'play');

$('#constructor-wrapper').createTable(10,8);

// $('#constructor-wrapper').crossword(sampleArr, 'create');

})

})(jQuery)