МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования «Самарский национальный исследовательский университет имени академика С.П. Королева (Самарский университет)»

Институт информатики, математики и электроники

Факультет информатики

Кафедра программных систем

**ОТЧЕТ**  
  
 к лабораторному практикуму

по дисциплине «Технологии программирования»

по теме «Автоматизированная система составления и

разгадывания кроссворда»

Студент В.А. Байрамов

Студент В.Д. Мавлютов

Студент Н.Д. Перевозчиков

Студент С.В. Федоров

Руководитель Л.С. Зеленко

Самара 2018

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования «Самарский национальный исследовательский университет имени академика С.П. Королева (Самарский университет)»

Институт информатики, математики и электроники

Факультет информатики

Кафедра программных систем

**ЗАДАНИЕ**

на лабораторный практикум по дисциплине

«Технологии программирования»

студентам группы № 6402-090301D

В.А. Байрамов  
В.Д. Мавлютов   
Н.Д. Перевозчиков  
С.В. Федоров

1. **Тема проекта:** «Автоматизированная система составления и разгадывания линейного кроссворда по выбранной теме»
2. **Исходные данные к проекту:** см. приложение к заданию
3. **Перечень вопросов, подлежащих разработке:**
   1. Произвести анализ предметной области: изучить основные принципы составления кроссвордов, изучить алгоритмы генерации кроссвордов
   2. Выполнить обзор существующих систем-аналогов
   3. Разработать информационно-логический проект системы
   4. Разработать и реализовать программное и информационное обеспечение, провести его тестирование и отладку .
   5. Оформить документацию проекта
   6. Подготовить презентацию по разработанной системе
4. **Перечень графических разработок**
   1. Структурная схема системы
   2. Диаграмма классов, диаграмма модулей (компонентов)
   3. 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 | 09.10.2018 |  |
| 2 | Описание и анализ предметной области (1 раздел) | 10 | 16.10.2018 |  |
| 3 | Проектирование системы (2 раздел) | 30 | 04.12.2018 |  |
| 3.1 | Разработка структурной схемы системы | 5 | 16.10.2018 |  |
| 3.2 | Разработка функциональной спецификации системы | 10 | 06.11.2018 |  |
| 3.3 | Разработка прототипов экранных форм | 10 | 06.11.2018 |  |
| 3.4 | Разработка основных алгоритмов | 5 | 04.12.2018 |  |
| 4 | Реализация проекта, разработка контрольных примеров. Предъявление реализации руководителю (3 раздел) | 45 | 11.12.2018 |  |
| 5 | Корректировка проекта и оформление документации проекта. Защита проекта с представлением презентации | 10 | 25.12.2018 |  |

Задание принял  
 к исполнению В.А. Байрамов

В.Д. Мавлютов

Н.Д. Перевозчиков

С.В. Федоров

**ПРИЛОЖЕНИЕ  
к заданию на лабораторный практикум**студентам группы № 6402-090301D  
В.А. Байрамов

В.Д. Мавлютов

Н.Д. Перевозчиков

С.В. Федоров

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

## количество ролей пользователей – 2;

## минимальная ширина кроссворда – 7 символов;

## максимальная ширина кроссворда – 30 символов;

## минимальная высота кроссворда – 7 символов;

## максимальная высота кроссворда – 30 символов;

1. минимальная длина слова – 3 символа;
2. максимальная длина слова – 20 символов;

## количество режимов создания – 2;

## минимальное количество букв в пересечении – 2;

1. максимальное количество букв в пересечении – 9;

## количество видов сортировки словаря понятий – 2;

## минимальное количество подсказок – 1;

## максимальное количество подсказок – 10% от количества слов;

## язык записи понятий – русский.

1. **Требования к информационному обеспечению:**
2. информационное обеспечение разрабатывается на основе следующего источника:
   * Описание структуры кроссворда [Электронный ресурс]. URL: ru.wikipedia.org/wiki/Кроссворд (дата обращения: 01.10.2018);
3. структура словаря понятий (понятие и его определение располагаются в одной строке, разделены пробелом);
4. словари понятий хранятся в текстовых файлах формата \*.dict;
5. кроссворды хранятся в файлах, структура файла определяется в процессе проектирования.
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 проекта.
11. **Требования к программному обеспечению:**
12. тип операционной системы – Windows 7/8 и выше;
13. язык программирования – С#;
14. среда программирования – Visual Studio 2017;
15. среда проектирования – StarUML 2.8.0.
16. **Общие требования к проектируемой системе:**

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

1. *общесистемные функции:*
   * авторизация пользователя в системе (ввод логина/пароля);
   * аутентификация пользователя в системе, настройка интерфейса пользователя на заданную роль;
   * визуализация процессов работы с кроссвордом;
   * автоматическое составление кроссворда по заданным параметрам;
   * контроль количества взятых подсказок;
   * проверка правильности разгадывания кроссворда;
   * выдача справочной информации о системе;
2. *функции администратора:*
   * настройка параметров кроссворда при создании:
3. задание высоты;
4. задание ширины;
5. подключение словаря понятий;
   * составление/редактирование кроссворда:
6. добавление слова;
7. удаление слова;
8. изменение ориентации слова;
9. фильтрация словаря понятий по маске;
10. сортировка словаря;
    * сохранение кроссворда в файл заданной структуры;
    * загрузка кроссворда из файла;
    * работа со словарями понятий:
11. добавление понятия;
12. удаление понятия;
13. изменение понятия;
14. проверка дублирования понятий;
15. проверка языка записи понятий;
16. сортировка словаря по выбранному критерию;
17. поиск по заданной маске;
18. загрузка словаря из файла;
19. сохранение словаря из файла;
20. создание нового словаря понятий;
21. *функции пользователя:*
    * загрузка кроссворда из файла;
    * разгадывание кроссворда с организацией системы подсказок:
22. выбор слова;
23. вписывание/удаление/изменение буквы;
24. взятие подсказки;
    * сохранение кроссворда в файл.

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

1. режим работы – диалоговый;
2. время автоматической генерации кроссворда – не более 50 с;
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 – 15-35°С;
6. влажность воздуха – 45-75%.

Руководитель   
проекта Л.С. Зеленко

Задание принял  
к исполнению В.А. Байрамов

В.Д. Мавлютов

Н.Д. Перевозчиков

С.В. Федоров

**РЕФЕРАТ**

Пояснительная записка 35 с, 14 рисунков, 5 таблиц[[1]](#footnote-1), 12 источников,  
2 приложения.

Графическая часть: ??? слайдов презентации PowerPoint.

ДЕРЕВО ПОИСКА, ГЕНЕРАТОР КРОССВОРДОВ, ГОЛОВОЛОМКА, СЛОВАРЬ ПОНЯТИЙ, РАЗГАДЫВАНИЕ, ВИЗУАЛИЗАЦИЯ

Во время лабораторного практикума разработаны алгоритмы и соответствующая им программа, позволяющая выполнять автоматическую генерацию классического кроссворда. Задания (понятие и его расшифровка) хранятся в текстовом файле и могут дополняться вручную (с использованием текстового редактора) или внутри программы, при этом ограничений на длину словаря не существует. Тема кроссворда выбирается пользователем в соответствии с содержанием словаря заданий. 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 С# в среде Visual Studio 2017 и функционирует под управлением операционной системы Windows 7/8/10.

**СОДЕРЖАНИЕ**

[1 Описание и анализ предметной области 13](#_Toc527475490)

[1.1 Классификация кроссвордов 13](#_Toc527475491)

[1.2 Правила построения кроссворда 15](#_Toc527475492)

[1.3 Описание систем-аналогов 16](#_Toc527475493)

[1.3.1 Приложение «Сrosswordus» 16](#_Toc527475494)

[1.3.2 Приложение «Puzzlecup» 17](#_Toc527475495)

[1.3.3 Приложение «Hot Potatoes» 18](#_Toc527475496)

[1.4 Диаграмма объектов предметной области 20](#_Toc527475497)

[1.5 Постановка задачи 20](#_Toc527475498)

[2 Проектирование системы 24](#_Toc527475499)

[2.1 Структурная схема системы 24](#_Toc527475500)

[2.2 Спецификация системы 24](#_Toc527475501)

[2.2.1 Функциональная спецификация 24](#_Toc527475502)

[2.2.2 Спецификация качества 27](#_Toc527475503)

[2.2.3 Перечень исключительных ситуаций 27](#_Toc527475504)

[2.3 Разработка прототипа интерфейса пользователя системы 28](#_Toc527475505)

[2.4 Разработка структур данных и классов 29](#_Toc527475506)

[2.5 Логическая модель данных (при необходимости) 29](#_Toc527475507)

[2.6 Выбор и обоснование алгоритмов обработки данных /Разработка и описание алгоритмов обработки данных 29](#_Toc527475508)

[2.7 Выбор и обоснование комплекса программных средств 29](#_Toc527475509)

[2.7.1 Выбор языка программирования и среды разработки 31](#_Toc527475510)

[2.7.2 Выбор операционной системы 31](#_Toc527475511)

[2.7.3 Выбор среды программирования 31](#_Toc527475512)

[2.7.4 Выбор системы управления базами данных (при необходимости) 31](#_Toc527475513)

[3 Реализация системы 32](#_Toc527475514)

[3.1 Разработка и описание интерфейса пользователя 32](#_Toc527475515)

[3.1.1 Разработка и описание пользовательского меню 32](#_Toc527475516)

[3.1.2 Описание тестового примера 32](#_Toc527475517)

[3.2 Реализация классов и структур данных 32](#_Toc527475518)

[3.3 Физическая модель данных (при необходимости) 32](#_Toc527475519)

[3.4 Реализация и описание модулей программы 32](#_Toc527475520)

[3.5 Выбор и обоснование комплекса технических средств 32](#_Toc527475521)

[3.5.1 Расчет объема занимаемой памяти 32](#_Toc527475522)

[Расчет объема внешней памяти 32](#_Toc527475523)

[Расчет объема ОЗУ 33](#_Toc527475524)

[3.5.2 Минимальные требования, предъявляемые к системе 34](#_Toc527475525)

[ПРИЛОЖЕНИЕ А Руководство пользователя 39](#_Toc527475526)

[А.1 Назначение системы 39](#_Toc527475527)

[А.2 Условия работы системы 39](#_Toc527475528)

[А.3 Установка системы 39](#_Toc527475529)

[А.4 Работа с системой 39](#_Toc527475530)

[А.4.1 Работа с системой в режиме администратора (если необходимо) 40](#_Toc527475531)

[Вход в систему (авторизация) 40](#_Toc527475532)

[А.4.2 Работа с системой в режиме пользователя 40](#_Toc527475533)

[Вход в систему (авторизация) 40](#_Toc527475534)

[Вход в систему (регистрация) 40](#_Toc527475535)

[Настройка параметров кроссворда 40](#_Toc527475536)

[ПРИЛОЖЕНИЕ Б Листинг модулей программы 41](#_Toc527475537)

ВВЕДЕНИЕ

Кроссворд – головоломка, представляющая собой переплетение рядов клеточек, которые заполняются словами по заданным значениям. Обычно значения слов задаются описательно под этой схемой, сначала значения слов, которые должны получиться по горизонтали, затем – по вертикали [1].

Исследователями встречались находки, похожие на кроссворд, датированные еще 1-4 вв. н. э. По одной из версий, прототипы современных кроссвордов появились еще в XIX веке. Самый первый дошедший до нас кроссворд был опубликован в 1875 году в номере журнала «Святой Николас» в Нью-Йорке.

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.

В общем случае кроссворды, как и другие интеллектуальные головоломки, – это полезное увлечение. Они расширяют кругозор, тренируют память и учат выуживать из нее необходимые сведения.

Для того чтобы повысить эрудицию в конкретных областях деятельности, можно выбирать тематические кроссворды. Подобные головоломки также используются в обучении, например, иностранным языкам – они помогают запоминать новые слова.

Перед авторами поставлена задача – разработать автоматизированную систему «Классический кроссворд» с возможностью составления кроссворда в автоматическом и ручном режимах и его разгадывания.

Разработка системы будет вестись по технологии RAD (Rapid Application Development), которая поддерживается методологией объектно-ориентированной декомпозиции предметной области и принципами структурного проектирования. Модель RAD очень хорошо подходит к разработке учебных программ, т.к. включает в себя три составляющие [2]:

1. Небольшую команду программистов (от 2 до 4 человек);
2. Короткий, но тщательно проработанный производственный график (от 2 до 4 мес.);
3. Повторяющийся цикл, при котором разработчики по мере того, как приложение начинает обретать форму, запрашивают и реализуют в продукте требования, полученные через взаимодействие с заказчиком.

При проектировании системы буде использоваться методология ООАП (Объектно-ориентированный анализ и проектирование). Объектно-ориентированный анализ – это методология анализа предметной области, при которой требования к проектируемой системе воспринимаются с точки зрения классов и объектов, выявленных в предметной области [3].

Объектно-ориентированное проектирование – это методология проектирования, соединяющая в себе процесс объектной декомпозиции и приемы представления логической и физической, а также статической и динамической моделей проектируемой системы.

* + 1. Описание и анализ предметной области

Предметная область – часть реального мира, которая имеет существенное значение или непосредственное отношение к процессу функционирования программы [4]. Предметная область включает в себя только те объекты и взаимосвязи между ними, которые необходимы для описания требований и условий решения некоторой задачи.

* 1. Классификация кроссвордов

Кроссворд – игра-задача, в которой фигура из рядов пустых клеток заполняется перекрещивающимися словами со значениями, заданными по условию игры [5].

Существует большое количество видов кроссвордов. Самые распространенные из них [6]:

* Классические кроссворды. В отличие от сканвордов, в классических кроссвордах все вопросы представлены в виде отдельного пронумерованного списка, разделенного на две части: «по горизонтали», по «вертикали». Ответы необходимо вносить в поле кроссворда в соответствующие блоки из пустых ячеек слева направо и сверху вниз, как правило, в именительном падеже в единственном числе, за исключением слов, правописание которых предусматривает только множественное число. На рисунке 1 приведен классический кроссворд.

  
Рисунок 1 – Пример классического кроссворда

* Сканворды. В этих кроссвордах все вопросы внесены прямо в сетку самого кроссворда. При помощи стрелок-указателей определяется расположение разгадываемых слов. В отличие от классических кроссвордов, разгадываемые слова в сканвордах имеют большее количество взаимных пересечений, что в некоторой степени упрощает их разгадывание. В связи с ограниченным пространством для написания вопросов, в сканвордах очень распространены вопросы-задания, построенные на различных ассоциациях, словах-антагонистах, словах-синонимах, определениях пропущенных слов в различных известных словосочетаниях. Пример сканворда представлен на рисунке 2.



Рисунок 2 – Пример сканворда

* Венгерские кроссворды (филворды). Филворд представляет собой поле, заполненное буквами. Из всей этой комбинации букв необходимо отыскать слова, которые указываются в виде отдельного списка или отыскать слова-ответы на ряд вопросов, приложенных к кроссворду. Искомые слова в поле филворда могут располагаться в любом направлении в виде ломаной под прямыми углами линии. Каждая из букв может быть использована только один раз. Пример венгерского кроссворда представлен на рисунке 3.



Рисунок 3 – Пример венгерского кроссворда

* 1. Правила построения кроссвордов

Кроссворд не имеет строгих правил и жестких ограничений, но есть традиции, которых придерживается большинство «кроссвордных» изданий. Обычно, когда упоминаются «правила кроссворда», имеется в виду именно этот негласный стандарт, и уточняются отклонения [7].

Суть кроссворда состоит в разгадывании слов по определениям. К каждому слову дается текстовое определение, в описательной или вопросительной форме указывающее некое слово, являющееся ответом. Ответ вписывается в сетку и, благодаря пересечениям с другими словами, облегчает нахождение ответов на другие определения.

Загаданные слова представляются в виде цепочки ячеек, в каждую из которых по порядку вписываются буквы ответа – по одной в каждую ячейку. Слова пересекаются друг с другом, образуя сетку. Классическая сетка состоит из слов, написанных по вертикали и горизонтали. Любое слово должно быть пересечено как минимум дважды. Сетка не должна иметь изолированных участков.

Слова-ответы должны быть существительными в именительном падеже и единственном числе. Множественное число допускается, когда оно обозначает единственный предмет.

Хорошим тоном считается симметрия сетки кроссворда относительно вертикальной, горизонтальной, диагональных осей. Пример представлен на рисунке 4.



Рисунок 4 – Пример кроссворда с симметричной сеткой

* 1. Описание систем-аналогов

Программа, предназначенная для составления и разгадывания кроссворда, может быть выполнена в виде программы, которую можно установить на компьютер, либо в виде веб-приложения. Также подобный продукт может быть представлен мобильным приложением для смартфона.

Рассмотрим возможности известных программ, предоставляющих инструменты для создания и решения кроссвордов.

* + 1. Приложение «Сrosswordus»

Для создания кроссвордов можно использовать веб-приложение «Crosswordus» [8]. Сервис предоставляет большой выбор классических и фигурных сеток различных размеров, который постоянно пополняется.

Для создания можно воспользоваться генераторами кроссвордов. Генераторы позволяют автоматически создавать сетку с заданными параметрами, располагать введённые с клавиатуры слова на сетке, заполнять сетку словами из словаря.

В системе существует несколько способов создания кроссвордов:

* Создать кроссворд из заданных слов;
* Нарисовать сетку кроссворда и заполнить ее словами из словаря.

Созданным кроссвордом можно поделиться с другими пользователями сервиса или друзьями в социальных сетях, либо распечатать и разгадывать в бумажном варианте.

На рисунке 5 приведена главная экранная форма программы «Crosswordus», на которой можно заметить основные функции для создания кроссворда.

Рисунок 5 – Экранная форма сайта «Crosswordus»

* + 1. Приложение «Puzzlecup»

Puzzlecup – еще один сервис, который помогает без труда составлять кроссворды [8].

На рисунке 6 приведена главная экранная форма программы «Puzzlecup». На форме присутствует рабочая область, в которой пользователь с помощью мышки выделяет место для очередного слова и выбирает автоматически подобранные слова из словаря. Также можно задавать свои слова, заранее придумав определения к ним.

Составленный кроссворд можно сохранить и распечатать. Сохраняется он в куки браузера.

Рисунок 6 – Экранная форма программы «Puzzlecup»

* + 1. Приложение «Hot Potatoes»

«Hot Potatoes» – инструментальная программа-оболочка, предоставляющая преподавателям возможность самостоятельно создавать интерактивные задания, кроссворды и тесты для контроля и самоконтроля учащихся без знания языков программирования и привлечения специалистов в области программирования [9].

С помощью программы можно создать 10 типов упражнений и тестов по различным дисциплинам с использованием текстовой, графической, аудио- и видеоинформации.

Особенностью этой программы является то, что созданные задания сохраняются в стандартном формате веб-страницы: для их использования ученикам необходим только веб-браузер (например, Internet Explorer).

Ученикам не нужна программа «Hot Potatoes», она требуется только преподавателям для создания и редактирования упражнений.

Программа широко используется во всем мире для создания заданий для изучения любых дисциплин.

В состав «Hot Potatoes» входят 5 блоков программ для составления заданий и тестов разных видов. Один из блоков может быть использован как программа для создания кроссвордов. Он называется «JCross» [10].

На рисунке 7 приведена главная экранная форма программы «JCross», в которой существуют два режима:

* ручное размещение слов;
* автоматическое размещение слов.

  
Рисунок 7 – Экранная форма программы «Hot Potatoes: JCross»

При ручном размещении слов положение слов в сетке определяется пользователем: слова буква за буквой вводятся в поле кроссворда. При автоматическом размещении слов все слова вводятся списком в поле ввода слов в открывшемся диалоге, при этом все слова вводятся в отдельные строки, а затем программа подбирает варианты их размещения. В поле Максимальный размер указывается максимальное число ячеек в кроссворде. С помощью кнопок-стрелок в левом нижнем углу экрана можно перемещать созданный кроссворд в рамках поля. Распечатать кроссворд можно в двух вариантах: незаполненный и заполненный.

* 1. Диаграмма объектов предметной области

Объектно-ориентированный анализ и проектирование – технология разработки программных систем, в основу которых положена объектно-ориентированная методология представления предметной области в виде объектов, являющихся экземплярами соответствующих классов [11].

Главными объектами в системе являются кроссворды, которые будет разгадывать пользователь. Кроссворды создаются и редактируются администратором и хранятся в файлах. Кроссворды состоят из слов, которые образуют сетку, и заданий к ним, которые состоят из определений. Слова и определения хранятся в словаре понятий. Пользователь разгадывает кроссворд, после чего может проверить правильность своих ответов. На рисунке 7 приведена диаграмма объектов предметной области.

Рисунок 7 – Диаграмма объектов предметной области

* 1. Постановка задачи

Перед авторами поставлена задача – разработать систему составления и разгадывания классического кроссворда с функциями администратора. Система должна представлять собой настольное приложение. В ней должны быть предусмотрены две роли пользователей: администратор и пользователь.

Для работы в системе, пользователь должен авторизоваться в ней путем ввода логина и пароля. После чего система должна проверить, зарегистрирован ли пользователь и корректен ли введённый пароль. Если пользователь с указанным логином не зарегистрирован, или введен неверный пароль, то система должна выдать соответствующее сообщение. В зависимости от роли пользователю будут доступны различные функции.

*Режим администратора*

Администратору должны быть доступны следующие функции.

1. *Создание кроссворда*. Перед созданием кроссворда администратор должен указать его параметры (высота, ширина и выбрать словарь). Создание нового кроссворда может производиться в одном из двух режимов: автоматическом или ручном.

* *Автоматический режим.*  В этом режиме генерация кроссворда производится по правилу максимального заполнения сетки.
* *Ручной режим.*  В этом режиме администратор заполняет кроссворд с помощью инструментов для создания кроссвордов: добавления, удаления и изменения слова, изменения ориентации слова, а также посредством сортировки словаря. Размещение слов будет возможно либо определением места слова на сетке, либо с использованием технологии «Drag-and-drop».

Для этих режимов необходимо реализовать сохранение кроссворда в файл.

1. *Работа со словарями понятий*. Система должна предоставить администратору возможность создания нового словаря понятий. Вести словарь с широким перечнем возможностей. Например, с добавлением, удалением, изменением понятий.

*Режим пользователя*

Пользователю должна быть представлена следующая функция:

*Разгадывание кроссворда*. Система должна предоставить пользователю возможность открыть из файла кроссворд для решения. Соответствующие функции для этого: выбор слова, выписывание, удаление, изменение буквы, взятие подсказки. Так же для сохранения промежуточного результата решения кроссворда, должна быть возможность сохранения кроссворда в файл.

Таким образом, система должна выполнять следующие функции:

1. *общесистемные функции:*
   * авторизация пользователя в системе (ввод логина/пароля);
   * аутентификация пользователя в системе, настройка интерфейса пользователя на заданную роль;
   * визуализация процессов работы с кроссвордом;
   * автоматическое составление кроссворда по заданным параметрам;
   * контроль количества взятых подсказок;
   * проверка правильности разгадывания кроссворда;
   * выдача справочной информации о системе.
2. *функции администратора:*
   * настройка параметров кроссворда при создании:
3. задание высоты;
4. задание ширины;
5. подключение словаря понятий;
   * составление/редактирование кроссворда:
6. добавление слова;
7. удаление слова;
8. изменение ориентации слова;
9. фильтрация словаря понятий по маске;
10. сортировка словаря
    * сохранение кроссворда в файл заданной структуры;
    * загрузка кроссворда из файла;
    * работа со словарями понятий:
11. добавление понятия;
12. удаление понятия;
13. изменение понятия;
14. проверка дублирования понятий;
15. проверка языка записи понятий;
16. сортировка словаря по выбранному критерию;
17. поиск по заданной маске;
18. загрузка словаря из файла;
19. сохранение словаря из файла;

10) создание нового словаря понятий;

1. *функции пользователя:*
   * загрузка кроссворда из файла;
   * разгадывание кроссворда с организацией системы подсказок:
2. выбор слова;
3. выписывание/удаление/изменение буквы;
4. взятие подсказки
   * сохранение кроссворда в файл.
5. Проектирование системы
   1. Структурная схема системы

Под системой понимается единство целей, ресурсов и строения (структуры), свойства которого определяются отношение входящих в рассматриваемую совокупность объектов (элементов).

*Структурная схема* – это совокупность элементарных звеньев объекта и связей между ними, один из видов графической модели, которая разрабатывается на начальных стадиях проектирования. Она определяет основные функциональные части, назначение и взаимосвязи между ними [12].

При разработке структурной схемы используется методология структурного проектирования, в основе которой лежит алгоритмическая декомпозиция и иерархия вида «часть-целое», учитывающая, что внутренние связи элементов внутри подсистем сильнее, чем связь между подсистемами. Декомпозиция системы может повторяться многократно, вплоть до уровня конкретных процедур, при этом должна быть обеспечена целостность системы, а все составляющие компоненты взаимоувязаны. Для этого используются такие принципы разработки, как «сверху-вниз», «иерархическое упорядочивание» и другие [13].

Структурная схема состоит из одной большой части и изображена на рисунке 8.

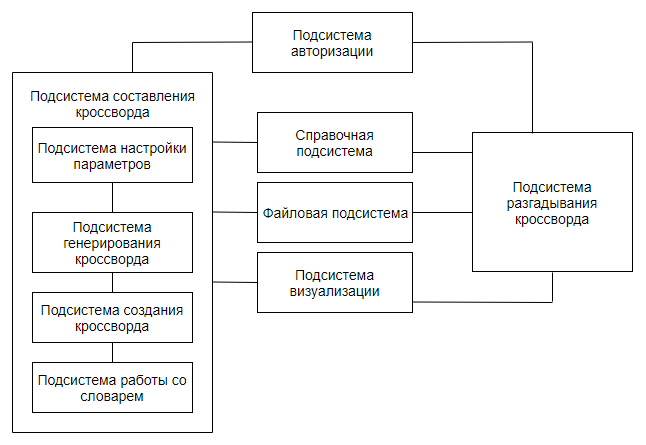


Рисунок 8 – Структурная схема системы

1. *подсистема авторизации*, которая определяет роль пользователя и настраивает интерфейс;
2. *подсистема «Администратор»*,которая включает в себя:

* *подсистему составления кроссворда*, в состав которой входят *подсистема настройки параметров*, которая отвечает за выбор значений параметров кроссворда и поверку корректности этих значений, *подсистема генерирования кроссворда*, позволяющая сгенерировать кроссворд на основе заданных параметров, *подсистема создания кроссворда*, позволяющая добавлять, удалять слова, изменять их ориентацию, а также *подсистема работы со словарем,* которая позволяет добавлять, удалять, изменять слова из словаря понятий;

1. *подсистема «Пользователь»*,в состав которой входит *подсистема разгадывания кроссворда*, которая позволяет разгадать кроссворд;
2. *файловая подсистема*, позволяющая загружать и сохранять кроссворд в файл;
3. *подсистема визуализации*, которая отображает процесс разгадывания;
4. *справочная подсистема*, которая выдает сведения о системе (руководство пользователя) и об ее разработчиках.
   1. Спецификация системы

*Спецификация* *системы* – это документ, который содержит полное и четкое описание разрабатываемого продукта [14]. В терминологии программной инженерии ее называют «разработкой требований» (спецификацией требований).

*Требования* – это свойства, которыми должно обладать программное обеспечение для адекватного задания необходимых функций, а также условия и ограничения на программное обеспечение, данные, среду выполнения и технику [14]. Требования отражают потребности людей (заказчиков, пользователей, разработчиков), заинтересованных в создании программного обеспечения. Различают требования программные, системные, а также функциональные и нефункциональные требования.

*Программные требования* определяют требования к процессу, операционной системе, режиму выполнения ПО, выбору платформы, а также к ним относятся извлечение (сбор), специфицирование и утверждение требований к разрабатываемой программной системе [14].

*Системные требования* описывают требования к программной системе, состоящей из взаимосвязанных программных и аппаратных подсистем и разных приложений [14].

*Функциональные требования* задают назначения системы, а *нефункциональные* – условия выполнения программного обеспечения (например, скорость отклика при выполнении заданной операции) [14].

К нефункциональным требованиям относятся защита данных, адаптивность, изменчивость, которые в свою очередь описываются атрибутами качества (*спецификация качества*) [14].

* + 1. Функциональная спецификация

Функциональная спецификация – это документ, описывающий требуемые характеристики системы (функциональность) [15]. Документация описывает необходимые для пользователя системы входные и выходные параметры. Спецификации помогают устранить дублирование и несоответствия, позволяют точно оценить необходимые действия и ресурсы, выступают в качестве согласующего и справочного документов о внесённых изменениях, предоставляют документацию с конфигурацией, и даёт возможность взаимодействия лиц, работающих с основными функциями системного проектирования. Они дают точное представление о решении проблемы, повышая эффективность разработки системы и оценивая стоимость альтернативных путей проектирования [15]

Функциональная спецификация не определяет операции, происходящие внутри данной системы и каким образом будет реализована её функция. Вместо этого, она рассматривает взаимодействие с внешними агентами (например, персонал, использующий программное обеспечение; периферийные устройства компьютера или другие компьютеры). Функциональная спецификация должна в полном объёме отображать информационные связи проектируемой системы как с внешним миром, так и между подсистемами.

Функциональная спецификация системы приведена в таблице 2.1.

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 (МЕМО) | Визуальное отображение информации | ‑ |
| 1.2 Выдать сведения о системе | Файл справки | Текстовый (\*.HTML) |
| Код ошибки | целое |
| 2.2 Подключить словарь понятий | Имя файла | Строка, \*.dict | Список понятий и их определений | Динамический массив строк |
| Код ошибки | Целое |
| 2 Настройки параметров | 2.1 Задать количество букв в пересечении | Диапазон количества букв: минимальное максимальное | Целое  1  3 | Текущее значение букв в пересечении |
| Код ошибки | Целое |

Продолжение таблицы 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  | … | … | … | … | … |
| 3 Файловая | 3.1 Загрузить файл с кроссвордом | Имя файла | Строка, \*.kros | Кроссворд | Объект, структура определяется в ходе проектирования |

* + 1. Спецификация качества

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

* + 1. 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.

В таблице 2 приведен перечень исключительных ситуаций для разрабатываемой системы и описаны реакции системы на их возникновение.

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 аналогичного сообщения |
| … | … | … |

* 1. Разработка прототипа интерфейса пользователя системы

Вводные слова Вводные слова Вводные слова Вводные слова Вводные слова Вводные слова Вводные слова Вводные слова Вводные слова. Дать определение интерфейса, отметить основные особенности разработки интерфейса.

Здесь должны быть разработаны прототипы **всех** основных форм приложения с описанием привязанной к ней функциональности, например:

На рисунке 5 приведен прототип экранной формы начальной настройки приложения. Здесь пользователь должен выбрать язык программирования, на котором написан алгоритм, категорию (поиск или сортировка) и нажать кнопку «Далее» для перехода к следующему экрану (форме).

  
Рисунок 5 – Прототип экранной формы начальной настройки приложения

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

На рисунке ??? приведена навигационная модель разрабатываемого приложения.

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

* 1. Разработка структур данных и классов

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

* 1. Логическая модель данных (при необходимости)

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

* 1. Выбор и обоснование алгоритмов обработки данных /Разработка и описание алгоритмов обработки данных

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

* 1. Выбор и обоснование комплекса программных средств

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

****

Рисунок ??? ‒ Навигационная модель приложения

* + 1. Выбор языка программирования и среды разработки

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

* + 1. Выбор операционной системы

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

* + 1. Выбор среды программирования

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

* + 1. Выбор системы управления базами данных (при необходимости)

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

1. Реализация системы
   1. Разработка и описание интерфейса пользователя
      1. Разработка и описание пользовательского меню
      2. Описание тестового примера
   2. Реализация классов и структур данных
   3. Физическая модель данных (при необходимости)
   4. Реализация и описание модулей программы
   5. Выбор и обоснование комплекса технических средств
      1. Расчет объема занимаемой памяти

Расчет объема внешней памяти

Для расчета необходимого объема свободной внешней памяти, необходимой для функционирования системы, воспользуемся следующей формулой:

VЖД = VОС + VПР + [VБД] + [VСПО] + [VФ][[2]](#footnote-2),

где VОС – объем памяти, занимаемый операционной системой (операционная система Windows XP с пакетом обновлений SP3, VОС = 1,5 Гб);

VПР – объем памяти, занимаемый непосредственно файлами приложения (VПР = 80 Мб);

VБД – объем памяти, занимаемый базой данных (всеми таблицами) при ее максимальном заполнении. Пример расчета этой составляющей приведен в таблице 7; исходные данные для расчета взяты из описания таблиц БД.

VСПО – объем памяти, занимаемый всем необходимым сопутствующим программным обеспечением (сюда входят СУБД, фреймворки, MS Office (PowerPoint) и другие средства разработки; дадим оценку сверху VСПО в 2 Гб);

VФ – объем памяти, необходимый для хранения файлов, необходимых для работы программы (дадим ему оценку сверху в 2,5 Мб);

Таблица 7 – Расчет объема внешней памяти, необходимой для хранения БД (фрагмент)

|  |  |  |  |
| --- | --- | --- | --- |
| Таблица | Размер записи (байт) | Максимум записей | Всего (байт) |
| Пользователь | 396 | 50 | 19800 |
| Роль | 56 | 10 | 560 |
| Назначенная роль | 48 | 500 | 24000 |
| … | … | … | .. |
| Итого | | | 4045117680 |

VБД = 4045117680 байт = 3950310 Кб = 3858 Мб = 3,76 Гб.

Таким образом, суммарный объем внешней памяти составит:

VЖД = 1,5 Гб + 80 Мб + 3,76 Гб + 2 Гб + 2,5 Мб ≈ 7,5 Гб.

Расчет объема ОЗУ

Для расчета необходимого объема ОЗУ воспользуемся следующей формулой:

VОЗУ = VОС + VПР + [VСПО] + [VБД],[[3]](#footnote-3)

где VОС – ОЗУ, занимаемое операционной системой (256 Мб);

VПР – ОЗУ, которое займет само приложение (не превысит 8 Мб);

VСПО – ОЗУ, занимаемое СУБД и другим сопутствующим ПО (оценим его сверху значением в 128 Мб);

VБД – объем данных из базы, который может быть одновременно загружен в оперативную память (дадим ему оценку сверху в 10 Мб).

Суммарные объемы ОЗУ составит:

VОЗУ = 256 Мб + 8 Мб + 128 Мб + 20 Мб = 412 Мб.

Таким образом, 512 Мб оперативной памяти можно счесть минимально необходимым для функционирования системы.

* + 1. Минимальные требования, предъявляемые к системе

Для корректного функционирования системы необходимо:

1. тип ЭВМ: x86-64 совместимый;
2. объем ОЗУ – не менее 512 Мб;
3. объем свободного дискового пространства – не менее 10 Гб;
4. клавиатура или иное устройство ввода;
5. мышь или иное манипулирующее устройство.

И другие средства, поддерживающие функционирование системы.

ЗАКЛЮЧЕНИЕ

Во время лабораторного практикума была разработана автоматизированная система …, позволяющая ….

В заключении должны быть отражены основные результаты работы, желательно сделать это с привязкой к разделам отчета, например:

В первом разделе приведены основные понятия предметной области, исследованы характеристики систем-аналогов, на основании этого выполнена объектная декомпозиция, отраженная в диаграмме объектов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

**Книги**

***Целиком***

1. Буч Г., Рамбо Д., Якобсон А. Язык UML. Руководство пользователя. Изд. 2-е. М.: ДМК Пресс, 2006. 546 с.

***Если нужно указать номера конкретных страниц***

1. Буч Г., Рамбо Д., Якобсон А. Язык UML. Руководство пользователя. Изд. 2-е. М.: ДМК Пресс, 2006. С. 21.

***Если повторная ссылка на тот же документ***

1. Буч Г., Рамбо Д., Якобсон А. Язык UML … С. 31.

***Если больше 3 авторов***

1. Нестационарная аэродинамика баллистического полета/ Липницкий Ю.М. и [др.]. М.: Физматлит, 2003. 176 с.

**Журналы**

1. Зеленко Л.С., Шумская Е.А. Комплекс программ для работы с учебным контентом в дистанционных обучающих системах// Известия СНЦ РАН. 2015. №2 (5). Т. 17. С. 992-1003.

**Руководящие материалы и ГОСТы**

1. РД 34.20.571. Методические указания по расчету показателей готовности к работе электростанции и энергосистем. Введ. 1976-10-22. М., 1976. 25 с.
2. ГОСТ Р 7.0.4-2006. Издания. Выходные сведения. Общие требования и правила оформления. М., 2006. II. 43 с. (Система стандартов по информ., библ. и изд. делу).

**Методические указания или учебные пособия**

1. Зеленко Л.С. Методические указания к лабораторному практикуму по дисциплине «Программная инженерия». Самара: СГАУ, 2012. 67 с.

**Электронные ресурсы**

1. Российская гидроэнергетика [Электронный ресурс] // Русгидро: [сайт]. URL: http://www.rushydro.ru/industry/russianhydropower/ (дата обращения: 20.12.2014).
2. Гидроэлектростанция (гидроэлектрическая станция, ГЭС) // Энциклопедический словарь юного техника М.: Издательство «Педагогика», 1987 [Электронный ресурс] // Библиотекарь.Ру: электрон. библ. 2006-2017. URL: http://www.bibliotekar.ru/enc-Tehnika/58.htm (дата обращения: 20.12.2014).
3. Субботин А.С. Основы гидротехники [Электронный ресурс]. URL: http://www.cawater-info.net/bk/dam-safety/files/subbotin.pdf (дата обращения: 03.02.2015).
4. Филиальная структура компании [Электронный ресурс] // Системный оператор Единой энергетической системы: [сайт]. [2009-2017]. URL: http://so-ups.ru/index.php?id=about (дата обращения: 20.12.2014).
5. Автоматизированные системы управления технологическими процессами гидроэлектростанции [Электронный ресурс] // Микроника. Инжиниринговый центр: [сайт]. [1999-2016]. URL: http://mikronika-energo.ru/products/asutp/ges-asu-tp/ (дата обращения: 24.12.2014).
6. Автоматизированная система управления производственными процессами [Электронный ресурс] // MEScontrol: [сайт]. [2003-2017]. URL: http://mescontrol.ru/articles/systems (дата обращения: 02.04.2017).
7. Пушников А.Ю. Введение в системы управления базами данных: учеб. пособие [Электронный ресурс] // CITForum: электрон. библиотека. 1997-2017. URL: https://citforum.ru/database/dblearn/ dblearn06.shtml (дата обращения: 20.12.2017).
8. Пользовательский интерфейс [Электронный ресурс] // Википедия: электрон. энциклопедия. 2001-2017. URL: https://ru.wikipedia.org/ wiki/Пользовательский\_интерфейс (дата обращения: 17.03.2015).

***Если необходимо указать системные требования для доступа к документу (наличие специального ПО), то***

1. Белова С.В. Язык UML. Диаграмма вариантов использования. Систем. требования: PowerPoint. URL: nkse.ru/component/k2/item/  
   download/7\_754f5a247edc6ec6be78218f187338a5.html (дата обращения: 17.03.2017).

**Сборники научных трудов или трудов конференций**

1. Философия культуры и философия науки: проблемы и гипотезы: межвуз. сб. науч. тр./ Саратов. гос. ун-т; [под ред. С.Ф. Мартыновича]. Саратов: изд-во Сарат. ун-та, 1999. 199 с.
2. Акимова А.Е., Трешников А.А., Зеленко Л.С. Информационная среда ГЭС. Подсистема расчета показателей эффективности работы оборудования // Перспективные информационные технологии (ПИТ-2017): сб. науч. тр. межд. научно-техн. конф.; [под ред. С.А. Прохорова]. Самара: Изд-во СНЦ РАН, 2017. С. 41-44.

***Если электронное издание***

1. Акимова А.Е., Трешников А.А., Зеленко Л.С. Подсистема расчета показателей эффективности работы оборудования // Математика. Компьютер. Образование: труды XXIV межд. конф., 23-28 января 2017 г., г. Пущино. URL: http://www.mce.su/rus/presentations/ p283063/ (дата обращения: 02.03.2017).

ПРИЛОЖЕНИЕ А  
Руководство пользователя

А.1 Назначение системы

Приводится краткое описание возможностей системы.

А.2 Условия работы системы

***Пример.***

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

1) Требования к техническому обеспечению:

* ЭВМ типа IBM PC;
* процессор типа x86 или x64 тактовой частоты 1400 МГц и выше;
* …

2) Требования к программному обеспечению:

* операционная система Windows XP SP3 и выше;
* установленная платформа .Net версии 4.0 и выше;
* установленная СУБД ….

А.3 Установка системы

***Пример.***

Система поставляется в виде zip-архива. Данный файл необходимо распаковать в любую директорию на жестком диске. Запускаемым файлом системы является файл ххх.exe.[[4]](#footnote-4)

А.4 Работа с системой

А.4.1 Работа с системой в режиме администратора (если необходимо)

Вход в систему (авторизация)

…

А.4.2 Работа с системой в режиме пользователя

Вход в систему (авторизация)

Вход в систему (регистрация)

Настройка параметров кроссворда

ПРИЛОЖЕНИЕ Б   
Листинг модулей программы

7-10 страниц исходного кода шрифт Times New Roman 10 пт 1 интервал

1. *Количество страниц, рисунков, таблиц указывается с учетом приложений* [↑](#footnote-ref-1)
2. […] – значения, указанные в таких скобках, могут отсутствовать [↑](#footnote-ref-2)
3. То, что выделено в [] является необязательным и, если не используется, при расчетах, должно быть убрано из формулы. [↑](#footnote-ref-3)
4. Если необходимы дополнительные ресурсы для обеспечения работоспособности системы, то все для них также должны быть перечислены условия установки. *Если установка нестандартная, то она должна быть подробно описана (в объеме, достаточном для понимания пользователя).* [↑](#footnote-ref-4)