Олимпиада школьников «Шаг в будущее»

1477
регистрационный номер

Информатика и системы управления

Программный комплекс семантического анализа текстовой информации.

Автор: Голубева Полина Николаевна
г. Пересвет, школа №5, 11 класс

Руководитель: Пугачев Евгений Константинович
доцент, к.т.н., МГТУ им. Н.Э. Баумана, каф. ИУ6

Консультант: Пешкова Элина Николаевна
программист, ООО НТЦ «ЭРПА»

МОСКВА 2020

Аннотация

В работе представлен программный комплекс, который помогает пользователю формировать и редактировать семантические словари, идентифицировать словоформы на основе использования декларативного метода, проводить статистические исследования текстовой информации на базе частотного анализа, проводить сравнительный анализ текста с целью оценки уникальности и др.

При создании программного комплекса были использованы современные технологии: нисходящий подход разработки, модульный принцип проектирования, объектно-процедурный подход, визуальное и событийное программирование. В целом данный программный продукт разрабатывался по спиральной схеме с использованием метода прототипирования.

В отчете представлены результаты исследования предметной области задачи, результаты основных проектных операций, связанных с разработкой интерфейса с пользователем, обрабатывающих компонент и компонент данных, а также результаты тестирования программного комплекса.

Содержание

Введение

Ввиду массовой доступности информации, с относительно недавнего времени научные и образовательные сообщества стали активно развивать методы выявления заимствования одних идей другими, то есть обратили свое внимание на решение задачи проверки на плагиат, как в научных статьях, так и в выпускных и прочих работах. Особенно остро эта проблема стоит среди ученических и студенческих работ в различных учреждениях. А ведь именно в этих местах люди получают знания и то, насколько они будут полезными и долговечными, напрямую зависит от самостоятельности при выполнении учеником или студентом различных работ.

В интернете существует множество сайтов с программами антиплагиата, но многие из них можно легко обойти просто заменив слова на синонимы. Также эти программы разбивают тексты на фрагменты и ищут совпадения частей текста с представленной информацией на различных сайтах. Анализ показал, что в открытых источниках отсутствует информация о существовании локальных версий программ антиплагиата, которым не нужен доступ к интернету и которыми смогут пользоваться большинство преподавателей с целью создания локальной базы данных для проверки работ студентов.

В данной работе была сделана попытка решить задачу более точной проверки текстовой документации на плагиат. Основной особенностью разработанной программы является возможность отлавливать плагиат, даже если в тексте слова были заменены на синонимы.

Для решения поставленной задачи были выделены следующие этапы:

* Исследование предметной области;
* Выбор и изучение инструментального средства разработки системы;
* Обоснование выбора методов представления информации;

* Разработка алгоритмов работы системы;
* Кодирование, тестирование и отладка системы;
* Оформление отчета по проведенной работе.

К разрабатываемой системе были сформулированы следующие функциональные требования: система должна определять процент заимствования информации представленных преподавателем студенческих работ; позволять редактировать и создавать семантические словари, необходимые для анализа работ; осуществлять поиск и редактирование базы данных текстов.

1 Исследование систем обработки текстовой информации

На начальном этапе исследования были рассмотрены аналоги программ для проверки на плагиат. Было выявлено, что в основном такие программы проверяют на оригинальность, используя тексты, находящиеся в свободном доступе. Особенностью является то, что для работы таких программ нужен доступ к Интернету, что не всегда является приемлемым с точки зрения безопасности и защиты информации.

Рассмотрим несколько популярных программ на проверку уникальности текстов.

Одним из самых известных является сервис Antiplagiat.ru. Эта программа разбивает текст на фразы и затем ищет совпадения по текстам в Интернете. Одним из ее плюсов является то, что программа ищет совпадения не только в Интернете, но и в базе данных студенческих работ Восточной экономико-юридической гуманитарной академии. К сожалению, бесплатно вы можете загрузить текс только раз в 6 минут. Также программа не дает полного отчета об оценке, так как проверяет только по Интернету и базам данных ВЭГА, не затрагивая другие ВУЗы. Конечно, за проверку по базе юридических документов LEXPRO, по коллекции диссертаций РГБ и по базе электронной библиотеки научных работ eLIBRARY можно заплатить от 75 до 900 рублей в зависимости от комплекта покупки. Но даже это не дает гарантии верной оценки на плагиат, так как не учитывается специфика работ ВУЗов различного профиля, где студенты сдают отчеты. Дополнительным ограничением является то, что невозможно проверить текст без регистрации на сайте.

Еще одной известной программой является eTXT Антиплагиат. Процесс проверки устроен точно так же, как во всех аналогичных программах. Все отчеты проверяются только по текстам из Интернета, и при этом, не используются никакие базы данных ВУЗов. Для работы с

программой ее следует скачать и установить, но если вы не хотите скачивать программу, то можно воспользоваться ей онлайн, но для этого нужно заплатить 1,5 рубля за 1000 символов. К тому же требуется обязательная регистрация.

Также была рассмотрена программа Антиплагиат Advego. Эта программа предоставляет многие услуги бесплатно, в отличие от предыдущих программ. Но есть и минусы, программа не предназначена для проверки студенческих работ, так как оценивает уникальность только на основе имеющей в Интернете информации. При этом, программу можно скачивать и устанавливать без регистрации.

На основе проведенных исследований аналогичных программ можно сделать вывод, что почти все программы дают оценку только на основе информации, имеющейся в свободном доступе, где многие из них требуют регистрацию и установку. Кроме выше сказанного, некоторые услуги могут быть платными. Поэтому одной из задач данной работы было создание программы для проверки оригинальности студенческих работ, которая не требует подключения к Интернету и позволяет формировать базу данных работ студентов по различным дисциплинам с целью проверки на плагиат на основе использования методов семантического анализа.

2 Исследование методов обработки текстовой информации

Анализ показал, что существует много методов обработки текстовой информации, из которых только часть можно использовать для проверки на оригинальность.

Одним из таких методов является метод шинглов. Шингл - это фраза, состоящая из последовательно стоящих слов. Основная идея метода состоит в том, что текст разбивается на части, так называемые шинглы, и затем эти части программа ищет с сети Интернет. Чем меньше размер шинглов, тем точнее будет результат проверки. Однако нужно учитывать, что если размер шинглов будет маленьким, то некоторые устойчивые фразы будут разбиты на части, что повлияет на конечные результаты проверки. После того, как программа разбила текст на шинглы, она делает запросы в поисковые системы (Google, Яндекс и т.п.). Ещё одним фактором, влияющим на итоговый результат проверки, является настройка этих запросов, обычно эта настройка называется «Число выборок». Выборка - это небольшая часть исходного текста, которая будет отправлена в качестве запроса поисковику. Результаты работы поисковика затем анализируется программой, то есть скачиваются страницы по ссылкам и сравниваются с исходным текстом. Были выявлены следующие закономерности: чем больше выборка, тем больше запросов к поисковикам, а также тем больше сравнений со страницами из интернета и тем качественней проверка. Также используется еще один параметр как время отклика интернет-страницы. Если скорость соединения слишком мала или страница перегружена, то программа может пропустить и не загрузить для сравнения эту страницу. Так результаты проверки могут быть не точными, но чтобы повысить точность нужно увеличить время ожидания.

Существует еще один метод проверки на оригинальность – метод проверки на рерайт. Текст разбивается на словоформы и затем эти

словоформы сравнивают со словоформами другого текста. Этот метод позволяет выявить перестановку внутри предложений и абзацев, что повышает точность результатов проверки на оригинальность.

Также можно рассмотреть частотный анализ, так как он может помочь при оценке на оригинальность текстовой информации. Частотный анализ предполагает, что частота появления заданной буквы алфавита в достаточно длинных текстах одна и та же для разных текстов одного языка.

Для разработки программного комплекса ПКСАТИ был выбран метод проверки на рерайт, так как другие методы относятся к проверке текстов через Интернет, что не входило в задачи разработки. Также был использован частотный и семантический анализ текстовой информации.

3 Выбор технологии разработки программного комплекса ПКСАТИ

Проведенный анализ показал, что существует множество видов технологий разработки программного обеспечения (ПО). Технология разработки ПО – это совокупность процессов и методов создания программного продукта[1].

В рамках данной работы был рассмотрен ряд технологий разработки ПО. Было выявлено, что одной из распространенных технологий создания программной продукции в настоящее время является структурное программирование, идея которого заключается в том, что структура программы должна отражать структуру решаемой задачи, чтобы алгоритм решения был ясно виден из исходного текста[2].

Структурное программирование может быть использовано при создании средних по размерам программных комплексов или систем (несколько тысяч строк кода). Для создания такой системы необходимо иметь программные среды, которые позволяют отражать конкретную структуру программы. Также необходимо создавать подпрограммы, части кода, которые выполняют определенную функцию и не зависят от других частей кода. Имея множество подпрограмм, можно формировать итоговый алгоритм не из простых операторов, а из цельных блоков, к которым можно обращаться по их именам.

Главный недостаток структурного подхода заключается в следующем: процессы и данные существуют отдельно друг от друга (как в модели деятельности организации, так и в модели программной системы), причем проектирование ведется от процессов к данным. Таким образом, помимо функциональной декомпозиции, существует также структура данных, находящаяся на втором плане.

Еще одной из распространенных технологий создания программой продукции является объектно-ориентированное программирование. В этом

подходе основной категорией объектной модели является класс. Он объединяет в себе данные и операции, которые могут быть с ними связаны (методы). Это и является главным отличием этого подхода от других[2].

Следует выделить еще одну технологию – декларативное программирование. Программный продукт, который включает в себя декларативное программирование представляет собой описание действий, которые необходимо выполнить, а не набор команд. Этот подход обычно формулируется математическими средствами. Поэтому, программы проще тестировать и верифицировать. Такой метод направлен на решение задач искусственного интеллекта[2].

Также, в настоящее время развивается такая технология, как параллельное программирование. Этот метод представляет собой модифицированную версию процедурного программирования, но в отличие от него выделяет одновременно выполняемые последовательности команд. Программы представляют собой совокупность описаний процессов, которые могут выполняться как в действительности одновременно, так и в псевдопараллельном режиме. В последнем случае устройство, обрабатывающее процессы, функционирует в режиме разделения времени, выделяя время на обработку данных, поступающих от процессов, по мере необходимости (а также с учетом последовательности или приоритетности выполнения операций). Языки параллельных вычислений позволяют достичь заметного выигрыша при обработке больших массивов информации, поступающих от одновременно работающих пользователей, либо имеющих высокую интенсивность.

Еще одним методов разработки является процедурный подход. Процедурное программирование — программирование на императивном языке, при котором последовательно выполняемые операторы можно собрать в подпрограммы, то есть более крупные целостные единицы кода, с помощью механизмов самого языка.

В результате было принято решение, что для разработки программного комплекса ПКСАТИ можно использовать процедурный подход к разработке обрабатывающих компонентов, а для разработки пользовательского интерфейса целесообразнее использовать объектно-ориентированный подход.

4 Выбор инструментального средства разработки.

При анализе средств разработки было выявлено, что существует множество инструментальных средств разработки, но не все подходят под выбранную в предыдущем разделе технологию.

Часто используемыми средствами разработки являются языки и системы программирования. Языки программирования можно разделить на две группы: машинные и алгоритмические языки.

Например, машинные языки не подходят, так как содержат машинные команды, соответствующие простейшим операциям обработки и не поддерживают даже процедурного подхода. Машинно-ориентированные языки программирования являются языками низкого уровня, поскольку они требуют хороших знаний внутренней архитектуры компьютеров. Кроме того, программирование на таких языках трудоемко, но в целом программы могут быть наиболее оптимальными. Примерами машинно-ориентированных языков программирования являются различные ассемблеры (Macro Assembler, Turbo Assembler и др.), которые используются в своем классе компьютеров.

С другой стороны языки, которые используются для создания веб-сайтов, например, JavaScript, не подходят, так как стояла задача создать локальную версию программного комплекса ПКСАТИ.

В итоге было уделено особое внимание языкам общего назначения, таким как Visual C++, Delphi и др. Кроме процедурного и объектного подхода такие языки поддерживают модульный подход, а также визуальное и событийное программирование.

Язык программирования C считается более профессиональным, но есть некоторые особенности, которые не способствуют выбору данного языка. В частности, язык C чувствителен к регистру символов, что усложняет его использование начинающим программистом. Например, Button1.Caption – правильное написание кода, а BUTTon1.capTioN – считается ошибкой.

В итоге было выбрано инструментальное средство Delphi, так как в нем имеются следующие важные особенности:

- Прозрачная обработка объектов через ссылки или указатели;
- Свойства как часть языка, вкупе с функциями Get и Set, которые являются прозрачной инкапсуляцией доступа к членам полям;
- Свойства индекса и свойствами по умолчанию, которые обеспечивают доступ к коллекции удобным и прозрачным способом;
- Делегирование реализации интерфейса в поле или свойство класса и др[3].

5 Проектирование программного комплекса ПКСАТИ

5.1 Определение структуры программного комплекса

На начальном этапе проектирования была разработана структурная схема программного комплекса, которая представлена на рисунке 5.1. Из рисунка видно, что программный комплекс состоит из трех частей, которые позволяют обрабатывать исходные словари, формировать и редактировать семантические словари, проводить анализ текста и др.

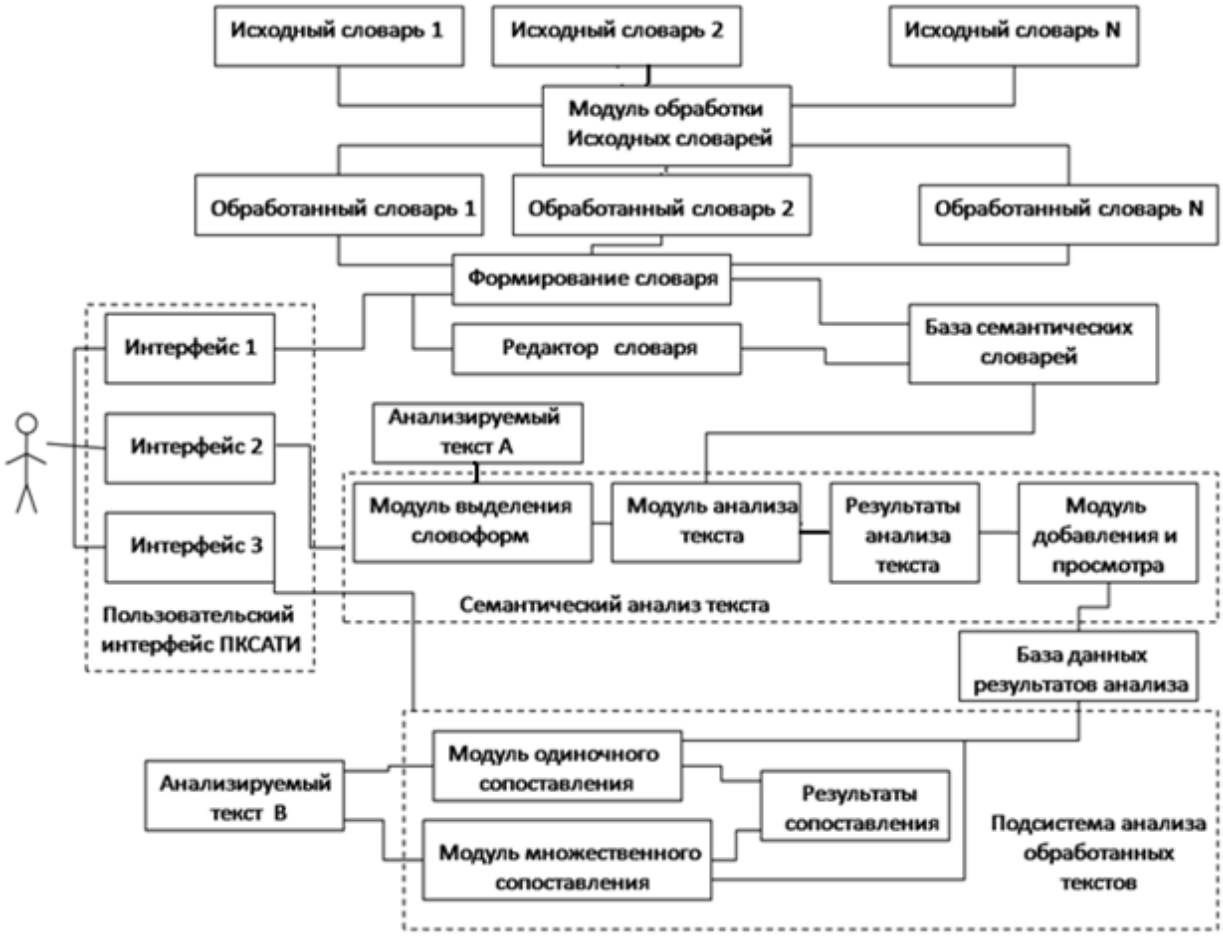

Рисунок 5.1 – Структурная схема программного комплекса ПКСАТИ

Назначение основных компонентов, представленных на структурной схеме следующее:

- Исходный словарь 1, исходный словарь 2 и др. представляют собой обычные словари на естественном языке различного назначения. В частности, это может быть технический, медицинский, финансовый,

экономический и другие словари. При разработке и тестировании программного комплекса ПКСАТИ использовался технический словарь.

- Модуль обработки исходных словарей необходим для исключения избыточной информации, упорядочения словоформ и представления словоформ в удобном для последующей обработки виде.

- Обработанный словарь 1, обработанный словарь 2 и др. представляют собой результаты работы предыдущего модуля.

- Компонент «Формирование словаря» необходим для автоматизации процесса семантического распределения словоформ, конечной задачей которого является создание семантического словаря.

- Редактор словаря необходим для корректировки уже существующих словоформ и добавления новых.

- База семантических словарей представляет собой набор специальных файлов, каждый из которых хранит один из видов словарей.

- Анализируемый текст А и анализируемый текст В представляют собой исследуемые тексты.

- Модуль выделения словоформ отвечает за разбивку анализируемого текста на словоформы и их сохранение.

- Модуль анализа текста на основе выбранного семантического словаря формирует результаты, которые в последствие используются при сопоставлении с целью выявления на заимствование.

- Модуль добавления и просмотра необходим для формирования базы данных результатов анализа с возможностью просмотра как общей информации по анализируемому тексту, так и непосредственно результатов анализа.

- База данных результатов анализа хранит информацию обо всех текстах, которая может быть использована при анализе на плагиат.

- Модуль одиночного сопоставления позволяет выявить процент заимствования для одного указанного текста относительно другого указанного текста.

- Модуль множественного сопоставления позволяет выявить процент заимствования для одного указанного текста относительно всего множества текстов базы данных результатов анализа.

- Пользовательский интерфейс ПКСАТИ состоит из трех частей, которые отвечают за взаимодействие пользователя и программного комплекса с целью выполнения основных функций, таких как формирование и редактирование семантического словаря, проведение анализа текстов, сопоставление результатов анализа и др.

5.2 Разработка диаграммы вариантов использования

При проектировании разрабатываемого программного продукта была разработана диаграмма вариантов использования, которая представлена на рисунке 5.2.

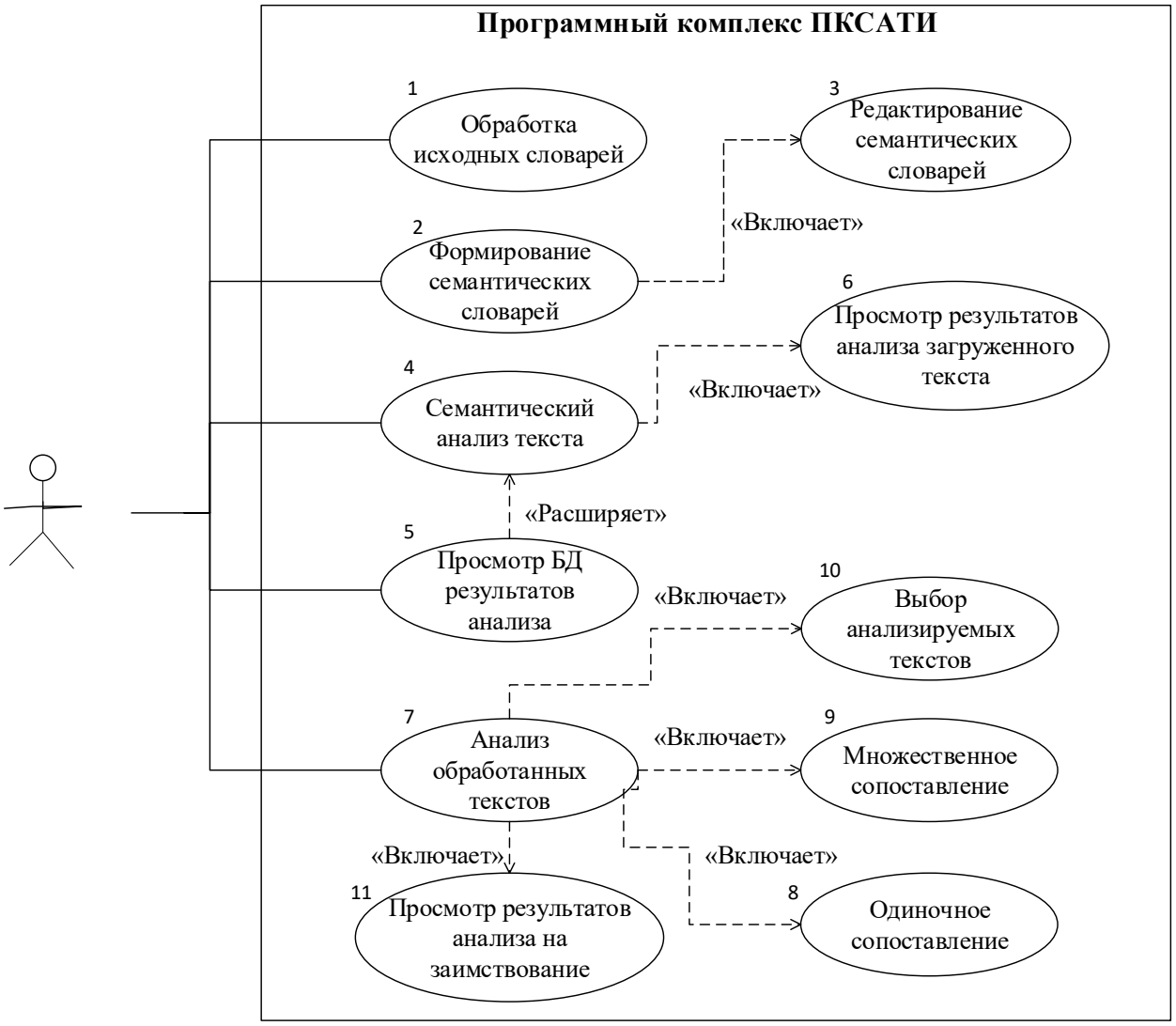

Рисунок 5.2 – Диаграмма вариантов использования программного комплекса ПКСАТИ

При разработке диаграммы вариантов использования были выявлены различные последовательности операций, которые позволяли получать конечные результаты конкретным пользователем. Другими словами, так были определены варианты использования, представленные на рисунке 5.2. Описание вариантов использования представлены в таблице 5.1.

Таблица 5.1 – Описание вариантов использования (начало)

№	Название варианта	Цель	Действующие лица	Краткое описание	Тип
1	Обработка исходного словаря	Сжатие информации	Преподаватель-лингвист	Предоставить пользователю сервис для обработки исходного словаря	Вспомогательный
2	Формирование семантического словаря	Получение специального словаря	Преподаватель-лингвист	Распределить словоформы по смыслу: однокоренные слова и слова синонимы	Вспомогательный
3	Редактирование семантического словаря	Редактирование словоформ	Преподаватель-лингвист	Модификация имеющихся и добавление новых слов	Вспомогательный
4	Семантический анализ текста	Получение информации о тексте	Преподаватель	Получение статистических данных по анализируемому тексту	Основной
5	Просмотр БД результатов анализа	Просмотр и удаление информации о текстах	Преподаватель	Просмотр записей, пометка записей для удаления и др.	Вспомогательный
6	Просмотр результатов анализа загруженного текста	Убедиться о наличии статистических результатов	Преподаватель	Выдать на экран результаты анализа загруженного текста с информацией о частоте встречаемости семантических единиц	Основной
7	Анализ обработанных текстов	Предоставить интерфейс для выбора текстов и сопоставления	Преподаватель	Использовать открытую форму диалога и синхронный способ взаимодействия	Основной

Таблица 5.1 – Описание вариантов использования (окончание)

8	Одиночное сопоставление	Получение результатов сопоставления одного текста относительно другого	Преподаватель	Предоставить возможность выбрать текст и провести анализ на заимствование	Основной
9	Множественное сопоставление	Получение результатов сопоставления одного текста относительно множества текстов	Преподаватель	Предоставить возможность выбрать текст и провести анализ на заимствование со всеми текстами, имеющимися в базе данных	Основной
10	Выбор анализируемых текстов	Задать тексты для анализа	Преподаватель	Предоставить возможность пользователю выбрать тексты для сопоставления	Основной
11	Просмотр результатов анализа на заимствование	Просмотреть результаты и сделать выводы	Преподаватель	Выдать на экран в текстовой форме результаты сопоставления	Основной

В итоге из таблицы 5.1 видно, что 4 варианта использования являются вспомогательными, а 7 являются основными. Для первых трех вариантов использования внешними действующими лицами может быть преподаватель или лингвист.

5.3 Проектирование пользовательского интерфейса

При проектировании пользовательского интерфейса были определены формы интерфейса, состояния интерфейса, а также внешние и внутренние форматы сообщений, способы взаимодействия пользователя и программного комплекса, сценарий диалога и др.

5.3.1 Выбор формы интерфейса и способа взаимодействия пользователя и программного комплекса

Анализ показал, что в современных системах может быть использована как закрытая, так и закрытая форма интерфейса. При разработке комплекса было отдано предпочтение открытой форме интерфейса, которая предполагает манипулирование элементами на экране. В частности используется меню-вектор**.** Достоинствами меню являются: меню ориентированы на пользователя непрограммиста, простое взаимодействие, легкость обучения, так как меню является по существу подсказкой и представляет собой перечень элементов.

При выборе способа взаимодействия пользователя и программного комплекса было учтено, что партнеры диалога активизируются поочередно, что общение четко регламентировано и ни один из партнеров не может прервать другого. Другими словами это свойство синхронного способа взаимодействия. Данный способ широко распространен и реализуется достаточно просто.

Отдельно можно выделить сканирующий вывод системы с оперативным вмешательством пользователя. Здесь характерно наличие периодического изменения содержимого экрана в темпе машинной обработки. Было принято решение, что сканирующий вывод целесообразно использовать в визуализации процессов анализа.

5.3.2 Разработка диаграммы состояний интерфейса

При разработке интерфейса были определены состояния, в которых пользователь может осуществлять различные события, а также были определены возможные переходы из одного состояния в другое и др. На рисунке 5.3 представлена диаграмма состояний пользовательского интерфейса, на которой можно увидеть предоставляемые системой возможности для пользователя, связанные с управлением извне.

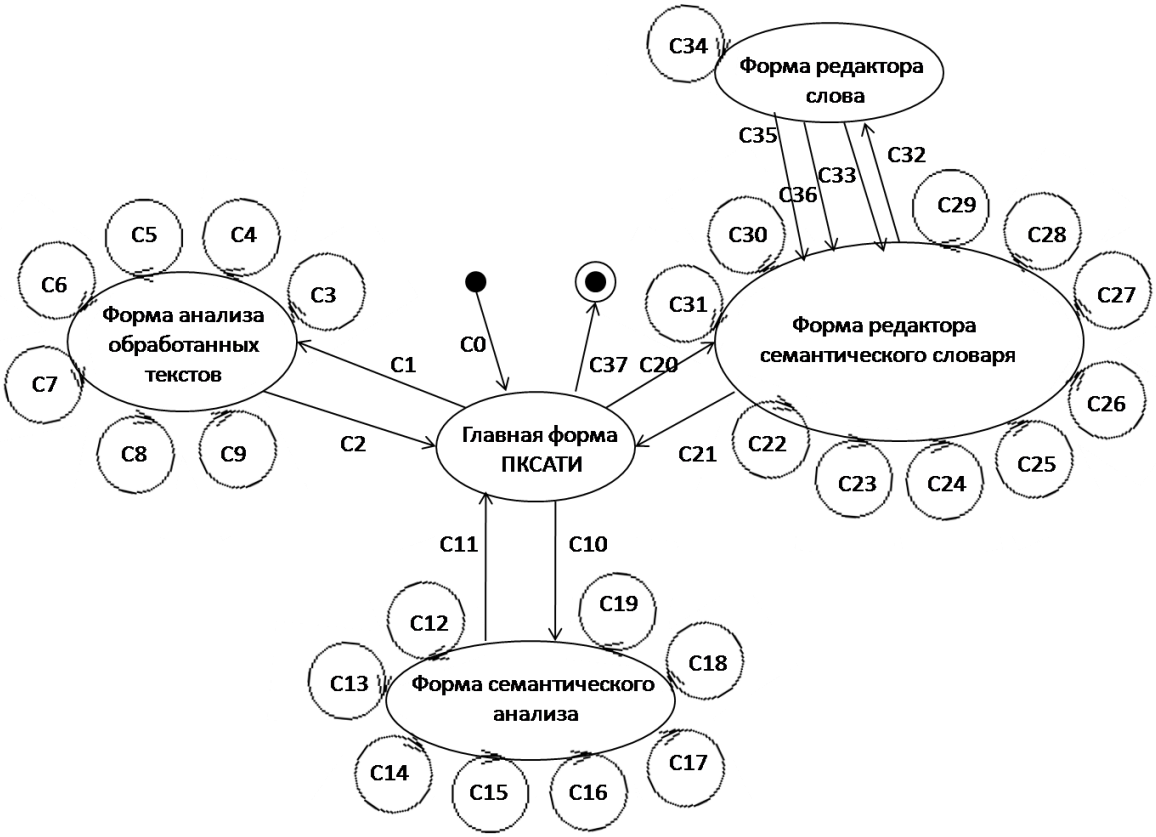

Рисунок 5.3 – Диаграмма состояний интерфейса программного комплекса ПКСАТИ

Из рисунка 5.3 Видно, что ПКСАТИ имеет 5 основных состояний и множество событий, описание которых приведено ниже:

С0 – запуск программного комплекса семантического анализа текстовой информации;

С1 – выбран пункт меню «Анализ обработанных текстов»;

С2 – выход из формы анализа обработанных текстов с помощью стандартной кнопки;

С3 – выбор файла в левом окне;

С4 – выбор файла в правом окне;

С5 – просмотр результатов анализа в левом окне;

С6 – просмотр результатов анализа в правом окне;

С7 – одиночное сопоставление;

С8 – множественное сопоставление;

С9 – просмотр результатов сопоставления;

С10 – выбран пункт меню «Семантический анализ текста»;

С11 – выход из формы семантического анализа с помощью стандартной кнопки;

С12 – загрузка текста и просмотр словоформ текста;

С13 – просмотр статистики по смыслу текста;

С14 – заполнение в поле ФИО автора и другой информации;

С15 – добавление результатов семантического анализа в БД;

С16 – просмотр БД результатов семантического анализа;

С17 – поиск по БД результатов семантического анализа;

С18 – работа с пометками в БД результатов семантического анализа;

С19 – удаление помеченных результатов семантического анализа из БД;

С20 – выбран пункт меню «Редактор семантического словаря»;

С21 – выход из формы редактора семантического словаря с помощью стандартной кнопки;

С22 – загрузка и просмотр исходного словаря;

С23 – создание семантического словаря;

С24 – открытие семантического словаря;

С25 – просмотр семантического словаря;

С26 – перемещение слов из исходного словаря в семантический словарь и наоборот;

С27 – создание новой записи в семантическом словаре;

С28 – сохранение записи в семантическом словаре;

С29 – поиск слова в семантическом словаре;

С30 – очистка окон просмотра семантического словаря;

С31 – выбран пункт меню «Удалить» в окне просмотра семантического словаря;

С32 – выбран пункт меню «Редактировать» в окне просмотра семантического словаря;

С33 – выход из формы редактора слова с помощью стандартной кнопки;

С34 – редактирование слова в поле;

С35 – сохранение слова;

С36 – отмена вызова формы;

С37 – выход из программного комплекса семантического анализа текстовой информации с помощью стандартной кнопки.

5.3.3 Разработка таблиц типичного хода событий

При проектировании разрабатываемого программного продукта были разработаны 3 таблицы типичного хода событий. Таблица типичного хода событий в редакторе семантического словаря представлена в таблице 5.2. Таблица типичного хода событий в модуле семантического анализа текста представлена в таблице 5.3. Таблица типичного хода событий в модуле анализа обработанных текстов представлена в таблице 5.4. В этих таблицах можно увидеть ходы событий в разных модулях программного комплекса.

Таблица 5.2 – Таблица типичного хода событий в редакторе семантического словаря (начало)

Действия исполнителя	Отклик системы
1. Пользователь нажал кнопку «Загрузить».	2. Открытие окна выбора txt файла.
3. Пользователь выбрал txt файл.	4. Вывод файла в окно просмотра слов исходного словаря. Вывод имени файла исходного словаря.

Таблица 5.2 – Таблица типичного хода событий в редакторе семантического словаря (окончание)

5. Пользователь перенес слова из исходного словаря в семантический словарь.	6. Вывод на экран содержания семантического и исходного словарей.
7. Пользователь нажал на кнопку «Создать запись».	8. Вывод на экран новой записи и номера записи.
9. Пользователь нажал кнопку «Сохранить».	10. Форма осталась в прежнем состоянии.
11. Пользователь нажал кнопку пролистывания записей семантического словаря.	12. Вывод на экран соответствующей записи.
13. Пользователь нажал кнопку «Очистить окно».	14. Вывод на экран пустого окна просмотра записей семантического словаря.
15. Пользователь нажал кнопку «Открыть».	16. Открытие окна выбора ss файла.
17. Пользователь выбрал ss файл.	18. Вывод файла в окна просмотра записей семантического словаря и вывод имени ss файла.
19. Пользователь нажал кнопку «Создать».	19. Открытие окна выбора ss файла для его создания.
21. Пользователь ввел слово или его часть в окно поиска и нажал на кнопку «Найти».	22. Вывод на экран слова в соответствующей записи.
23. Пользователь выделил слово в окне просмотра записей семантического словаря и выбрал пункт меню «Удалить».	24. Вывод на экран записи семантического словаря без удаленного слова.
25. Пользователь выделил слово в окне просмотра записей семантического словаря и выбрал пункт меню «Редактировать».	26. Вывод на экран формы редактора слова.
27. Пользователь ввел или изменил слово с поле редактора и нажал кнопку «Сохранить».	28. Закрытие формы редактора слова и вывод на экран записи семантического словаря с измененным словом.
29. Пользователь нажал на кнопку «Отмена».	30. Закрытие формы редактора слова и вывод на экран записи семантического словаря без изменений.

Таблица 5.3 – Таблица типичного хода событий в модуле семантического анализа текста

Действия исполнителя	Отклик системы
1. Пользователь нажал кнопку «Загрузить».	2. Открытие окна выбора txt файла.
3. Пользователь выбрал txt файл.	4. Вывод на экран словоформ выбранного текста, количества слов в файле, количества распознанных слов и информации о статистики текста.
5. Пользователь ввел информацию в поле «Автор и другая информация» и нажал на кнопку «Добавить результаты анализа в базу».	6. Вывод на экран базы данных результатов анализа.
7. Пользователь выбрал один из результатов анализа и нажал на кнопку «Пометить».	8. Вывод на экран помеченного файла и количества помеченных файлов.
9. Пользователь выбрал одну из помеченных записей и нажал на кнопку «Снять пометку».	10. Вывод на экран файла без пометки.
11. Пользователь нажал на кнопку «Удалить помеченные».	12. Вывод на экран неудаленных файлов.
13. Пользователь нажал кнопку пролистывания файлов в базе результатов анализа.	14. Вывод на экран соответствующего файла.
15. Пользователь ввел слово или его часть в окно поиска и нажал на кнопку «Найти».	16. Вывод на экран файла, в котором встретилось это слово.
17. Пользователь нажал на кнопку «Новый запрос».	18. Вывод на экран пустой строки ввода для нахождения нужного файла.

Таблица 5.4 – Таблица типичного хода событий в модуле анализа обработанных текстов

Действия исполнителя	Отклик системы
1. Пользователь выбрал в первом окне выбора файл.	2. Вывод на экран информации из выбранного файла в левом окне.
3. Пользователь выбрал во втором окне выбора файл.	4. Вывод на экран информации из выбранного файла в правом окне.
5. Пользователь нажал на кнопку «Одиночное сопоставление».	6. Вывод на экран результатов одиночного сопоставления.
7. Пользователь нажал на кнопку «Множественное сопоставление ».	8. Вывод на экран результатов множественного сопоставления.

5.3.4 Проектирование форм диалогов

В программном комплексе ПКСАТИ используется 5 формы, где 1 форма вспомогательная и 4 формы основные. Форма программного комплекса представлена на рисунке 5.4.

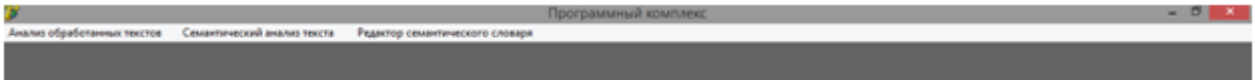

Рисунок 5.4 – Форма программного комплекса ПКСАТИ

При нажатии на пункт меню «Анализ обработанных текстов» открывается форма модуля анализа обработанных текстов, которая представлена на рисунке 5.5.

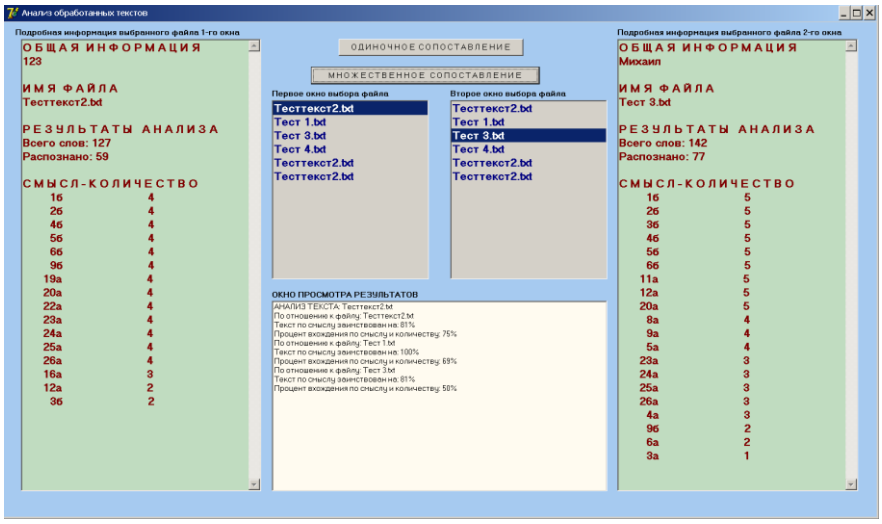

Рисунок 5.5 – Форма модуля анализа обработанных текстов

При нажатии на пункт меню «Семантический анализ текста» открывается форма модуля семантического анализа текста, которая представлена на рисунке 5.6.

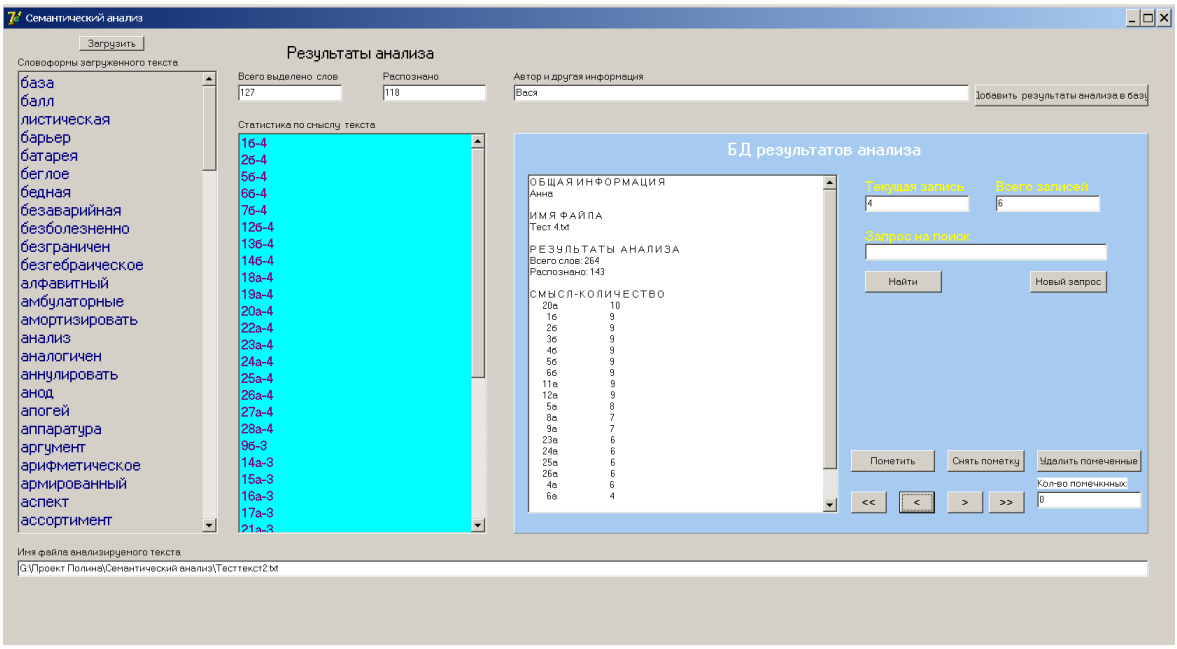

Рисунок 5.6 – Форма модуля семантического анализа текста

При нажатии на пункт меню «Редактор семантического словаря» открывается форма редактора семантического словаря, которая представлена на рисунке 5.7.

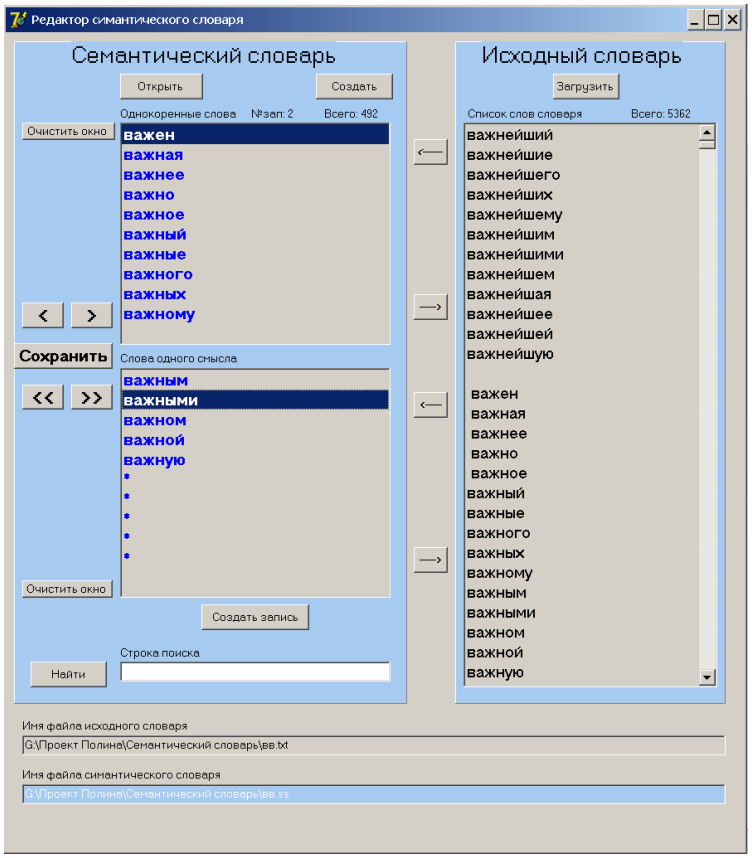

Рисунок 5.7 – Форма редактора семантического словаря

Из этой формы при нажатии на слово и выборе пункта меню «Редактировать» открывается форма редактора слова, которая представлена на рисунке 5.8.

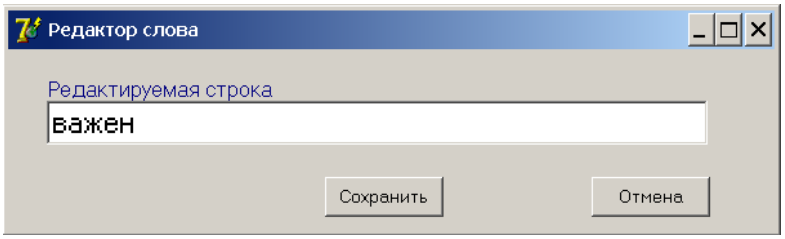

Рисунок 5.8 – Форма редактора слова

5.4 Разработка диаграммы потока данных

При проектировании программного комплекса ПКСАТИ был использован процедурный подход, в котором основное внимание уделялось обрабатывающим компонентам. В соответствии с процедурным подходом были разработаны: контекстная диаграмма потоков данных (рисунок 5.9) и детализирующая диаграмма потоков данных (рисунок 5.10).

Из контекстной диаграммы видно, что основными данными, которые продуцируются преподавателем, как внешней сущностью, являются:

- Имя документа – по существу это является именем файла, который необходимо проверить на заимствование;

- Имя словаря – представляет собой имена файлов словарей различных назначений, которые необходимо подключить для анализа текста;

- Словоформы – представляют собой слова, которые преподаватель добавляет в семантический словарь с целью дальнейшего их использования;

- Информация о документах – включает в себя сведения об авторе и другую информацию о документе.

С другой стороны система выдает пользователю-преподавателю следующую информацию:

- Результаты анализа обработанных текстов – представляют собой конечные результаты с информацией о заимствовании по анализируемому документу;

- Информация семантического словаря – выдается преподавателю для просмотра с целью уточнения смысла слов, их редактирование, добавление и т.п.;

- Результаты семантического анализа – представляют собой промежуточные результаты анализа текста.

После разработки контекстной диаграммы бала разработана детализирующая диаграмма первого уровня, которая представлена на рисунке 5. Из диаграммы видно, что программный комплекс состоит из 7 процессов и 8 накопителей данных.

Назначение процессов и накопителей данных следующее:

- Процесс №1 отвечает за разбор текста на словоформы и удаление повторяющихся элементов. Этот процесс использует данные, которые хранятся в накопителях D11-D1n и представляют собой текстовые файлы с анализируемой информацией. Результаты работы процесса №1 сохраняются в накопителе D4 и представляет собой массив словоформ;

Рисунок 5.9 – Контекстная диаграмма потов данных

- Процесс №2 отвечает за семантический анализ полученных словоформ. В качестве результатов он выдает массив результатов анализа (накопитель D5). Для его работы необходима информация накопителей D4 и D6. Накопитель D6 представляет собой массив семантических слов, который формируется с помощью процесса №5;

- Процесс №3 и процесс №6 отвечают за формирование статистических данных и их упорядочивание. Результаты работы данных процессов сохраняются в накопителе D7, который представляет собой базу данных результатов анализа;

- Процесс №4 отвечает за формирование семантического словаря. Исходными данными являются необработанные текстовые файлы, которые хранятся в накопителях D21-D2R. Результатами этого процесса являются файлы семантического словаря, которые сохраняются в накопителях D31-D3M;

- Процесс №5 отвечает за формирование массива семантических слов, который используется процессом №2;

- Процесс №7 представляет собой итоговый процесс, отвечающий за сопоставление текстов и за формирование результатов на заимствование, которые сохраняются в накопителе D8.

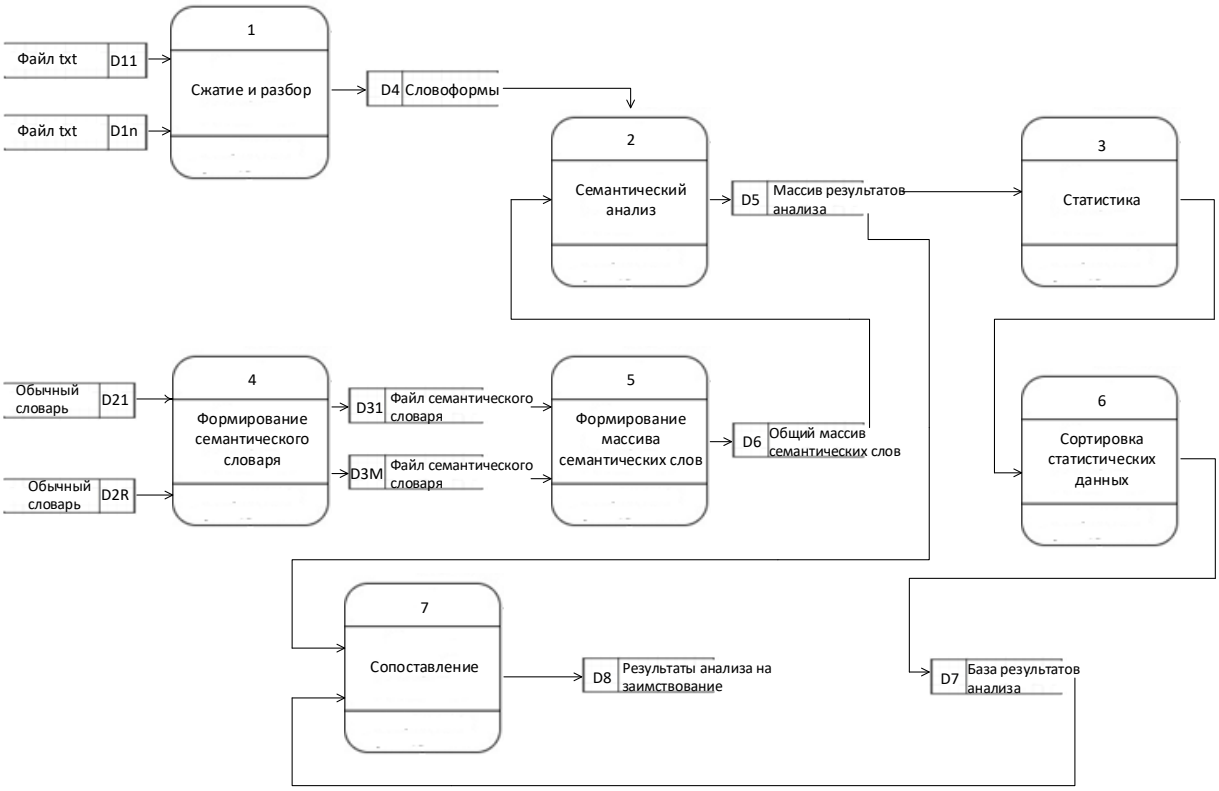

Рисунок 5.10 – Детализирующая диаграмма потоков данных первого уровня

5.5 Разработка основных схем алгоритмов программного комплекса

После получения детализирующей диаграммы потоков данных были разработаны схемы алгоритмов процессов. Основная схема алгоритма представлена на рисунке 5.11.

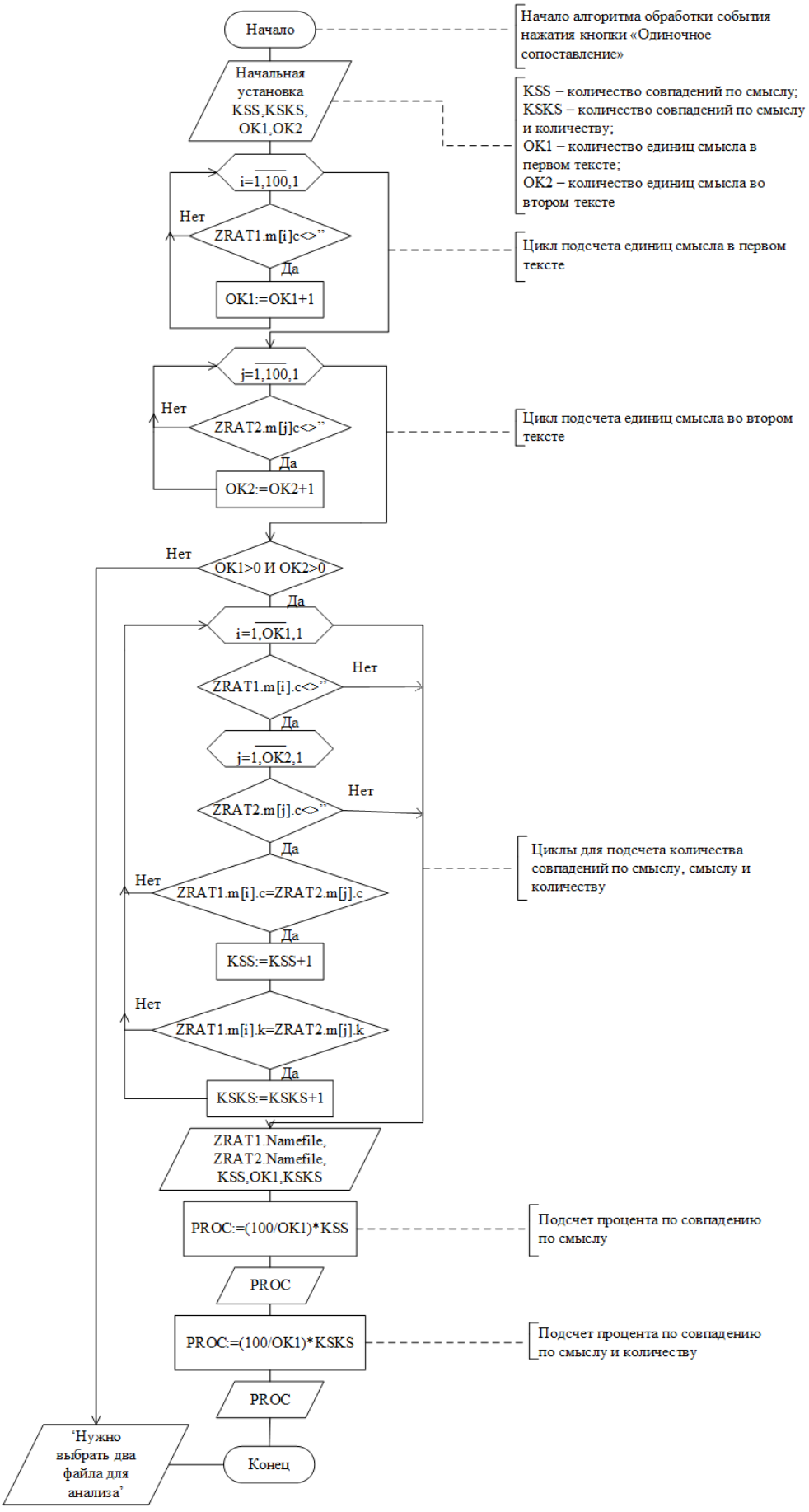

Рисунок 5.11 – Схема алгоритма процесса сопоставления

5.6 Проектирование структур данных

При разработке программного комплекса ПКСАТИ были спроектированы структуры классов всех модулей комплекса. На рисунке 5.12 представлена структура класса редактора семантического словаря. На рисунке можно увидеть, что структура состоит из двух классов TForm1 и TForm2, которые состоят из следующих компонентов:

- ListBox;
- Memo;
- Button;
- OpenDialog;
- ButtonClick.

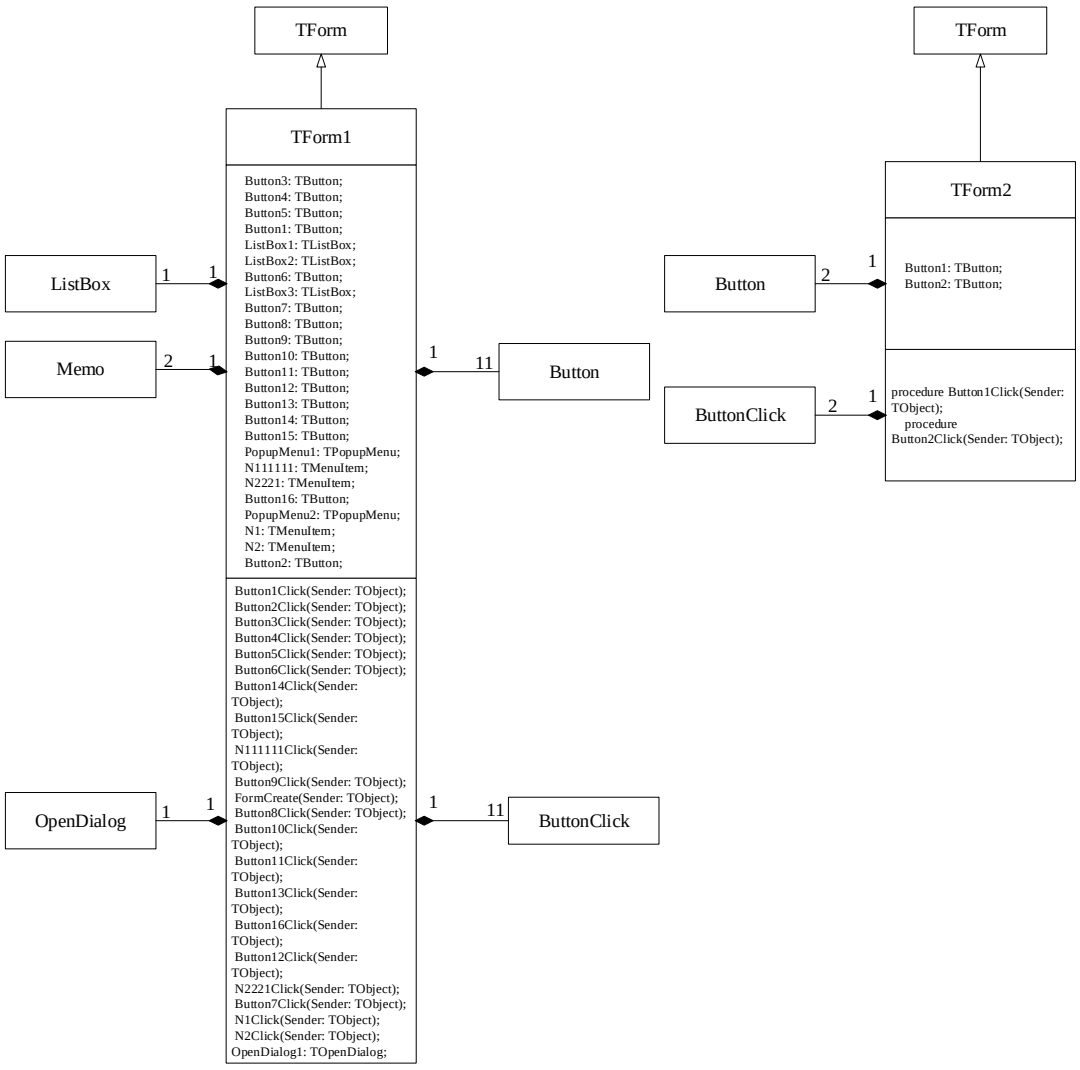

Рисунок 5.12– Структура классов редактора семантического словаря

На рисунке 5.13 представлена структура класса модуля семантического анализа. На рисунке можно увидеть, что структура состоит из одного класса TForm1, который состоит из следующих компонентов:

- ListBox;
- Memo;
- Button;
- OpenDialog;
- ButtonClick.

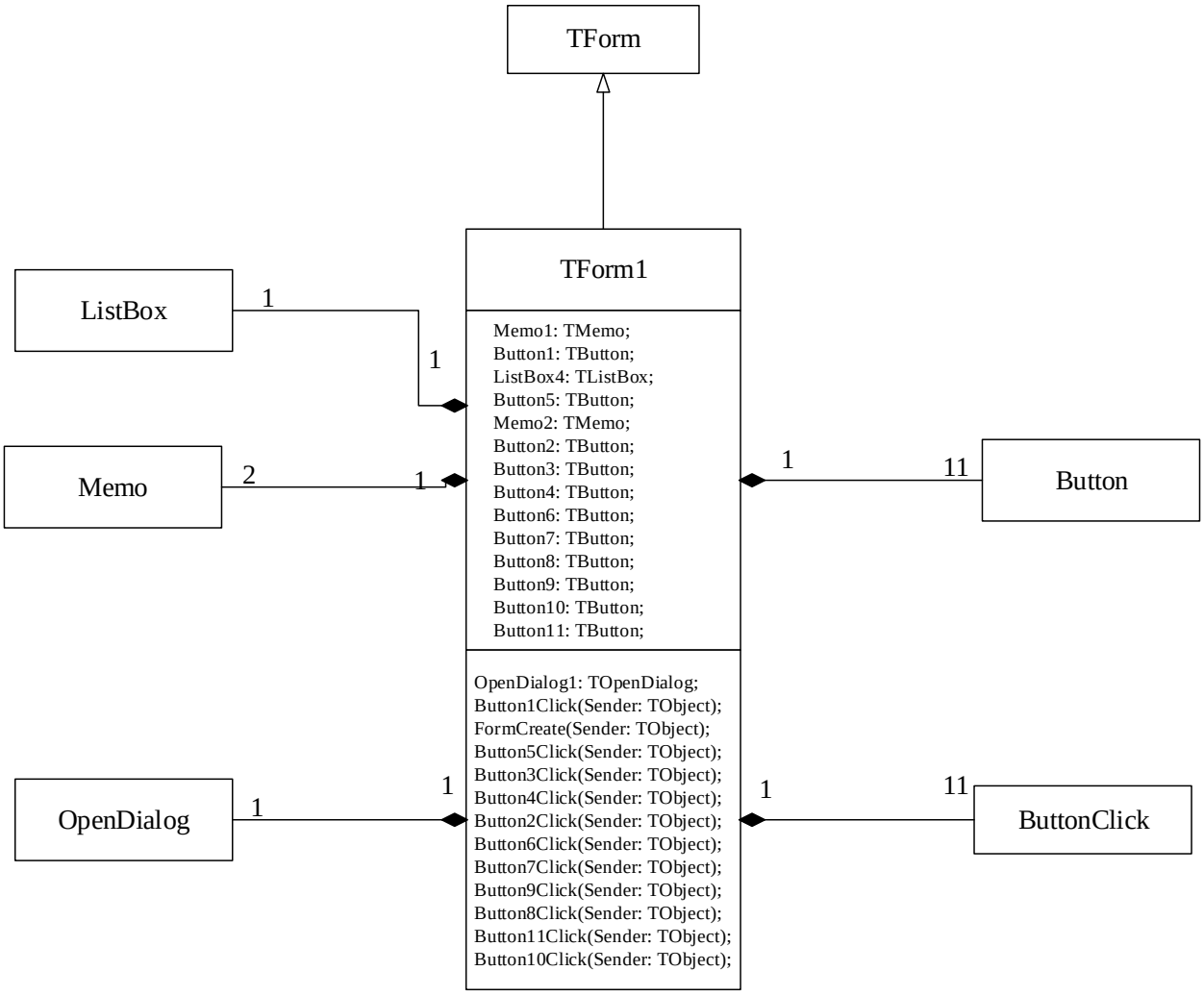

Рисунок 5.13– Структура классов модуля семантического анализа

На рисунке 5.14 представлена структура класса модуля анализа обработанных текстов. На рисунке можно увидеть, что структура состоит из одного класса TForm1, который состоит из следующих компонентов:

- ListBox;
- Memo;
- Button;
- OpenDialog;
- ListBoxClick;
- ButtonClick.

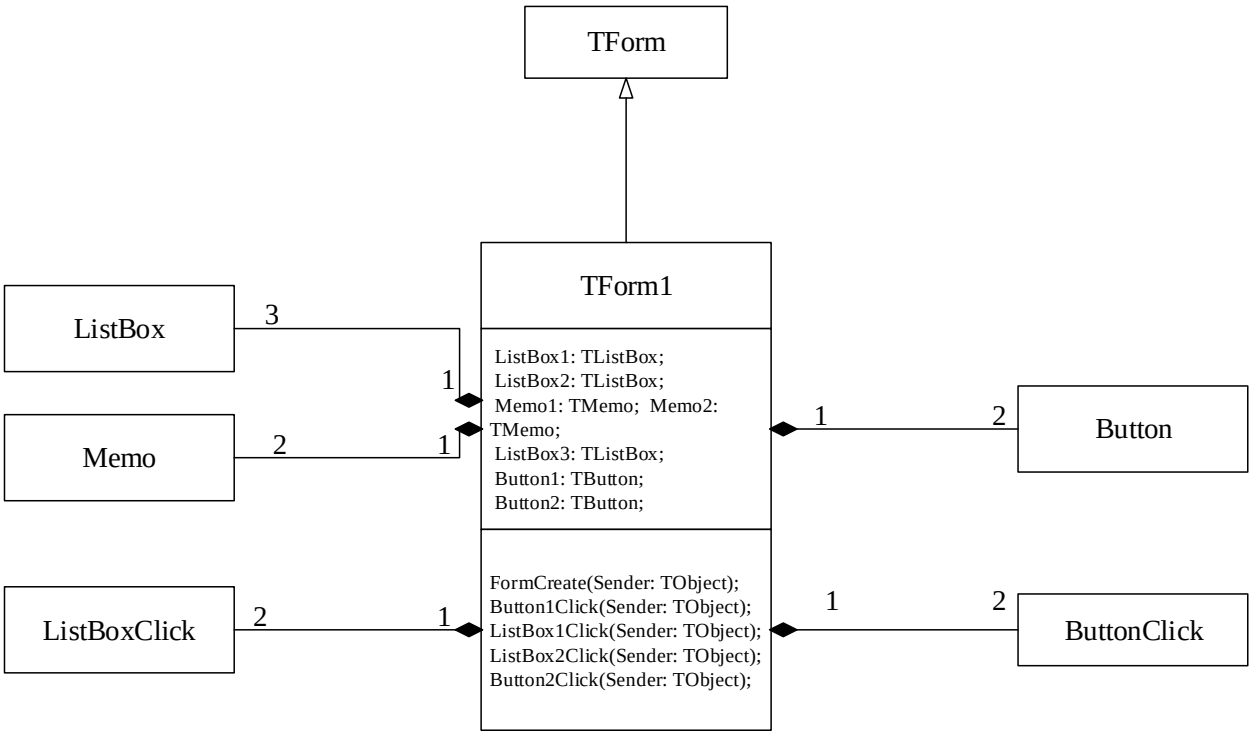

Рисунок 5.14– Структура классов модуля анализа обработанных текстов

После проектирования структур классов были разработаны структуры записей некоторых используемых данных. На рисунке 5.15 представлена структура записи файла результатов анализа. На этом рисунке видно, что основная структура Trat состоит из 5 частей:

- общая информация об анализируемом файле, которая используется для хранения имени автора и другой информации об анализируемом тексте;

- имя файла необходимо для хранения идентификатора анализируемого текста;

- массив результатов анализа представляет собой структуру frec, которая позволяет хранить информацию о каждом слове и состоит из идентификатора слова в семантическом словаре и частоты встречаемости этого слова;

- количество слов в анализируемом файле;

- количество распознанных слов по семантическому словарю.

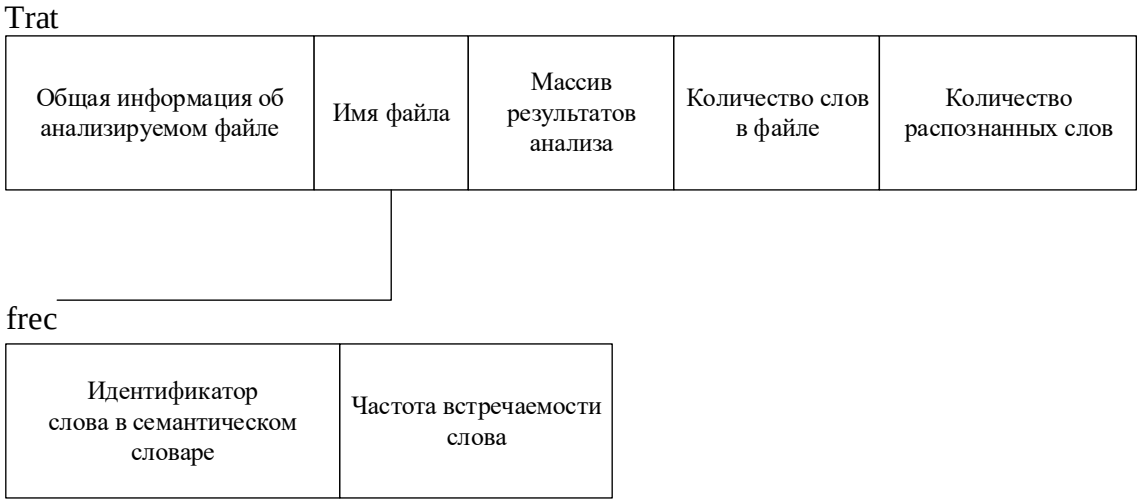

Рисунок 5.15 – Структура записи файла результатов анализа

Ниже представлена часть кода, содержащая структуру, представленную на рисунке 5.15.

```
frec=record c:Tids; k:word; end;
Trat=record    //структура записи файла БД результатов
    Inf:string[100]; //общая информация об анализируемом файле txt
    Namefile:string[100];
    m:array [1..100] of frec; //массив результатов анализа
    all:word; //всего слов в файле txt
    raspoz:word; //кол-во распознанных по семантическому словарю слов
```

На рисунке 5.16 представлена структура записи массива слов семантического словаря. На этом рисунке видно, что структура Tzs состоит из массива слов семантического словаря. Каждый элемент этого массива представляет собой структуру записи Tms, которая состоит из:

- слова;

- первой буквы этого слова;

- идентификатора смысла, который представляет собой номер смысла в семантическом словаре.

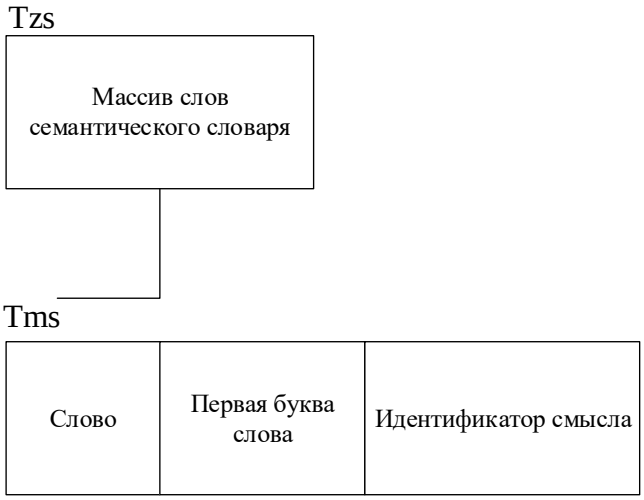

Рисунок 5.16– Структура записи массива слов семантического словаря

Ниже представлена часть кода, содержащая структуру, представленную на рисунке 5.16.

```
Tzs=record
  w:string[50]; //слово
  c:char;       // первая буква
  n:word;       // номер -  в целом идентификатор смысла
  end;
  Tms=array[1..Ks] of Tzs;//семантический словарь в массиве
```

На рисунке 5.17 представлена структура записи семантического словаря. На этом рисунке видно, что структура tzw состоит из массива однокоренных слов и массива слов одного смысла (синонимов).

Рисунок 5.17 – Структура записи семантического словаря

Ниже представлена часть кода, содержащая структуру, представленную на рисунке 5.17.

```
tzw=record
 ms,ns:array [1..10] of string[50];
```

6 Тестирование программного комплекса

Исследования показали, что с точки зрения нахождения ошибок, достаточно эффективными являются методы ручного контроля, которые предназначены для периода разработки, когда программа закодирована, но тестирование на машине еще не началось. Доказано, что эти методы способствуют существенному увеличению производительности и повышению надежности программ и с их помощью можно находить от 30 до 70% ошибок логического проектирования и кодирования.

Основными методами ручного тестирования являются: инспекции исходного текста; сквозные просмотры; просмотры за столом и обзоры программ.

При разработке ПКСАТИ с помощью методов ручного тестирования было обнаружено ряд ошибок, которые связаны со следующими вопросами:

1) Не превышены ли максимальные (или реальные) размеры массивов и строк?
2) Корректно ли осуществляется работа с файлами?
3) Не выходят ли индексы за границы массивов?
4) Соответствуют ли вычисления заданным требованиям точности?
5) Будут ли корректно завершены циклы?
6) Корректно ли отрабатываются ситуации "элемент найден" и "элемент не найден"?

При тестировании комплекса были использованы некоторые методы «белого ящика», в частности, были получены тестовые данные путем анализа логики программы и осуществлялось выполнение отдельных компонентов по всем возможным маршрутам передач управления.

Из методов «черного ящика» использовались анализ граничных значений и анализ причинно-следственных связей.

Особое внимание при тестировании было уделено проверки самой главной функции, т.е. проверки на плагиат при этом проверялось как одиночное сопоставление, так и множественное.

Таблица 6.1 – Таблица результатов тестирования

№ теста	Имя файла	Объем файла	Количество выделенных слов	Количество распознанных слов	Максимальное заимствование из файла	Заимствовано по смыслу (%)	Заимствовано по смыслу и количеству(%)
1	T1.txt	5 Кб	597	110	T2.txt	29	26
2	T2.txt	3 Кб	382	74	T1.txt	100	100
3	T3.txt	5 Кб	1026	262	T1.txt	31	21
4	T4.txt	5 Кб	613	165	T1.txt	36	29
5	T5.txt	5 Кб	590	156	T1.txt	28	25
6	T6.txt	16 Кб	1974	493	T1.txt	64	49
7	T7.txt	3 Кб	307	81	T1.txt	64	63
8	T8.txt	7 Кб	905	192	T1.txt	79	71
9	T9.txt	3 Кб	306	82	T1.txt	79	76
10	T10.txt	11 Кб	1416	296	T1.txt	34	28

Для тестирования были выбраны сайты, на которых имелись инструкции по эксплуатации микроволновой печи. Инструкции были скачены и представлены в виде текстовых файлов. После тестирования программного комплекса ПКСАТИ были получены результаты, которые представлены в таблице 6.1.

Из таблицы 6.1 видно, что информация файла T2.txt полностью заимствована из файла T1.txt, т.е. заимствовано по смыслу на 100%, смыслу и количеству на 100%. Также из данной таблицы видно, что имеется большой процент заимствования и у остальных файлов из файла T1.txt, информация которого была отображена на первом месте по запросу инструкции.

Важно отметить, что количество распознанных слов значительно меньше, чем общее число слов. Можно сделать вывод, что узким местом в программном комплексе является семантический словарь, которому следует уделить особое внимание.

Заключение

В результате проделанной работы были получены:

1. Программная модель программного комплекса семантического анализа текстовой документации.
2. Реализованный прототип программного комплекса ПКСАТИ, состоящий из 3 компонентов: редактора семантического словаря (более 550 строк кода, 13 килобайт), модуля семантического анализа текста (более 510 строк кода, 13 килобайт) и модуля анализа обработанных текстов (более 260 строк кода, 8 килобайт).

В качестве дальнейшего развития системы может быть предложено следующее:

- увеличение количества семантических словарей по различным предметным областям и усовершенствование имеющегося;
- представление результатов в виде текстового файла;
- добавления функций для проверки правильного оформления документации;
- добавления модуля для хранения данных о студентах и их регистрации;
- построение графиков по результатам анализа для визуализации выводов по сопоставлению;
- добавления других способов проверки текстовой информации на заимствование.

Список использованных источников

1. В.В.Бахтизин, Л.А.Глухова «Технология разработки программного обеспечения», издательство БГУИР, Минск БГУИР 2010-266 с.
2. https://life-prog.ru/1_21300_tehnologii-razrabotki-programmnih-kompleksov.html (11.02.2020).
3. http://progaprosto.ru/doc/yazyk_programmirovaniya_delphi.php (21.02.2020).
4. https://studfile.net/preview/2802367/page:5/ (11.02.2020).
5. Г.С.Иванова, Е.К.Пугачев «Оценка методов обработки данных и качества программы», издательство МГТУ им. Н.Э.Баумана, Москва 2015-38 с.
6. Г.С. Иванова, Т.Н. Ничушкина, Е.К. Пугачев «Объектно-ориентированное программирование», издательство МГТУ им. Н.Э.Баумана, Москва 2001-315 с.