**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«Рязанский государственный радиотехнический университет»**

**(ФГБОУ ВО «РГРТУ», РГРТУ)**

Факультет ВТ К защите

Кафедра ВПМ зав. кафедрой \_\_\_\_\_\_\_\_ Пылькин А.Н.

Направление 09.03.03 «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2017 г.

**Пояснительная записка**

к выпускной квалификационной работа бакалавра на тему

Разработка информационной системы для представления цепочек преобразования типов формообразования естественного языка с помощью графов

Дипломник: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Яровова О. С./

Руководитель проекта: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Пруцков А. В./

Консультант кафедры: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Пруцков А. В./

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2017 г.

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc484517717)

[1 ПОСТАНОВКА ЗАДАЧИ 7](#_Toc484517718)

[2 ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ 9](#_Toc484517719)

[3 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 12](#_Toc484517720)

[3.1 Анализ предметной области 12](#_Toc484517721)

[3.2 Классификация типов формообразования естественного языка 15](#_Toc484517722)

[3.3 Метод генерации и определения 16](#_Toc484517723)

[4 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ 23](#_Toc484517724)

[5 ПРОГРАММНАЯ ДОКУМЕНТАЦИЯ 24](#_Toc484517725)

[6 ТЕСТИРОВАНИЕ 25](#_Toc484517726)

[6.1 Общие положения 25](#_Toc484517727)

[6.2 План тестирования 26](#_Toc484517728)

[6.3 Тестирование работоспособности программной системы 27](#_Toc484517729)

[6.4 Анализ результатов тестирования 30](#_Toc484517730)

[ЗАКЛЮЧЕНИЕ 31](#_Toc484517731)

[БИБЛИОГРАФИЧЕСКИЙ СПИСОК 32](#_Toc484517732)

[ПРИЛОЖЕНИЕ 34](#_Toc484517733)

# ВВЕДЕНИЕ

Под автоматической обработкой текста принято понимать процесс, направленный на преобразование текста на естественном языке при помощи ЭВМ. Эти преобразования используются для машинного перевода, анализа речи, ее синтеза, проверки правописания текстов, поиска и индексации информации в глобальных сетях и решения множества других задач [1].

Во все времена языковая грамотность является неотъемлемой частью любого образованного человека. Каждый из существующих языков обладает своими уникальными чертами и правилами, которые подчас вызывают сомнения в том, какими грамматическими характеристиками обладает слово, какую форму того или иного слова нужно употребить в определенном контексте и как получить необходимую форму. Для решения описанных задач используются информационные системы, которые содержат необходимую информацию.

Русская грамматика и словообразование являются одними из самых сложных, не зависимо от того является ли этот язык родным для изучающего или нет. Именно поэтому в настоящей работе рассматривается формообразование глаголов русского. Выбор данной части речи обусловлен многообразием ее форм и сложностей, связанных с ними, ведь правила русского формообразования глаголов сложны, разнообразны и богаты на исключения, именно поэтому для более эффективного изучения необходимо представление этапов формообразования необходимо осуществить наиболее наглядно.

Реализация этой задачи возможна при помощи представления преобразований в виде графа преобразований. Для его реализации необходимо осуществить классификацию глаголов по типам формообразования и описать получение цепочек словоформ последовательностью преобразований. После этого необходимо оптимизировать полученные цепочки, путем построения древовидной структуры, что позволит сократить скорость определения грамматического значения и основы. Полученные структуры дадут возможность программного получения из любой введенной формы глагола его грамматических характеристик и инфинитива, от которого образована рассматриваемая форма, также возможны и обратные преобразования - получение формы с необходимыми грамматическими характеристиками по инфинитиву [2].

**Целью выпускной квалификационной работы** является создание информационной системы по морфологии русского языка, позволяющей оптимизировать сложные и громоздкие последовательности преобразований, путем представления их в виде графа, а также позволяющей с помощью полученного графа получить необходимые грамматические характеристики определенной словоформы.

Справочная система по морфологии русского языка, на основе рассмотренного подхода, позволит:

* получить грамматические характеристики введенной глагольной формы;
* получить инфинитив глагола по одной из его форм;
* образовывать форму глагола в нужном виде, времени, лице, числе;
* образовывать особые формы глагола (причастия, деепричастия) в нужном виде, времени, лице, числе, падеже.

**Актуальность** темы выпускной квалификационной работы связана с распространенностью исследуемой проблемы и заключается в необходимости применения современных информационных технологий для оптимизации работы.Так как ручное формообразование является процессом, требующим затраты значительного количества временных ресурсов, а также глубоких познаний в области лингвистики, то создание информационной системы, позволяющей автоматизировать данный процесс, поможет избежать ошибок и потери времени в процессе определения грамматических характеристик рассматриваемой формы, а также образования словоформы с требуемыми грамматическими характеристиками от заданного инфинитива.

**Для достижения поставленных целей** в выпускной квалификационной работе реализованы следующие задачи:

* осуществлена классификация глаголов русского языка по типам формообразования;
* составлены полные парадигмы формообразования для каждого образованного типа;
* оптимизирован поиск необходимой парадигмы путем их объединения в граф преобразований;
* осуществлена разработка ПО для поиска грамматических характеристик по составленному графу.

**Практическая ценность.** Данная система может стать полезной для многих, чья жизнь в той или иной мере связана с русским языком. Школьникам она будет полезна для решения вопросов, связанных с трудностями, возникающими при выполнении домашних заданий: поиске инфинитива, определению грамматических характеристик слова. Иностранные студенты здесь смогут найти необходимую информацию, которая поможет им определить правильность употребления той или иной формы в построенной ими фразе на русском языке. Да и каждый человек сможет просто проверить себя, найдя интересующую его информацию.

Также возможна реализация метода и для других естественных языков. Справочная система может стать частью более крупных информационных ресурсов для автоматизации обучения и проверки знаний.

**Для реализации** выпускной квалификационной работы был выбран объектно-ориентированный язык программирования С#. Разрабатываемая информационная система функционирует под управлением операционной системы Microsoft Windows 7/8/8.1/10.

В данной работе представлено 8 разделов, содержание которых описано ниже.

**В разделе постановка задачи** рассматриваются задачи, которые нужно реализовать в разрабатываемой информационной системе.

**В разделе технико-экономическое обоснование** рассматривается состояние проблемы, обосновывается актуальность темы проекта с точки зрения необходимости совершенствования программного обеспечения, дается оценка новизны темы проекта с указанием аналогичных разработок, раскрывается практическая ценность, обосновывается выбор средств разработки.

**В разделе теоретическая часть** анализируется предметная область разрабатываемой информационной системы – уровни автоматической обработки текста, описывается разработанная классификация глаголов русского языка.

**Раздел программная реализация** посвящен проектированию базы преобразований, разработке и настройке программных продуктов, необходимых для полноценной работы системы.

**В разделе программная документация** представлена программная документация, в которой описывается назначение программы, условия ее применения, приведено руководство программиста, содержащее руководство по настройке системы, руководство пользователя, содержащее информацию о том, как эксплуатировать программу.

**В разделе тестирование программы** содержится описание тестирования полученной в ходе разработки информационной системы.

**В заключении** подведены итоги проведенной работы, отмечены преимущества предлагаемого варианта решения, перспективы дальнейшего исследования в этой области.

**В приложении** приведены наиболее значимые части программы.

## ПОСТАНОВКА ЗАДАЧИ

Целью выпускной квалификационной работы является создание информационной системы по морфологии русского языка, позволяющей оптимизировать сложные и громоздкие последовательности преобразований, путем представления их в виде графа, а также позволяющей с помощью полученного графа получить необходимые грамматические характеристики определенной словоформы.

Основаниями для выбора темы для создания системы послужили проблемы, с которыми сталкивался почти каждый, в тех или иных ситуациях, а именно:

* от какого инфинитива образована та или иная словоформа;
* какими грамматическими характеристиками обладает словоформа;
* существует ли у инфинитива определенная словоформа;
* как будет выглядеть определенная словоформа, образованная от нужного инфинитива;
* как образовать все существующие словоформы инфинитива.

Информационная система по морфологии русского языка должна решать следующие задачи:

* по имеющемуся словарю цепочек преобразований построить граф;
* осуществлять по построенному графу определение грамматического значения словоформы и ее инфинитива;
* по инфинитиву осуществлять синтез необходимой грамматической формы.

Разработанная система должна полностью отвечать тем требованиям, которые поставлены перед ней, иметь возможность модернизации для решения новых возникающих задач.

Пользователями системы будут являться школьники, иностранные студенты, а также люди, которым требуется найти необходимую им информацию о формообразовании глаголов русского языка.

Данная информационная система поможет увеличить скорость поиска необходимой информации в данной сфере, что говорит об актуальности выбранной темы.

## ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ

Основой грамотной письменной и устной речи каждого человека служит употребление верных словоформ в том или ином контексте. Каждый язык обладает своими правилами и особенностями формообразования, их изучение является процессом сложным и требует затраты большого количества времени. С другой стороны, для решения такой задачи можно прибегнуть к помощи словаря, но это также довольно трудоемкий процесс при необходимости поиска большого количества информации, осуществляемом вручную. Поэтому и возникла необходимость создания информационной системы, которая бы позволила получить необходимую словоформу, не прибегая к изучению правил формообразования и использованию бумажных словарей.

Информационная система по морфологии русского языка - это совокупность словаря цепочек преобразований, с относящимися к каждой из них инфинитивами, и программных средств и алгоритмов, позволяющих получить необходимую информацию о словоформе.

В настоящее время в сети Интернет существуют словари по морфологии русского языка, такие как «Морфология.ру» [3], «udarenieru.ru» [4]. Однако они имеют недостатки. Однако, эти словари содержат информацию, разнящуюся с информацией из толково-грамматического словаря [5], например, Интернет словари говорят о том, что такие глаголы как «портить, тратить» имеют формы страдательного причастия настоящего времени, однако, в [5] содержится информация об обратном. Возможно, это связано с тем, что рассматриваемые Интернет словари составлены на основе грамматического словаря [6], который является изданным в 1977 г., тогда как [5], по информации из которого составлена разрабатываемая система, является более современным и издан в 1989 г. Так же Интернет словарями не представляется возможности воспользоваться без наличия сети Интернет, однако это вряд ли можно назвать недостатком, так как в наше время он окружает нас по всюду, и возможность им воспользоваться есть практически у каждого.

Для разработки информационной системы была выбрана среда разработки Microsoft Visual Studio 2015.

Среда разработки Microsoft Visual Studio 2015 является популярной интегрированной средой разработки (integrated development environment — IDE), которая доступна в настоящее время программистам. Эта среда является результатом долгой истории развития языков программирования и интерфейсов и вобрала в себя достижения многих сред разработки программного обеспечения [7].

Microsoft Visual Studio 2015 включает в себя редактор исходного кода с возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных. Visual Studio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода, добавление новых наборов инструментов (например, для редактирования и визуального проектирования кода на предметно-ориентированных языках программирования) или инструментов для прочих аспектов процесса разработки программного обеспечения [8].

Microsoft Visual Studio 2015 представляет собой высокоэффективную среду разработки, которая позволяет решать различные прикладные задачи, соединять несколько технологий, дающих разработчикам возможность создания таких приложений как Windows, Web-служб, Web-приложений, сайтов и т.д. Microsoft Visual Studio 2015 позволяет обеспечить высокое качество кода на протяжении всего цикла разработки.

В качестве языка программирования был выбран язык С#, который является одним из самых современных языков программирования и встроен в среде Microsoft Visual Studio 2015. С# наследует большинство языковых конструкций языка С++, однако имеет и языковые конструкции, предназначенные для создания прикладного программного обеспечения, которые не имеют прямых аналогов в других языках, например, деревья выражений, позволяющие представлять код в виде древовидной структуры данных, LINQ- (Language-Integrated Query)- язык запросов к источнику данных, в качестве которого может выступать объект, реализующий интерфейс IEnumerable (например, стандартные коллекции, массивы), набор данных DataSet, документ XML. Но вне зависимости от типа источника LINQ позволяет применить ко всем один и тот же подход для выборки данных.

Так же для реализации поставленной задачи с помощью Microsoft Visual Studio 2015 не требует материальных затрат, так как в рамках поставленной задачи достаточно бесплатной версии Community.

Таким образом, можно сделать вывод, что использование среды Microsoft Visual Studio 2015 и языка С#, является оптимальным решением для выполнения поставленных задач.

## ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

### Анализ предметной области

Автоматическая обработка текста представляет собой преобразование текстов на естественном языке при помощи ЭВМ. Данная технология используется при анализе и синтезе речи, машинном переводе, проверке правописания текстов, а также для решения множества других задач [9].

Выделяют три уровня обработки текста:

* морфологический;
* синтаксический;
* семантический.

Морфологический анализ включает в себя выделение грамматической основы слова, определение принадлежности слова к той или иной части речи, приведение слова к словарной форме (инфинитиву), определение принадлежности словоформы слову путем анализа структуры словоформы.

Синтаксический анализ направлен на выявление связей между словами в предложении, построением синтаксической структуры предложения.

Семантический анализ характеризуется выявлением смысловых связей между словами и синтаксическими группами.

Каждый из уровней анализа является самостоятельной задачей, которая не имеет собственного практического применения, но является активно используемой для решения более общих задач. Многие исследовательские системы предназначены для решения именно таких задач. Эти системы применяются для апробации методов и проведения вычислительных экспериментов, либо в качестве составных частей (или библиотек) для систем, решающих ту или иную прикладную задачу [10]. Можно сказать, что уровни автоматической обработки текста являются подсистемами общей системы естественного языка, они взаимосвязаны, но в тоже время достаточно автономные.

К первому этапу обработки текста относится морфологический анализ слов. Входным параметром является текстовое представление исходного слова. Целью и основным результатом морфологического анализа являются установление связи анализируемой словоформы с ее представлением в словаре, а также определение всех морфологических характеристик слова.

После того как проведен анализ каждого слова (морфологический анализ), начинается анализ предложений (семантический анализ), который позволяет определить взаимосвязь между отдельными словами и частями предложений. Результатом такого анализа является граф, узлы которого представляют собой слова предложения, при этом, если два слова связаны каким-либо образом, то соответствующие им вершины графа связаны дугой с определенной окраской. Возможные окраски дуг зависят от языка, на котором написано предложение, от выбранного способа представления синтаксической структуры предложения. Например, при синтаксическом анализе предложений русского языка в качестве окрасок дуг можно использовать вопросы, задаваемые от одного слова к другому. В вершинах графа слова пишутся не в том виде, в котором они встречаются в предложении, а в своей основной словоформе. Некоторым словам (например, предлогам) вообще не соответствует ни одна из вершин графа, но эти слова влияют на вопросы, задаваемые от одного слова к другому [11].

Семантический анализ в свою очередь основывается на результатах синтаксического анализа, в результате которого получается структура, отражающая синтаксическую структуру каждого предложения, а не набор слов, разделенных на предложения. Решение задач семантического анализа основываются на результатах анализа отдельных слов, и вместо синтаксической структуры предложения анализу подвергаются наборы стоящих рядом слов. Так как методы семантического анализа работают со смыслом слов, то необходим универсальный набор семантических единиц при помощи которого можно было бы выразить смысл любого высказывания. На данном этапе основой является словарь синонимов, антонимов, омонимов, паронимов и т.д. [12].

Все уровни обработки являются взаимосвязанными, и чтобы преобразовать текстовой запрос пользователя в его смысловое представление и, обратно, преобразовать смысл ответа в его текстовое представление необходимо выполнить автоматическую обработку текста на морфологическом, синтаксическом и семантическом уровнях [13].

В основе проектируемой информационной системы лежит морфологический уровень автоматической обработки текста. На этом уровне осуществляется морфологический анализ и синтез.

При морфологическом анализе для каждого слова осуществляется задача определения - нахождения по данной словоформе ее нормальной формы (инфинитива) и грамматического значения.

Морфологический синтез заключается в решении для каждого слова задачи генерации. Генерация формы слова - это процесс получения необходимой словоформы по известному инфинитиву и грамматическому значению.

Для морфологической обработки текста используются словари, в которых содержатся классифицированные слова. Чтобы классифицировать слова, необходимо построить полную парадигму, то есть совокупность всех словоформ слова.

В проводимой классификации рассматривается получение словоформ слова с помощью двух основных методов преобразования:

* добавление подстроки к строке слева или справа;
* замена первой слева подстроки на другую подстроку.

Каждое основное преобразование имеет обратное к нему преобразование. Все остальные преобразования возможно представить в виде конечной последовательности перечисленных основных и/или обратных типов преобразований. Данные понятия используются в универсальной модели формообразования [14].

### Классификация типов формообразования естественного языка

В качестве естественного языка в рассматриваемой системы был выбран русский язык. Данный выбор обусловлен тем, что этот язык является богатым на различные словоформы, например, в отличие от многих других в русском языке возможно изменение слов по падежам.

Для более детального рассмотрения и классификации был выбран глагол- часть речи русского языка, обозначающая действия или состояния разного характера.

Глагол обладает различными грамматическими категориями и соответствующими словоформами. Как и другие части речи, он имеет категории числа, рода, лица, однако есть и уникальные, присущие только глаголу- категории вида, времени, залога, наклонения. Таким образом, глагол является самой сложной, с точки зрения, формообразования частью речи русского языка, так как он обладает уникальными признаками, присущими лишь глаголу, поэтому была проведена классификация именно этой части речи.

Все глагольные формы делятся на спрягаемые и неспрягаемые.

К спрягаемым относятся глаголы настоящего, будущего и прошедшего времени изъявительного наклонения, в первом, втором и третьем лице, единственном и множественном числе. А также глаголы повелительного, в единственном и множественном числе, и условного наклонений, в первом, втором и третьем лице, единственном и множественном числе.

К неспрягаемым формам относятся отглагольные формы глагола, деепричастия совершенного и несовершенного вида, а также причастия настоящего и прошедшего времени, действительного и страдательного залога, женского, мужского и среднего родов, а также именительного, родительного, винительного, дательного, творительного и предложного падежей.

Для проведения классификации был выбран толково-грамматический словарь глаголов русского языка [5].

Процесс классификации включает в себя следующие этапы:

* составить полную парадигму словообразования слова;
* сравнить составленную цепочку с уже существующими;
* при полном совпадении цепочек преобразований отнести рассматриваемое слово к уже имеющемуся типу;
* при отсутствии полного совпадения, образовать новый тип;
* типом называется группа, к которой относятся два и более слова, в противном случае, член типов относят к группе исключений - слов, которые изменяются индивидуальным образом.

В результате классификации было рассмотрено 2500 инфинитивов, которые разделились на 124 типа, к глаголам, изменяемым индивидуальным образом отнесено 134.

Такое многообразие типов связано с тем, что русский является богатым на особенности орфографии, а также с тем, что многие глаголы, имеющие одинаковые парадигмы в глагольных формах, отличаются наличием различных отглагольных форм – причастий, например, глагол может иметь только действительные причастия настоящего и прошедшего времени, или действительные причастия настоящего и прошедшего времени, и страдательные причастия настоящего времени, или действительные причастия настоящего и прошедшего времени, и страдательные причастия прошедшего времени, или действительные и страдательные причастия настоящего и прошедшего времен.

После проведения классификации была составлена подробная таблица преобразования форм для полученных типов, содержащая цепочки преобразования глаголов русского языка.

### Метод генерации и определения

Для решения поставленных задач используются алгоритмы генерации и определения.

Алгоритм определения заключается в определении грамматического значения и основы (инфинитива) рассматриваемой словоформы. Суть алгоритма заключается в последовательном переборе цепочек преобразований типов. Если последовательность преобразований применима к рассматриваемой словоформе и тип цепочки совпадает с типом полученной основы в ходе совершенных преобразований, то такие цепочки называются целевыми, в противном случае – ложными. Так же существуют тупиковые цепочки – цепочки, выполнение преобразований которых невозможно к рассматриваемой словоформе, что приводит к ее прерыванию. При нахождении первой целевой цепочки процесс поиска не заканчивается, это связано с существованием омонимии в естественных языках, таким образом не одна, а несколько цепочек могут быть применимы к рассматриваемой словоформе. Так, например, к словоформе золотящий может быть применена последовательность преобразований действительного причастия настоящего времени единственного числа мужского рода как в именительном падеже, так и в винительном падеже для неодушевленной формы, обе эти цепочки будут являться целевыми.

В свою очередь алгоритм генерации заключается в поиске необходимой последовательности преобразований по имеющейся основе (инфинитиву) и грамматическому значению, и применении ее для получения необходимой словоформы.

Цепочки преобразований можно представить в виде графа. Вершинами графа являются словоформы, корневая вершина – основа, промежуточные – результаты преобразований, конечные вершины – словоформы. Дуги графа – необходимые преобразования, которые необходимо осуществить, чтобы получить из одной формы другую. Путь в графе представляет собой цепочку преобразований, при этом для задачи генерации используется путь от корневой вершины к конечной – прямая цепочка преобразований, а для задачи определения – обратная, заключающаяся в движении от конечной вершины к корневой, то есть нахождения такой конечной вершины, которая приедет к корневой.

Процесс движения по графу в задаче определения состоит из следующих действий:

* применить преобразование к словоформе;
* проверить применимость преобразования к словоформе: если преобразование применимо – перейти в следующее состояние, иначе считать путь тупиковым;
* по достижении конечной вершины проверить совпадение типов цепочки и полученной основы, считать путь целевым, иначе – ложным.

Процесс движения по графу в задаче генерации заключается в:

* определении типа основы;
* определении цепочки преобразований, соответствующей заданному грамматическому значению и типу основы;
* применении найденной цепочки к основе.

Предлагаемый метод представления цепочек преобразования в виде графа является более компактным и быстрее обрабатываемым нежели метод полного перебора. Это связано с тем, что для нахождения необходимой основы (инфинитива) и грамматического значения словоформы, в методе полного перебора, необходимо осуществить применение всех имеющихся в словаре цепочек преобразования, в результате чего будут получены необходимые значения. Этот путь является более долгим и неоптимальным, так как в разных типах и формах возможны повторения цепочек преобразований, и повторный их перебор не имеет смысла в большинстве случаев. В свою очередь метод представления цепочек формообразования в виде графа позволяет объединить одинаковые цепочки в одну, имеющую при себе пары всех грамматических значений и соответствующих им типов, что сокращает количество перебираемых цепочек. Так же при обходе полученного графа в случае неприменимости какого-то преобразования из рассматриваемой цепочки все последующие цепочки, содержащие на этом уровне данное преобразование, также не рассматриваются.

В своей работе [15] Розанов А.К. предлагает представление и осуществление преобразований с помощью графов, однако, представленный в его работе граф, может содержать только два вида преобразований:

* добавление подстроки в конец строки;
* удаление подстроки с конца строки.

Однако предложенный им метод не является универсальным, так как подходит только для суффиксального формообразования. Так, например, в формообразовании русского языка на ряду с суффиксальным используется и приставочный способ формообразования.

Для построения графа преобразования для решения задачи определения необходимо проанализировать цепочки преобразования, осуществив следующие операции:

* сформировать список, содержащий все возможные варианты преобразований первого уровня;
* поставить в соответствии каждому элементу первого уровня принадлежащие ему элементы второго уровня;
* совокупности элементов первого и второго уровня, сопоставить элементы следующего уровня;
* продолжать сопоставление до тех, пор пока не закончатся все уровни преобразования;
* при окончании цепочки, сопоставить ей соответствующие пары значений: тип – грамматическое значение – (у цепочки может быть несколько).

Рассмотрим наглядное применение предложенного алгоритма.

Пусть имеются цепочки преобразований:

Таблица 3.1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ТИП | ФОРМА | ПР1 | ПР2 | ПР3 | ПР4 |
| Тип1 | Форма1 | a1 | b1 |  |  |
| Тип1 | Форма2 | a1 | b2 | c1 |  |
| Тип1 | Форма3 | a2 | b3 | c2 | d1 |
| Тип1 | Форма4 | a3 | b4 | c3 |  |
| Тип2 | Форма1 | a1 | b2 | c3 |  |
| Тип2 | Форма2 | a3 | b2 |  |  |
| Тип2 | Форма3 | a2 | b2 | c2 | d2 |
| Тип2 | Форма4 | a1 | b2 | c1 |  |

На первом этапе, рассмотрев первые преобразования, получим (Рис.3.1)

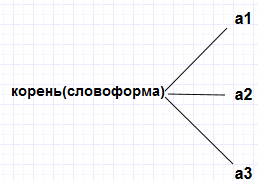


Рис. 3.1 Результата первого этапа при построении графа

На втором этапе, поставив в соответствии к выделенным на первом этапе преобразованиям преобразования второго уровня получим (Рис.3.2).

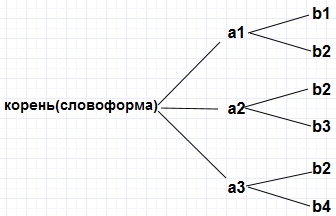


Рис. 3.2 Результат второго этапа при построении графа

На третьем этапе, установив необходимые соответствия, получим (Рис 3.3).

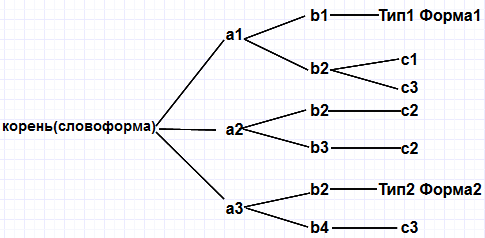


Рис. 3.3 Результат третьего этапа при построении графа

Рассмотрев все уровни преобразования, получим граф следующего вида (Рис. 3.4)

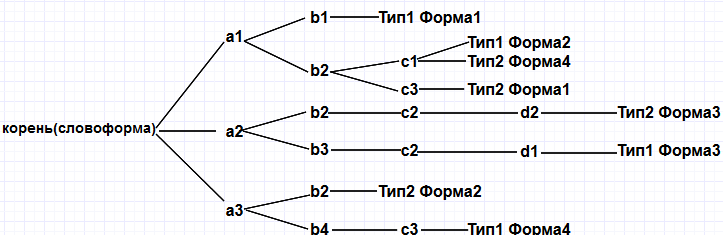


Рис. 3.4 Полученный граф преобразований

Так же следует отметить, что построение графа преобразований имеет смысл только для задачи определения, для задачи генерации построение графа не требуется. Это связано с тем, что для задачи генерации построение графа не целесообразно, так как первым уровнем графа будут являться пары тип – грамматическое значение, а они являются уникальными, значит никакого сокращения количества цепочек не произойдет. Цепочки преобразования, принадлежащие парам тип – грамматическое значения являются уникальными, а значит однозначно идентифицируемыми. В задаче генерации однозначно определяющим параметром для идентификации цепочки является принадлежность инфинитива определенному типу, а также словоформа, заданная пользователем.

## ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

### Структура информационной системы

Для реализации проектируемой информационной системы предлагается следующее схематичное представление состава и взаимодействия ее составных частей (Рис. 4.1).

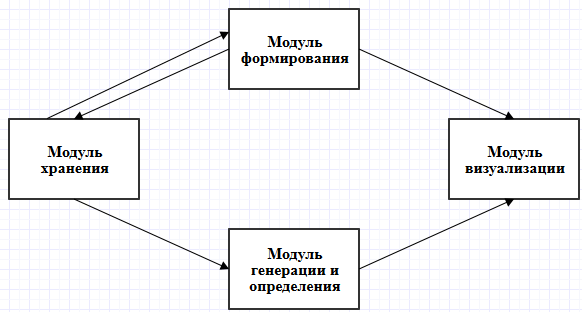


Рис. 4.1 Схема представления состава и взаимодействия частей системы

Рассмотрим подробно каждую из предлагаемых частей системы.

Модуль хранения представляет собой набор нескольких типов файлов:

* файл, содержащий полные цепочки преобразования, с указанием грамматического значения и типа, для задачи определения;
* файл, содержащий полные цепочки преобразования, с указанием грамматического значения и типа, для задачи генерации;
* файлы, соответствующие каждому типу формообразования, в каждом из которых содержатся основы (инфинитивы), принадлежащие определенному типу;
* файл, содержащий сформированный программно, в модуле формирования, граф преобразования для задачи определения.

Модуль формирования представляет собой алгоритм, который осуществляет оптимизацию процесса полного перебора, путем составления графа преобразования, полученный граф хранится в виде файла в модуле хранения для дальнейшего его использования для решения задачи определения.

Модуль генерации и определения представляет собой алгоритмы решения задач генерации и определения на основе данных файлов, представленных в модуле хранения, а также исходных данных, определенных пользователем в модуле визуализации.

Модуль визуализации представляет собой интерфейс пользователя, посредством которого осуществляется взаимодействие системы и пользователя, а именно ввод исходных данных пользователем, и отображение результата работы системы.

### Структура входных и выходных данных

Для корректной работы системы наборы данных должны быть представлены в определенных, понятных системе форматах.

Рассмотрим структуры файлов, представленных в модуле хранения.

Файл, содержащий в себе цепочки преобразования для задачи определения, и файл, содержащий цепочки преобразования для задачи генерации, представляют собой XML файлы имеющих определенную иерархическую структуру (Рис.4.2).

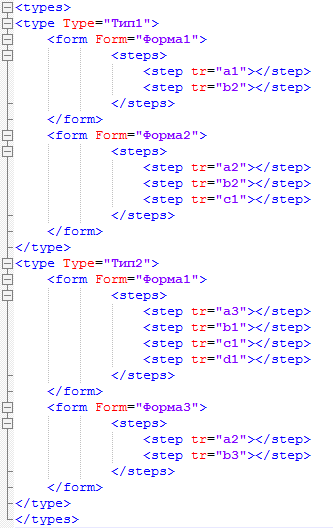


Рис. 4.2 Структура XML файла для представления цепочек формообразования

Рассмотрим структуру XML файла подробно.

Тег *<types>* является контейнером, содержащим все возможные типы формообразования. Этот тег содержит множество тегов *<type>,* каждый из которых имеетатрибут *Type*, значение которого однозначно идентифицирует этот конкретный тип среди множества других.

Каждый тег *<type>* содержит информацию о всех словоформах, образование которых возможно в данном типе. Каждая форма идентифицируется тегом *<form>* c атрибутом *Form*, значение которого содержит имя формы – ее грамматическое значение.

Каждый тег *<form>* содержит цепочку преобразований, соответствующую рассматриваемой форме. Цепочка преобразований заключается в тег *<steps>.*

Так как цепочка преобразований — это последовательность нескольких преобразований осуществляющихся поочередно, необходимо представление каждого этапа преобразования отдельно. Данная задача решается при помощи представления в теге *<steps>,* списка преобразований, каждое из которых само по себе заключено в тег *<step>.* Атрибутом тега *<step>* является *tr*, который содержит в себе само преобразование, которое необходимо осуществить.

В разрабатываемой системе возможны четыре типа преобразований:

* добавление подстроки в конец строки;
* добавление подстроки в начало строки;
* удаление подстроки с конца строки;
* удаление подстроки из начала подстроки.

Так как каждое из этих преобразований должно быть однозначно идентифицируемо в системе, принято соглашение об необходимой форме записи каждого вида преобразований.

Так для обозначения преобразования добавления подстроки в конец строки необходимо в теге *<steps>* написать:

<steps>

<step tr=”+[преобразование]”></step>

</steps>

Для обозначения преобразования добавления подстроки в начало строки необходимо в теге *<steps>* написать:

<steps>

<step tr=”\*[преобразование]”></step>

</steps>

Для обозначения преобразования удаления подстроки с конца строки необходимо в теге *<steps>* написать:

<steps>

<step tr=”-[преобразование]”></step>

</steps>

Для обозначения преобразования удаления подстроки из начала строки необходимо в теге *<steps>* написать:

<steps>

<step tr=”/[преобразование]”></step>

</steps>

Количество тегов *<step>* внутри тега *<steps>* не ограничено и зависит от длины цепочки преобразования.

Отличие файла преобразований для задачи генерации от файла преобразований для задачи определения состоит в обратных знаках преобразования, и обратном порядке осуществления преобразований.

Файлы, содержащие основы (инфинитивы) для каждого типа, представляют собой текстовые файлы в которых находится список слов (Рис.4.3).

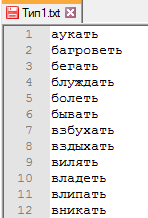


Рис. 4.3 Файл списка основ (инфинитивов) для определенного типа

Главным условием составления каждого файла являются полное совпадение имени файла и значения атрибута *Form* тега *<form>,* что позволит однозначно идентифицировать основы, принадлежащие данному типу*.*

Так же для правильного проведения процессов анализа и синтеза необходимо верное распределение основ по файлам типам в соответствии с классификацией.

Файл, содержащий сформированный программно граф преобразования для задачи определения представляет собой …????

### Структура классов системы

### Разработка интерфейса пользователя

Понятие интерфейс представляет собой комплекс средств и методов, который позволяет обеспечить взаимодействие пользователя с различными устройствами и программами. Графический интерфейс пользователя позволяет осуществлять это взаимодействие посредством визуальных компонентов (окон, кнопок, полей и т.д.).

Интерфейс, разрабатываемой информационной системы, представляет собой многостраничное окно, каждая страница которого отвечает за определенную программную функцию.

Первая страница – «Главная» предоставляет пользователю справочную информацию о системе.

Вторая страница – «Морфологический анализ». На данной странице пользователь имеет возможность получить по введенному значению словоформы ее грамматическое значение и основу (инфинитив), от которой словоформа образована.

Третья страница – «Морфологический синтез». На данной странице пользователь имеет возможность получить по введенной основе (инфинитиву) и грамматическому значению соответствующую словоформу.

Четвертая страница – «Граф преобразований». На данной странице пользователь может просмотреть граф преобразований, по которому осуществляется процесс решения задач морфологического анализа.

При запуске программы на экран выводится форма для работы с информационной системой, которая представляет собой многостраничное окно (Рис.4.4).

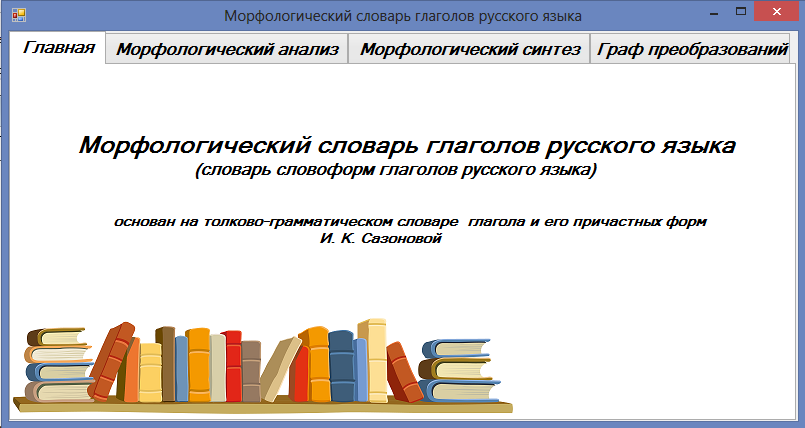


Рис. 4.4 Главная страница информационной системы

Первая страница – «Главная» содержит компоненты, представленные в таблице 4.1.

Таблица 4.1

|  |  |
| --- | --- |
| Компонент | Описание |
| label1 | Надпись «Морфологический словарь глаголов русского языка» |
| label2 | Надпись «(словарь словоформ глаголов русского языка)» |
| label3 | Надпись «основан на толково-грамматическом словаре глагола и его причастных форм» |
| label4 | Надпись И.К. Сазоновой |
| pictureBox1 | Изображение с книгами |

Вторая страница – «Морфологический анализ», представлена на (Рис. 4.5).

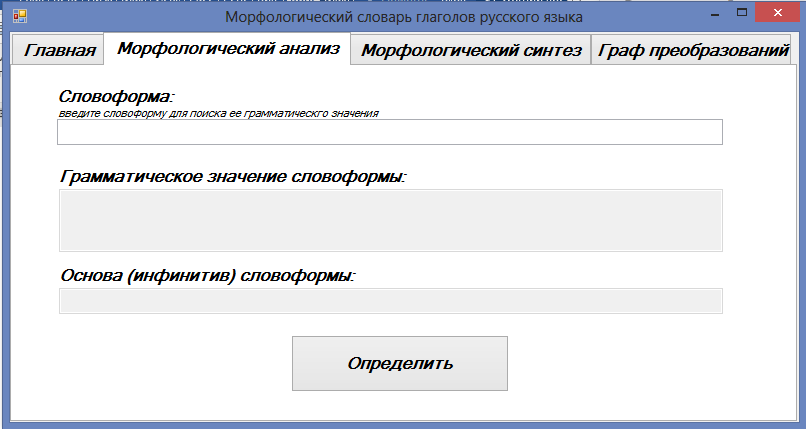


Рис. 4.5 Страница осуществления морфологического анализа

Страница содержит следующие компоненты, представленные в таблице 4.2.

Таблица 4.2

|  |  |
| --- | --- |
| Компонент | Описание |
| labelWordform | Надпись «Словоформа:» |
| labelComments | Надпись «введите словоформу для поиска ее грамматического значения» |
| textBoxWordform | поле ввода словоформы |
| labelGrammarValue | Надпись «Грамматическое значение словоформы» |
| textBoxGrammarForm | Поле для вывода грамматического значения |
| labelInfinitive | Надпись «Основа (инфинитив) словоформы» |
| textBoxResult | Поле для вывода инфинитива |
| buttonIdentify | Кнопка осуществления процесса морфологического анализа |

Третья страница – «Морфологический синтез», представлена на (Рис. 4.6).

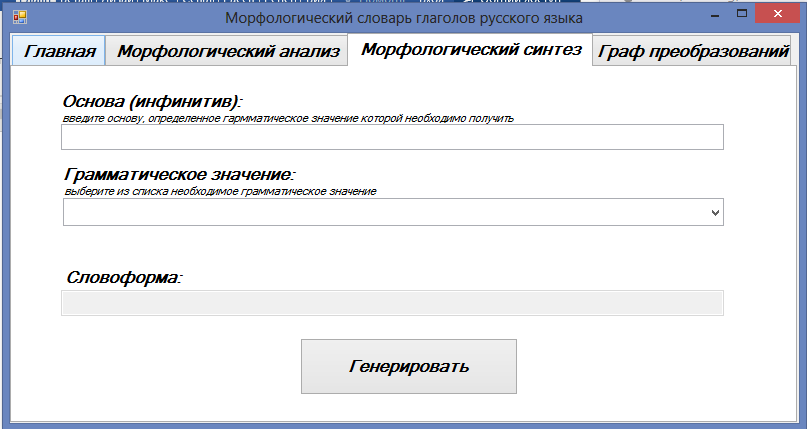


Рис. 4.6 Страница осуществления морфологического синтеза

Страница содержит следующие компоненты, представленные в таблице 4.3.

Таблица 4.3

|  |  |
| --- | --- |
| Компонент | Описание |
| labelInfinitive2 | Надпись «Основа (инфинитив):» |
| labelComments1 | Надпись «введите основу, определенное грамматическое значение которой необходимо получить» |
| textBoxInfinitive | Поле ввода основы (инфинитива) |
| labelGrammarForm2 | Надпись «Грамматическое значение:» |
| labelComments2 | Надпись «выберите из списка необходимое грамматическое значение» |
| comboBox1 | Выпадающий список выбора необходимого грамматического значения формы |
| labelWordform2 | Надпись «Словоформа:» |
| textBoxResult1 | Поле вывода результата морфологического синтеза |

Четвертая страница – «Граф преобразований», представлена на (Рис. 4.7).

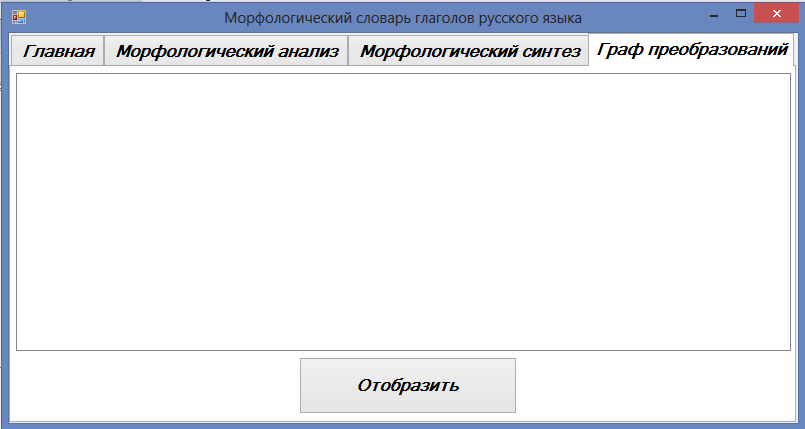


Рис. 4.7 Страница просмотра графа преобразований

Страница содержит следующие компоненты, представленные в таблице 4.4.

Таблица 4.4

|  |  |
| --- | --- |
| Компонент | Описание |
| treeView1 | Компонент для отображения древовидной структуры |
| buttonShow | Кнопка для отображения графа преобразования |

## ПРОГРАММНАЯ ДОКУМЕНТАЦИЯ

### Назначение программы

Разрабатываемая информационная система создается для хранения и обработки информации о правилах морфологии естественного языка, а также для автоматизации процесса морфологического анализа и синтеза глаголов русского языка. Данная система позволит осуществлять задачи определения грамматического значения и основы (инфинитива) введенной словоформы, задачи генерации словоформы основы (инфинитива) по грамматическому значению, а также просмотр полного графа преобразований для задачи определения.

Разрабатываемая система содержит словари цепочек преобразований типов и соответствующих им основ (инфинитивов), на основе которых осуществляется поиск и обработка информации, поступившей от пользователя.

### Условия для работы программы

Для корректной работы разрабатываемой информационной системы необходимо убедиться в наличие соответствия следующим системным требованиям:

* процессор с частотой 1,6 ГГц или выше;
* 256 Mb RAM: 256 Мб оперативной памяти;
* 200 Mb HDR: 200 Мб свободного места на жёстком диске;
* не требуется звукового или видео оборудования;
* монитор, клавиатура, мышь.

Программная конфигурация компьютера необходимая для работы –операционная система MS Windows 7/8/8.1/10.

### Общие сведения о главном окне информационной системы

Главное окно приложения представляет собой форму, содержащую 4 вкладки:

* Главная;
* Морфологический анализ;
* Морфологический синтез;
* Граф преобразований.

Вкладка «Главная» предоставляет краткую информацию о назначении системы.

Вкладка «Морфологический анализ» позволяет осуществлять задачу определения грамматического значения и основы (инфинитива) по словоформе.

Вкладка «Морфологический синтез» позволяет осуществить задачку генерации словоформы по основе (инфинитиву) и грамматическому значению.

Вкладка «Граф преобразований» позволяет посмотреть структуру графа преобразования типов для задачи определения.

### Руководство системного администратора

Перед установкой программного обеспечения системный администратор должен проверить соответствие персонального компьютера требованиям, предъявляемым к техническим и программным средствам, представленным выше. При несоблюдении требований при функционировании системы могут возникнуть проблемы.

Если требования соблюдены, можно приступить к установке программы.

Для установки разрабатываемой системы достаточно скопировать ее на жесткий диск компьютера и проверить, чтобы файлы словаря, находились в этом же каталоге. После чего можно приступать к эксплуатации системы.

### Руководство пользователя

При запуске программы пользователю предоставляется главное окно информационной системы (Рис. 5.1).

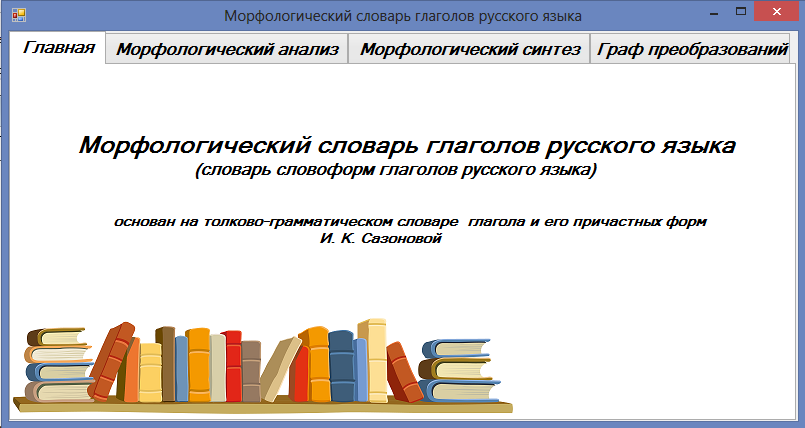


Рис. 5.1 Главное окно информационной системы

Для осуществления задачи определения необходимо выбрать вкладку «Морфологический анализ», после чего форма примет необходимы й вид, для осуществления выбранное операции (Рис.5.2).

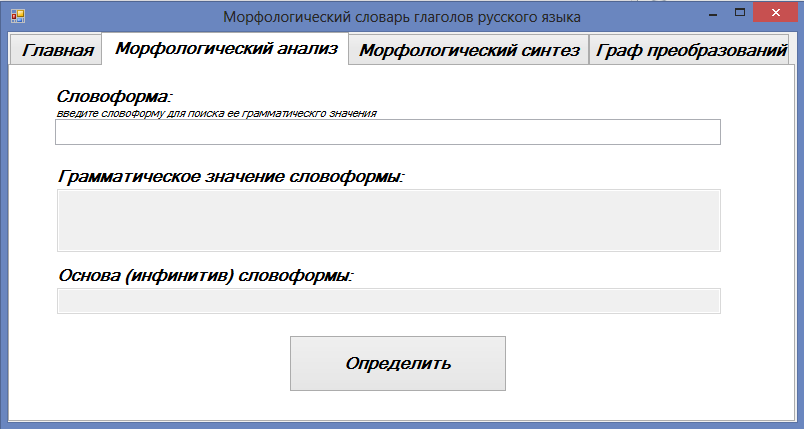


Рис. 5.2 Вкладка «Морфологический синтез»

Далее необходимо в поле «Словоформа» ввести словоформу, информацию о характеристиках которой необходимо получить. После чего нажать кнопку «Определить». После вышеописанных действий, система отобразит необходимую информацию о характеристике словоформы (Рис. 5.3).

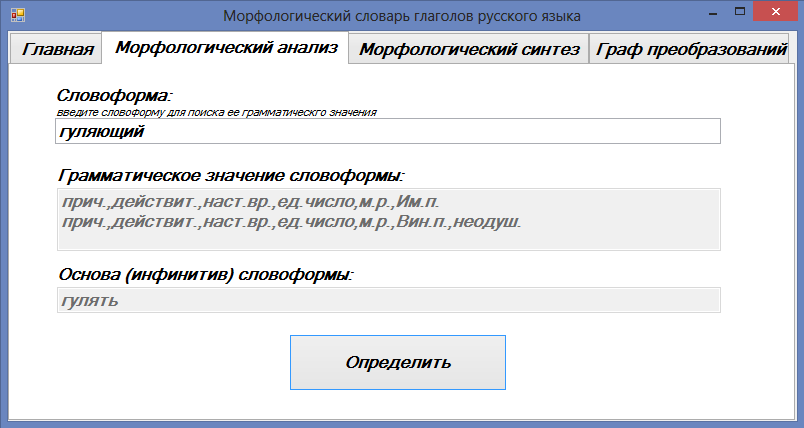


Рис. 5.3 Результат морфологического анализа

Для осуществления задачи генерации необходимо выбрать вкладку «Морфологический синтез», после чего форма примет необходимы й вид, для осуществления выбранное операции (Рис.5.4).

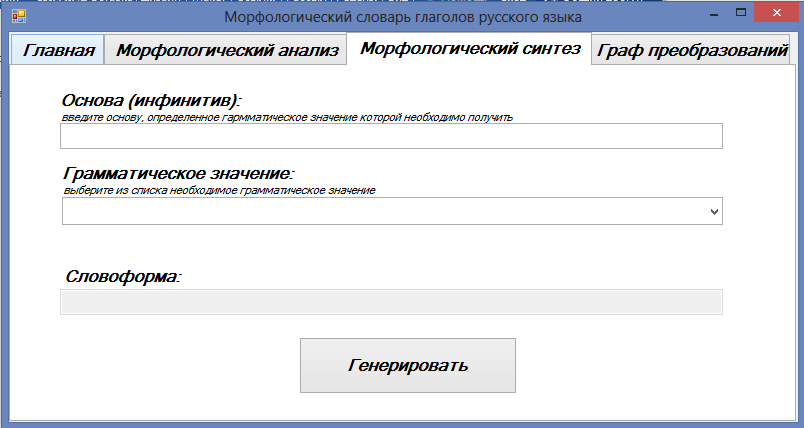


Рис. 5.4 Вкладка «Морфологический синтез»

Далее необходимо в поле «Основа (инфинитив)» ввести основу, от которой необходимо образовать определенную словоформу. Грамматическую характеристику для образования словоформы необходимо выбрать из выпадающего списка поля «Грамматическое значение». После чего нажать кнопку «Генерировать». После вышеописанных действий, система отобразит образованную словоформу с заданными характеристиками (Рис. 5.5).

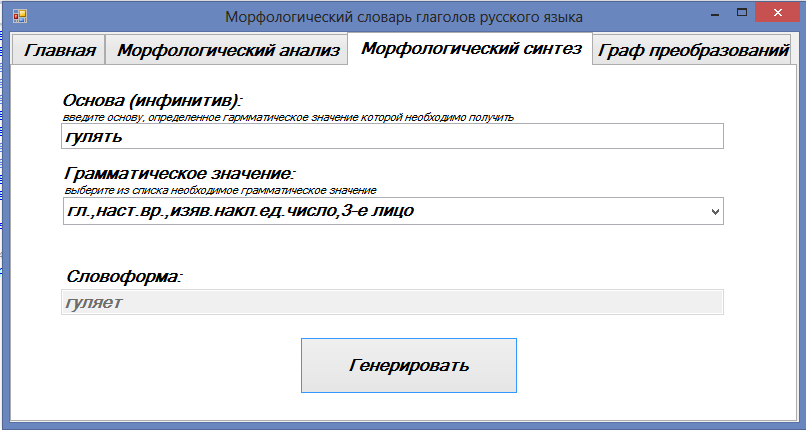


Рис. 5.5 Результат морфологического синтеза

Для просмотра графа преобразований для морфологического анализа необходимо выбрать вкладку «Граф преобразований», после чего форма примет необходимы й вид, для осуществления выбранное операции (Рис.5.6).

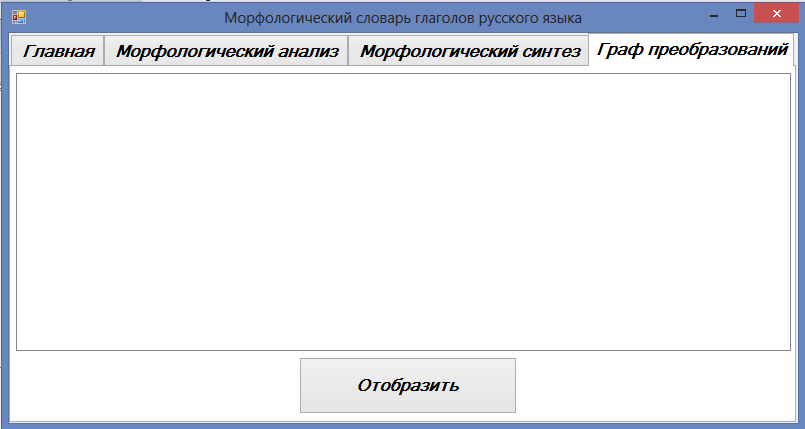


Рис. 5.6 Вкладка «Граф преобразований»

Далее необходимо нажать кнопку «Отобразить». После чего, система отобразит граф преобразования (Рис. 5.7).

РИСУНОК!!!

Рис. 5.7 результат отображения графа преобразований

### Сообщения для пользователя

В процессе эксплуатации системы возможно возникновение исключительных ситуаций, о возникновении этих ситуаций пользователю сообщают соответствующие информационные окна.

В случае, попытки определения грамматических характеристик и основы (инфинитива) словоформы, отсутствующей в словаре, система выдаст соответствующее информационное сообщение (Рис. 5.8).

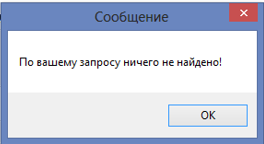


Рис. 5.8 Сообщение об отсутствии информации об искомой словоформе в словаре

В случае попытки генерации словоформы от основы (инфинитива), отсутствующей в словаре, система выдаст соответствующее сообщение (Рис. 5.9).

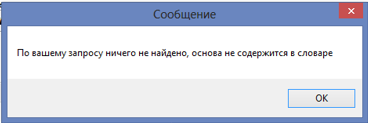


Рис. 5.9 Сообщение об отсутствии основы (инфинитива) в словаре

В случае попытки образования словоформы с грамматическим значением, которым не обладает тип, к которому принадлежит основа (инфинитив), система выдаст соответствующее информационное сообщение (Рис. 5.10).

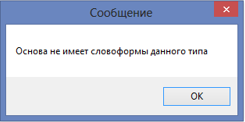


Рис. 5.10 Сообщение об отсутствии у основы (инфинитива) словоформы заданного типа

В случае попытки осуществления задачи генерации или определения с заполнением не всех требуемых полей, система выдаст соответствующее сообщение об ошибке (Рис. 5.11).

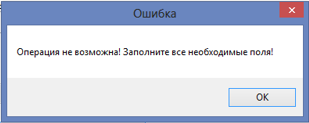


Рис. 5.11 Сообщение о неполном заполнением всех необходимых полей

## ТЕСТИРОВАНИЕ

### Общие положения

Тестирование программного обеспечения (Software Testing) – проверка соответствия между реальным и ожидаемым поведением программы, осуществляемая на конечном наборе тестов, выбранном определенным образом. В более широком смысле, тестирование - это одна из техник контроля качества, включающая в себя активности по планированию работ, проектированию тестов, выполнению тестирования и анализу полученных результатов [16].

При тестировании разработанной информационной системы необходимо использовать такой вид тестирования, как функциональное тестирование.

Функциональное тестирование программного обеспечения является одним из основных видов тестирования, его задачей является установление соответствия исходных функциональных требований заказчика и разработанного программного обеспечения. Таким образом главной задачей функционального тестирования является проверка способности системы выполнять задачи необходимые пользователю. Целью данного вида тестирования является выявление максимального числа ошибок, с использованием небольшого количества входных данных, а также проверка соответствия системы спецификации требований. Функциональные тесты должны охватывать все реализованные функции, с учетом наиболее вероятных типов ошибок.

Главным преимуществом функционального тестирования является то, что оно имитирует фактическое использование системы.

Функциональное тестирование делится на два вида в зависимости от степени доступа к коду тестируемой системы:

* тестирование «белым ящиком»;
* тестирование «черным ящиком».

Тестирование по стратегии «белого ящика» предполагает создание тестовых сценариев, преимущественно основанных на коде системы программного обеспечения.

Тестирование «черным ящиком» проводится без знания внутренних механизмов работы системы и опирается на внешние проявления ее работы [17].

При тестировании разработанной информационной системы было использовано функциональное тестирование по технологии «черного ящика».

Выполняя тестирование методом «черного ящика», пользователь знает только набор необходимых входных данных и ожидаемый результат работы системы, механизм, посредством которого система достигает требуемого результата не известен.

В процессе данного тестирования производится ввод корректных и некорректных данных, и оценивается соответствие реального поведения системы требованиям к ней.

### План тестирования

Основным назначением разработанной информационной системы является автоматизация процесса морфологического анализа и синтеза глаголов русского языка.

Целью данного тестирования является выявление возможных сбоев в системе, проверка ее корректной работы, а также соответствие заявленным требованиям.

Для проверки работоспособности системы проведем следующие тесты, которые позволят оценить разработанную информационную систему:

* поиск грамматического значения и основы существующего в словаре слова;
* поиск грамматического значения и основы не существующего в словаре слова;
* синтез существующей словоформы от основы, содержащейся в словаре;
* синтез не существующей словоформы от основы, содержащейся в словаре;
* синтез словоформы от основы, не содержащейся в словаре;
* отображение графа преобразований.

### Тестирование работоспособности программной системы

Последовательно проведем тестирование системы по выше изложенному плану.

Проведем **первое** испытание.

Данный тест проверяет возможность определения грамматического значения и основы, от которой образована словоформа, причем, полученная основа должна содержаться в словаре.

Попробуем получить необходимый результат. Для этого выберем вкладку «Морфологический анализ», в отобразившейся форме, в поле «Словоформа» введем корректную словоформу:

гуляющий.

Нажмем кнопку «Определить». После нажатия кнопки в полях «Грамматическое значение» и «Основа (инфинитив)» появится информация о словоформе (Рис.6.1).

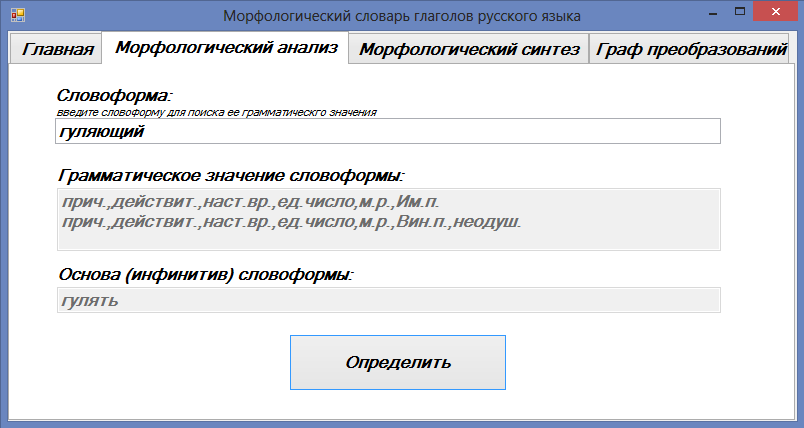


Рис. 6.1 Результат первого испытания

По полученному результату можно сделать вывод, что тест пройден, так как действительно словоформа гуляющий является действительным причастием настоящего времени в единственном числе, мужском роде, именительном падеже и образована от основы гулять.

Проведем **второе** испытание.

Данный тест проверяет невозможность определения грамматического значения и основы, от которой образована словоформа, если, основа не содержится в словаре.

Попробуем получить необходимый результат. Для этого выберем вкладку «Морфологический анализ», в отобразившейся форме, в поле «Словоформа» введем словоформу, основа которого не содержаться в словаре:

красной.

Нажмем кнопку «Определить». После нажатия кнопки, система выведет сообщение о том, что по запросу ничего не найдено (Рис.6.2).

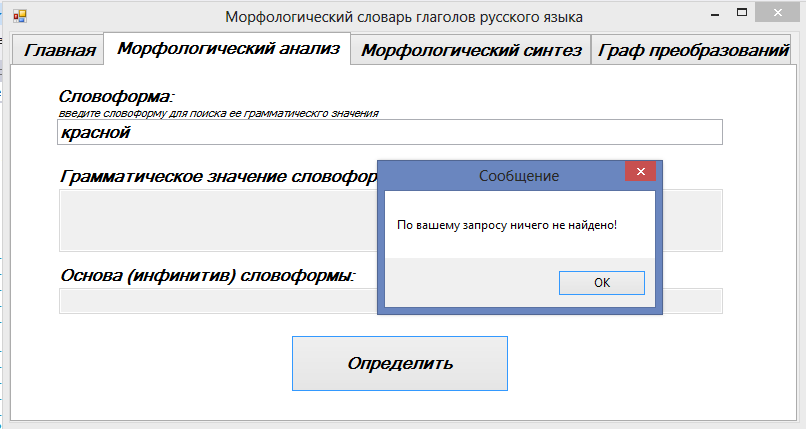


Рис. 6.2 Результат второго испытания

По полученному результату можно сделать вывод, что тест пройден, так как словоформа красной, не является словоформой глагола, следовательно, нахождение ее характеристик и основ невозможно в словаре.

Проведем **третье** испытание.

Данный тест проверяет возможность синтеза необходимой словоформы от основы, содержащейся в словаре, при условии, что у рассматриваемой словоформы существует словоформа с запрашиваемыми грамматическими характеристиками.

Попробуем получить необходимый результат. Для этого выберем вкладку «Морфологический синтез», в отобразившейся форме, в полях ввода ведем следующие данные:

* в поле «Основа/инфинитив» - гулять;
* в поле «Грамматические значение» - гл., наст. вр., изяв. накл., ед. число, 3-е лицо.

Нажмем кнопку «Образовать». После нажатия кнопки, система выведет полученный результат (Рис. 6.3).

По полученному результату можно сделать вывод, что тест пройден, так как словоформой, с грамматическим значением глагола в настоящем времени изъявительном наклонении единственном числе и втором лице действительно является гуляет.

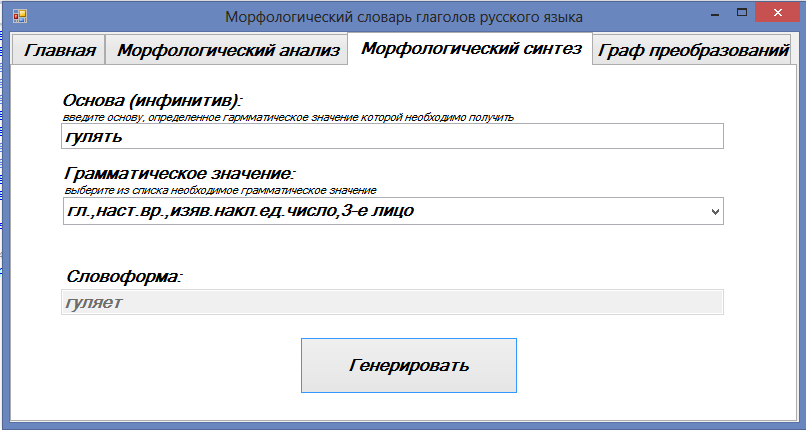


Рис. 6.3 Результат третьего испытания

Проведем **четвертое** испытание.

Данный тест проверяет невозможность синтеза необходимой словоформы, если данный тип основы такой словоформы не имеет.

Попробуем получить необходимы результат. Для этого выбираем вкладку «Морфологический синтез», в отобразившейся форме, в полях ввода введем следующие данные:

* в поле «Основа/инфинитив» - гулять;
* в поле «Грамматические значение» - прич., стардат., наст. вр., ед. число, ж. р., Род.п.

Нажмем кнопку «Образовать». После нажатия кнопки, система выдаст сообщение о том, что у данного инфинитива образование выбранной словоформы невозможно (Рис. 6.4).

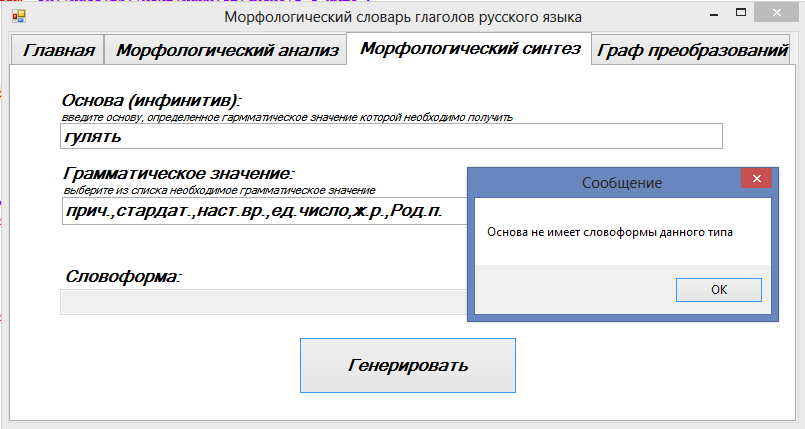


Рис. 6.4 Результат четвертого испытания

По полученному результату можно сделать вывод, что тест пройден, так как у глагола гулять, невозможно образование любых страдательных причастий.

Проведем **пятое** испытание.

Данный тест проверяет невозможность синтеза необходимой словоформы, если ее основа (инфинитив отсутствует) в словаре.

Попробуем получить необходимый результат. Для этого выбираем вкладку «Морфологический синтез», в отобразившейся форме, в полях ввода введем следующие данные:

* в поле «Основа/инфинитив» - траляля;
* в поле «Грамматические значение» - выберем любое значение.

Нажмем кнопку «Образовать». После нажатия кнопки, система выдаст сообщение о том, что преобразование введенной основы (инфинитива) невозможно, так как она отсутствует в словаре (Рис. 6.5).

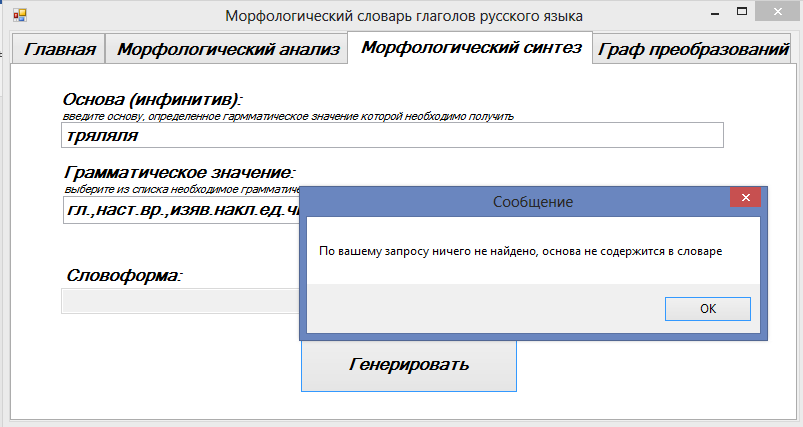


Рис. 6.5 Результат пятого испытания

По полученному результату можно сделать вывод, что тест пройден, так как, введенная основа является произвольной последовательностью символов, и, естественно, она отсутствует в словаре, а значит правил преобразования для нее тоже не существует.

Проведем **шестое** испытание.

Данный тест проверяет возможность просмотра графа преобразования пользователем.

Попробуем получить необходимый результат. Для этого выберем вкладку «Граф преобразований», в отобразившейся форме выберем кнопку «Отобразить».

После нажатия кнопки система отобразит граф преобразований (Рис. 6.6).

РИСУНОК!!!

Рис. 6.6

По полученному результату можно сделать вывод, что тест пройден, так как граф отобразился верно.

Так же следует отметить, что при неполном заполнении необходимых полей данными на всех рассмотренных формах, система выдаст соответствующее сообщение (Рис. 6.7).

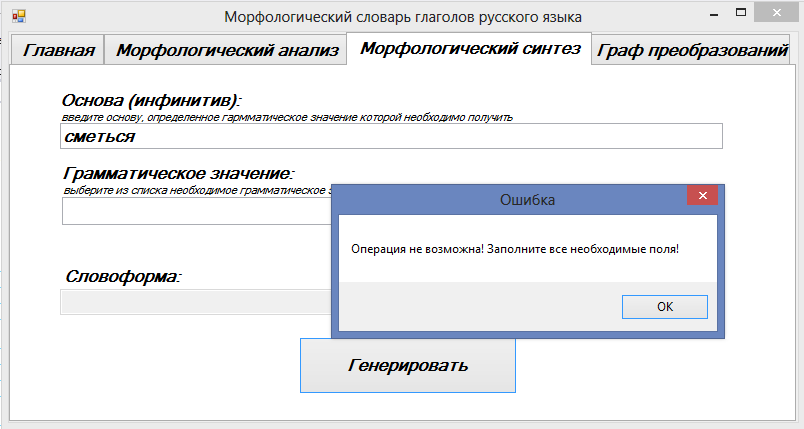


Рис. 6.7 Результат неполного заполнения входных данных

### Анализ результатов тестирования

В ходе тестирования процесс которого подробно описан выше, было выяснено, что разработанная информационная система соответствует необходимым заявленным функциональным требования, а также требованиям безопасности, в случаях некорректных действий выведены соответствующие сообщения. Блокирующих, критических и значительных дефектов в ходе тестирования не обнаружено.

Таким образом, из всего вышесказанного можно сделать вывод, что разработанная информационная система работоспособна и соответствующему уровню безопасности.

# ЗАКЛЮЧЕНИЕ

Данная информационная система разработана для автоматизации процесса морфологического анализа и синтеза глаголов русского языка, и может быть использована любым, кому необходимо решить эти задачи.

В процессе проектирования и разработки информационной системы проводились работы в нескольких направлениях:

* подбор и изучение теоретических материалов в области автоматической обработки текста;
* проведение исследований в области классификации глаголов русского языка по типам формообразования;
* разработка словаря формообразования;
* реализация системы.

Разработанная информационная система позволяет, осуществлять морфологический анализ и синтез глаголов русского в соответствие с типом формообразования. Она поможет любому, будь то школьник, студент или просто человек, желающий проверить себя, получить грамматическую характеристику введенной словоформы и основу (инфинитив), или словоформу по введенной основе и грамматическому значению, в зависимости от того задачу морфологического анализа или морфологического синтеза необходимо решить.

В процессе тестирования системы не было обнаружено блокирующих, критических или значительных дефектов, следовательно, можно сделать вывод, что информационная система готова к использованию.

В дальнейшем возможно расширение и усовершенствование разработанной системы путем добавления словарей формообразования других частей речи. Так же увеличение системы возможно с помощью создания словарей формообразования для других естественных языков.

# БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Пруцков А.В., Яровова О.С. Способ сокращения количества просматриваемых цепочек преобразований при определении форм слов естественных языков// Новые информационные технологии в научных исследованиях: материалы XXI Всероссийской научно-технической конференции студентов, молодых ученых и специалистов. / Рязан. гос. радиотехн. ун-т. – Рязань, 2016. – С.139-141.
2. Яровова О.С. Метод представления цепочек преобразования типов формообразования естественного языка с помощью древовидной структуры как основа справочной системы по морфологии русского языка**//** Современные технологии в науке и образовании – СТНО-2017 [текст]: мат. II между-нар. науч.-техн. и науч.-метод. конф./ под общ. ред. О.В. Миловзорова. – Рязань: Рязан. гос.радиотехн. ун-т.–С.44-45.
3. Морфология.ру [Электронный ресурс]: открытый морфологический словарь русского языка. – URL: http://[www.morfologija.ru](http://www.morfologija.ru) (дата обращения: 10.05.2017
4. UDARENIERU.RU [Электронный ресурс]: грамматический словарь русского языка. – URL: <http://udarenieru.ru> (дата обращения: 10.05.2017).
5. Сазонова И. К. Русский глагол и его причастные формы: Толково-грамматич. сл. – М., Русский язык, 1989. – 590 с.
6. Зализняк А. А. Грамматический словарь русского языка. Словоизменение. Около 100000 слов. – М., Русский язык, 1977. – 880 с.
7. Visual Studio 2010 для профессионалов/ Н. Рендольф, Д. Гарднер, М. Минутилло, К. Андерсон. – М.: Диалектика, 2011. – 1184 с.
8. Microsoft Visual Studio // Википедия: свободная электронная энциклопедия: на русском языке [Электронный ресурс] – URL: <http://ru.wikipedia.org>/wiki/Microsoft\_Visual\_Studio (дата обращения: 19.05.2017)
9. Пруцков А.В. Генерация и определения форм слов естественных языков на основе их последовательных преобразований // Вестник Рязанского государственного радиотехнического университета. — 2009. — № 27. — С. 51-58.
10. Ильвовский Д., Черняк Е. Системы автоматической обработки текстов// Открытые системы. — 2014. — № 1.
11. Селезнев К. Обработка текстов на естественном языке// Открытые системы. — 2003. — № 12.
12. Яровова О.С. Автоматическая обработка текста и ее уровни// ИНТЕЛЛЕКТУАЛЬНЫЕ И ИНФОРМАЦИОННЫЕ СИСТЕМЫ: Материалы Всероссийской научно-технической конференции / Тульский государственный университет. – Тула, 2015. – С. 92-94.
13. Пруцков А.В. Модели, методы и программы автоматической обработки форм слов в естественно-языковых интерфейсах// дис. … док.техн.наук: 05.13.11/ Рязан. гос. радиотехн. унив. — Рязань, 2015. – 279 с.
14. Пруцков А.В. Алгебраическое представление модели формообразования естественных языков // Cloud Of Science. – 2014. – T. 1. – № 1. – С. 88-97.
15. Розанов А.К. Математическое, алгоритмическое и программное обеспечение автоматического предсинтаксического анализа текста в системах управления базами лингвистических знаний// дис. … канд.техн. наук: 07.12.16 /Рязан. гос. радиотехн. унив. – Рязань, 2016. – 117 с.
16. ПроТестинг [Электронный ресурс] – URL : <http://www.protesting.ru/testing/>(дата обращения: 01.06.2017)
17. Функциональное тестирование программного обеспечения// APLANA [Электронный ресурс]– URL: <http://aplana.ru/services/testing/functionalnoe-testirovanie> (дата обращения: 01.06.2017)

# ПРИЛОЖЕНИЕ