# ВВЕДЕНИЕ

В настоящее время работа с технической документацией стала неотъемлемой частью практически любого процесса создания товаров и предоставления услуг. От того, насколько качественно она составлена, зависит эффективность создания продукта и его использование. Техническая документация необходима на производственных предприятиях, при разработке программного обеспечения, при исполнении государственного и т. д.

Однако, не являясь основным видом деятельности, подготовка максимально полной и качественной технической документации занимает значительное время и является сложной задачей.

Анализ инструментов для работы с документации выявил, что существует достаточное количество организаций, предоставляющий услуги по разработке документации. Например, «Философт», «Техническая документация», «ABBYY», «ПроТекст». Большинство из этих организаций при разработке используют стандарты: ГОСТ 2, ГОСТ 19, ГОСТ 34, ISO/IEC 26514, ISO 6592:2000, ISO/IEC 18019:2004 и др.

Помимо существующих стандартов при написании качественной технической документации необходимо учитывать специфику отрасли, знать протекающие в ней процессы и индивидуальные требования. Эти немаловажные обстоятельства значительно увеличивают трудоемкость процесса разработки документации, на который порой тратится времени сравнимо с созданием продукта.

Для автоматизации этого трудоемкого процесса создаются программные инструменты, с помощью которых можно редактировать содержание и структуру документа — текстовые редакторы (MS Word, OpenOffice, LibreOffice и др.), распознавать текст (ABBYY FineReader Engine), генерировать документацию автоматически (Dr. Explain). Так как над документом часто работает не один человек, а коллектив, то для повышения эффективности могут использоваться такие специальные средства, как SharePoint, «Google Задачи», «Планировщик» от Microsoft, «Битрикс24», «Яндекс.Трекер», Jira, SVN, Git и др.

Но зачастую необходимо, чтобы система для работы с документацией давала возможность быстрой обработки и анализа данных, что позволило бы непосредственно взаимодействовать с документацией и оперативно ее редактировать. Существующие программы дают возможность создавать, редактировать, распознавать текст и работать со структурой, но часто требуют глубоких познаний в принципах работы с документацией, т. е. они не решают даже отдельные задачи анализа создаваемой документации с точки зрения ее качества и соответствия требованиям.

Среди множества существующих проблем подготовки документации наиболее распространенными являются такие, как проверка соответствия структуры документации принятым стандартам, требованиям описываемого процесса или предмета, необходимость составления многочисленных однотипных документов, выявление сути документа, оперирование большим количеством сокращений и др. Многие из них могут решаться автоматически, тем самым существенно сократив время специалистов и обеспечив возможность больше времени уделить смысловой составляющей документов.

Целью выпускной квалификационной работы является создание информационной системы для работы с технической документацией, которая позволить упростить и ускорить решение ряда задач подготовки документации.

Для достижения цели в работе необходимо решить следующие задачи:

1. Провести анализ задач и проблем в области подготовки технической документации.
2. Провести анализ требований к технической документации.
3. Провести анализ аналогов и конкурентов приложений и ресурсов, которые позволяют решать задачи обработки документации.
4. Выявить основные проблемы при подготовке технической документации.
5. Разработать требования к информационной системе для работы с технической документацией.
6. Спроектировать информационную систему для работы с технической документацией.
7. Реализовать информационную систему для работы с технической документацией.

Работа состоит из введения, трех глав, заключения, списка использованных источников и N приложений.

Во введении описывается актуальность работы и задачи, которые необходимо выполнить.

В первой главе описывается необходимость подготовки и виды технической документации. Производится анализ принципов разработки, задач, методов и проблем обработки технической документации.

# 1 ГЛАВА АНАЛИЗ ЗАДАЧ И МЕТОДОВ ОБРАБОТКИ ДОКУМЕНТАЦИИ

## 1.1 Необходимость подготовки технической документации

Техническая документация необходима для описания процесса создания товаров и предоставления услуг. Подробное описание позволяет уменьшить финансовые расходы на обучение персонала, повысить его профессионализм и компетентность, оптимизировать деятельность предприятия и снизить ошибки на всех этапах производства продукции.

Техническая документация необходима при проведении сделок с недвижимостью и включает в себя документы технического учета, всевозможные акты осмотров, планы и чертежи, проектно-сметные документы и т.д. Обязанность оформления технической документации на квартиру лежит на ее собственнике, при принятии дома в эксплуатацию. Необходимость предоставления технической документации на квартиру возникает в таких случаях, как купля-продажа, мена, дарение недвижимости, заключение ипотечного договора, проведение перепланировки жилого помещения, проведение экспертиз и оценочных мероприятий, в том числе экспертизы технического состояния квартиры.

Не менее важна техническая документация для автомобилистов. Например, для ремонта необходима техническая документация на ремонт автомобилей, которая включает нормативные, организационные, конструкторские, технологические документы. Комплект документации содержит: карты эскизов, маршрутную или операционную карту, или карту типовой операции, ведомость деталей к типовому технологическому процессу, ведомости технологических документов, оборудования, оснастки и материалов.

При эксплуатации погрузочно-разгрузочных машин важное значение имеет правильное и своевременное ведение технической документации. Техническая документация в значительной мере характеризует уровень технической эксплуатации машин, дает возможность составить точное представление об интенсивности использования и состоянии машин и обоснованно принимать меры по их улучшению. На основании технической документации возможно установить потребность машин в проведении обслуживании или ремонтов, а также решать вопросы оценки технико-экономической эффективности их использования. По материалам документации возможно определять потребности в запасных частях, горюче-смазочных материалах и других ресурсах, необходимых для нормальной эксплуатации парка машин.

Обязательной техническая документация и в сфере пищевой продукции. Сертификация пищевой продукции представляет собой установленную процедуру проверки качества данной продукции и безопасности ее производства с выдачей разрешительных документов – сертификатов и деклараций о соответствии. Стоит отметить, что сертификация пищевой продукции является обязательной, а изготовление, распространение и перевозка пищевой продукции без указанной процедуры является незаконной. Необходимость обязательного проведения сертификации пищевой продукции установлена законодательством РФ, содержащим в себе различные нормативно-правовые и нормативно-технические предписания, правила и требования. В настоящее время действует множество технических регламентов Таможенного союза, которые являются основополагающими в проведении обязательной оценки соответствия, включая сертификацию продукции.

Техническая документация является составляющей проекта по созданию, внедрению, сопровождению, модернизации и ликвидации информационных систем на всем протяжении жизненного цикла. В комплекс технических документов, регламентирующих деятельность разработчиков, входят стандарты, руководящие документы, методики и положения, инструкции. Основным назначением технической документации является обеспечение эффективных процедур разработки и использования информационной системы как программного продукта, а также организация обмена между разработчиками и пользователями ИС. То есть к функциям технической документации относится описание возможностей системы, обеспечение закрепления принятых и реализованных проектных решений, определение условий функционирования ИС, предоставление информации об эксплуатации и обслуживании ИС.

Помимо вышеперечисленных сфер деятельности, техническая документация необходима и для электрооборудования, парфюмерно-косметической продукции, продукции легкой промышленности, продукции санитарно-гигиенического назначения, бытовой химии и прочих видов продукции.

Несмотря на широкий круг применения технической документации, различают несколько видов технической документации.

## 1.2 Виды технической документации

Существует несколько основных видов технической документации: технические задания, конструкторские документации, технические условия на продукцию, обоснования безопасности, паспорта технического устройства, руководства по эксплуатации, технологические инструкции.

### 1.2.1 Технические задания

Техническое задание (ТЗ) – основной документ, содержащий требования заказчика к системе, в соответствии с которыми осуществляется создание и разработка конечного продукта.

Техническое задание необходимо заказчику для того, чтобы понять, что ему необходимо и принять конечный продукт в соответствии с требованиями ТЗ, исполнителю для того, чтобы понять и усвоить поставленную задачу, грамотно спланировать ресурсы, избежать излишней работы над проектом, и конечному потребителю для удобства использования товара.

Разрабатывая техническое задание необходимо придерживаться принципов, которые помогут избежать абстракции в описании будущего товара, а также учитывать интересы заказчика, исполнителя и конечного потребителя.

### 1.2.2 Конструкторские документации

Конструкторская документация представляет собой совокупность документов, в основном чертежей и спецификаций, описывающих изделие или оборудование. Грамотно разработанная документация позволяет узнать всю необходимую информацию о будущем изделии: его устройство, габариты, способы изготовления и необходимые для этого материалы, требования ГОСТа, необходимость контроля и особенности эксплуатации.

Разработка конструкторской документации – это первоначальный процесс перед изготовлением любого изделия или детали, который требует соблюдения многих требований и нормативов. В Российской Федерации разработана и принята единая система конструкторской документации (ЕСКД), в которой прописаны все правила разработки, проектирования, оформления и сдачи документов.

Разработка конструкторской документации на оборудование или изделие на заказ начинается с технического задания, в котором описывается, что именно нужно изготовить. В этом документе также должны быть описаны условия контроля готового изделия и его приемки, и именно после полного заполнения технического задания компания-изготовитель может назвать примерную цену на изготовление требуемого изделия. После составления и согласования этого документа, конструктор может приступать к дальнейшему проектированию.

### 1.2.3 Технические условия на продукцию

Технические условия (ТУ) – документ, содержащий полный список требований к продукции, ее изготовлению, транспортировке и хранению, а также указания по эксплуатации, контролю и приемке. Использование ТУ закреплено Федеральным законом «О техническом регулировании». Технические условия нужны для идентификации продукции, а также для контроля ее качества.

ТУ на продукцию важны для производителя так как документ определяет практически весь процесс производства. На этот документ можно ориентироваться, чтобы определить уровень качества продукции. Фактически, это заявленный минимум, ниже которого компания не может опускаться.

Что же касается потребителя, то он может использовать ТУ для оценки изделия. В частности, именно к этому документу нередко обращаются стороны во время споров, которые способны возникнуть между продавцом и покупателем по поводу качества товара. Если состояние изделия и его характеристики не соответствуют ТУ, то клиент вправе потребовать возврат и компенсацию. ТУ принимается во внимание и во время проведения экспертиз, обычно тоже осуществляющихся в связи с необходимостью защитить права клиента магазина. Таким образом, данный документ для обеих сторон становится своеобразным ориентиром.

Стоит учитывать, что разработка ТУ во многом облегчает решение вопроса интеллектуальной собственности. Так как если компания уже разобралась с ТУ, то ей проще будет доказать, что именно она создала конкретный рецепт или же улучшила определенный технологический процесс, поскольку это все нашло отражение в документах.

Специалисты обращают внимание на то, что наличие ТУ облегчает многие юридические моменты. В частности, теперь можно не расписывать все характеристики продукта, а просто указать данные соответствующего документа. К тому же юристы советуют фиксировать именно ТУ в различных договорах. На основании подобных стандартов можно уже конкретизировать правила перевозки, требования к упаковке или же к хранению.

### 1.2.3 Обоснования безопасности

Обоснования безопасности (ОБ) – это технический нормативный документ, который содержит в себе анализ и оценку рисков, а также данные из документации (конструкторской, эксплуатационной и технологической) о мерах безопасности. Данный документ прилагается к оборудованию или технике на всех их жизненных этапах производства и эксплуатации, дополняемый данными о результатах анализа рисков при использовании данного оборудования или машин после прохождения капитального ремонта.

Подобный документ должен быть создан еще на фазе проектирования оборудования, машин или инструментария. Осуществление разработки должно проводится проектировщиком, производителем данного опасного производственного оборудования, а также сторонними организациями, которые могут привлекаться для разработки технических бумаг.

Разработка обоснования безопасности опасного производственного объекта преследует несколько целей: гарантирование продуктивности и эффективности работы производственного объекта на предприятии; гарантирование безопасности природы; обеспечение здоровья и полной безопасности рабочих на производстве, которые будут контактировать с опасным объектом; обеспечение сохранности остального имущества и технического оборудования компании или организации.

### 1.2.4 Паспорта технического устройства

Паспорт технического устройства представляет собой документ, содержащий сведения, удостоверяющие гарантии изготовителя, значения основных параметров и характеристик изделия, а также сведения о сертификации и утилизации изделия. Согласно ряду нормативных актов эксплуатация технического устройства невозможна без наличия технического паспорта на продукцию.

Паспорт на технические устройства крайне важный документ, особенно на устройства, использующиеся на опасном производственном объекте. Паспорт сопровождает устройство на всем периоде его эксплуатации, в нем содержится подробная информация по устройству, а также вносится вся дополнительная информация, которая появляется в процессе эксплуатации устройства.

В паспорт технического устройства во время эксплуатации устройства заносятся следующие сведения: проведенные технические освидетельствования и диагностики технического устройства; место установки технического устройства; год изготовления и ввода в эксплуатацию; заводской номер; штампы контролирующих органов.

### 1.2.5 Руководства по эксплуатации

Руководство по эксплуатации – это стандартизированный документ, включающий в себя данные о товаре, необходимые для его корректного использования. Также здесь содержится информация для оценки эксплуатационных качеств и решения о целесообразности продолжения пользования изделием.

Этот документ необходим как для людей, непосредственно использующих изделие в процессе своей трудовой или бытовой деятельности, так и для специалистов, выполняющих его сервисное обслуживание. Для некоторых особо сложных устройств в руководстве по эксплуатации выделяется особая часть – инструкция по ремонту, которая нередко выпускается в виде обособленной книги.

Разработка руководства по эксплуатации происходит еще на этапе подготовки предприятия. Обычно наличие этого документа необходимо для получения некоторых разрешительных документов на производство и реализацию продукции. Составляется этот документ по всем правилам оформления технической и конструкторской документации.

Основные требования к заполнению руководства по эксплуатации ничем не отличаются от правил заполнения других технических документов. В данной инструкции должна содержаться исключительно достоверная информация, понятно изложенная и дающая возможность в полной мере понять механизмы и принцип работы устройства. Использование схем, чертежей и графиков делает руководство более понятным и наглядным для целевой аудитории.

### 1.2.6 Технологические инструкции

Технологическая инструкция (ТИ) – одна из разновидностей внутренней технической документации, использующейся в процессе выпуска, использования и ремонта определенного товара. Этот документ входит в состав инструкций, установленных Единой системой конструкторской документации.

Этот документ описывает процесс и технологию производства товара. При этом описание может быть полным либо частичным. Этот документ не требует государственной регистрации, поэтому принимается согласно внутреннему распорядку компании.

Опираясь на стандарты технологической документации, данную инструкцию необходимо разрабатывать для одной или нескольких определенных операций технического процесса. Как правило, в этой ситуации используются типовые документы, принятые в качестве нормативов в данной отрасли. В технологическую инструкцию вносятся следующие данные: условия, необходимые для проведения операции; ее конечная цель; непосредственно процесс и технология ее выполнения.

Описанная процедура в обязательном порядке должна соответствовать технической последовательности и принятым ГОСТам. При необходимости в документ также вносятся корректирующие мероприятия и варианты проверки результатов.

Составление технологической инструкции начинают, как правило, с поиска типовой инструкции типа ГОСТ. На их основе составляется документ, адаптированный под конкретный процесс и предприятие.

Помимо вышеперечисленных видов технической документации на предприятиях возникает необходимость в разработке и использовании и таких документов, как чертежи, спецификации, пояснительные записки, технические отчеты, ремонтные документы, регламенты, руководства, приказы, акты, протоколы и приложения к ним, стандарты предприятия, должностные инструкции и описание бизнес-процессов компании.

## 1.3 Стандарты разработки технической документации

При разработке любой технической документации полезно и зачастую необходимо пользоваться определенными стандартами и нормативами, например, единой системой конструкторской документации (ЕСКД), единой системой технологической документации (ЕСТД), единой системой программной документации (ЕСПД), корпоративными стандартами.

### 1.3.1 Единая система конструкторской документации

Единая система конструкторской документации — комплекс стандартов, устанавливающих взаимосвязанные нормы и правила по разработке, оформлению и обращению конструкторской документации, разрабатываемой и применяемой на всех стадиях жизненного цикла товара.

Основное назначение стандартов ЕСКД состоит в установлении единых оптимальных правил выполнения, оформления и обращения конструкторской документации, которые обеспечивают применение современных методов и средств при проектировании изделий, автоматизацию обработки конструкторских документов и содержащейся в них информации, высокое качество изделий, правильную эксплуатацию изделий, сокращение сроков и снижение трудоемкости подготовки производства.

Стандарты ЕСКД служат основанием для разработки и издания документации, определяющей и регулирующей деятельность, связанную с составлением, обращением и обработкой конструкторских документов, например, положения, устанавливающие структуру и функции технических подразделений предприятий, положения о порядке прохождения и согласовании конструкторской документации, инструктивные материалы по группировке, комплектации, хранению и обработке технических документов.

В ЕСКД предусматривается классификация конструкторской документации, в которую относят графические и текстовые документы. Классификация конструкторской документации осуществляется путем деления на виды: чертежи, схемы, ведомости.

В ЕСКД на все виды изделия составляется конструкторская документация. Она представляет собой схемы, документации, пояснительные записки, технические условия, таблицы расчетов, ремонтные документы.

### 1.3.2 Единая система технологической документации

Единая система технологической документации (ЕСТД) – комплекс государственных стандартов и рекомендаций, устанавливающих взаимосвязанные правила и положения по порядку разработки, комплектации, оформления и обращения технологической документации, применяемой при изготовлении и ремонте изделий.

ЕСТД предназначена обеспечить единообразие в оформлении технологической документации, передачу ее от одного предприятия другому с минимальным переоформлением, создание условий для разработки прогрессивных технологических процессов, создание информационной базы для АСУ.

ЕСТД предусматривает в качестве обязательного документа при всех видах производства составление маршрутной карты, которая предназначена для описания технологического процесса изготовления или ремонта изделия по всем операциям в технологической последовательности с указанием данных об оборудовании, оснастке, материальных и трудовых нормативах в соответствии с установленными формами. Стандартами также предусмотрена ведомость деталей к типовому технологическому процессу, карта типового технологического процесса и ряд других документов, обеспечивающих широкую универсальность ЕСТД.

### 1.3.3 Единая система программной документации

Единая система программной документации (ЕСПД) – комплекс стандартов на программную документацию, которые содержат требования к составу, содержанию и оформлению документов, описывающих программу на разных стадиях ее жизненного цикла, несколько документов описывают порядок хранения и обновления документации.

Стандарты ЕСПД не регулируют, как надо писать документацию, они дают только перечень типов документов и список разделов первого уровня для каждого из них, но о каждом разделе сказано, какие сведения должны быть в нем изложены.

Каждый стандарт ЕСПД имеет небольшой объем и представляет собой набор формальных и легко проверяемых требований к документу или к комплекту документации. Так как ЕСПД четко определяет, из чего должен состоять и как должен выглядеть документ, его легко проверить на соответствие стандарту, что существенно упрощает задачу сдачи-приемки документации как для заказчика, так и для исполнителя.

### 1.3.4 Корпоративные стандарты

При реализации сложных проектов возникает необходимость в создании таких документов, которые бы могли описать конкретные процессы в данной организации, так как они могут быть новыми, усовершенствованными, уникальными для данного проекта. Такие документы, представляющие собой единые правила организации, называются корпоративными стандартами. К ним относятся: стандарты проектирования, стандарты оформления проектной документации, стандарты, пользовательского интерфейса и др.

За основу корпоративных стандартов могут приниматься отраслевые, национальные или международные стандарты. Сюда могут относиться различные методические материалы ведущих фирм разработчиков ПО, научных центов, фирм-консультантов, консорциумов по стандартизации.

## 1.4 Задачи автоматизированной и ручной обработки документации

При подготовке и работе с технической документацией наиболее распространенными задачами являются: ее создание с нуля или на основе шаблона, редактирование содержания, поддержка актуальности.

Для оцифровки старых документов необходимо применять распознавание текста. С этой работой помогают справиться многофункциональные устройства, которые выполняют ее в несколько этапов.

Необходимо ввести отсканированный документ в компьютер, страница в этом случае выглядит как изображение, еще не готовое для дальнейшей работы с ним. Затем производится анализ макета для определения, где на странице находится текст, а где – таблицы и рисунки. Этот процесс выполняется при помощи OCR-приложения, которое позволяет разбить текст на небольшие фрагменты, последовательно дробя их на предложения, слова и, наконец, самые мелкие – символы. Таким образом, конечным результатом данного этапа работы будет совокупность отдельных символов, каждый из которых находится в определенном месте страницы.

Далее программа начинает распознавать символы, т.е. идентифицировать их. От того, насколько правильно пройдет этот процесс, зависит весь результат распознавания. Главная проблема состоит в том, что существуют похожие по своему начертанию символы, которые несут различную смысловую нагрузку.

После распознавания текста начинается реконструкция документа. Программа имеет встроенный словарь, с помощью которого происходит процесс объединения символов в значимые слова, далее – в предложения и абзацы.

Для документов, написанных на иностранном языке, необходим технический перевод. Благодаря своей узкой специфике, этот вид перевода как правило связан с применением специальных знаний в области различных технологий, разобраться в которых под силу лишь специалистам, имеющим значительный практический опыт в данной сфере.

Также к услугам письменного технического перевода прибегают в случае, возникновения потребности в письменном переводе инструкций, техническом переводе текстов с целью создания профессиональных обучающих пособий для производственных специальностей, техническом переводе документов, отчетов, спецификаций, сертификатов, патентов, чертежей, инструкций, научных работ, нормативных актов, иностранной стандартизирующей документации и тому подобных текстов, относящихся к категории сложного перевода. В каждом отдельном случае результаты перевода технических описаний должны удовлетворять специалистов, которым предстоит пользоваться данной технической литературой.

Более узкими задачами обработки документации могут быть архивирование, утверждение, создание ссылок на источники.

## 1.5 Инструменты для решения задач обработки документации

Разработкой технической документации может заниматься штатный сотрудник или команда технических писателей организации. Этот способ используются на крупных предприятиях при высоких объемах производства продукции, необходимости ее постоянной поддержки и частого расширения ассортимента, так как это требует оперативности, детального и широкого познания в соответствующей области.

Предприятие, предоставив все необходимые материалы, может передать работу по созданию и поддержке документации специализированным компаниям. Этот способ применяется при малых объемах производства однотипной продукции. При этом заказчику не нужно иметь в штате постоянного специалиста, а значит и тратить деньги на его содержание. Организации, специализирующиеся на технической документации, имеют в штате опытных специалистов, способных вникнуть в суть многих задач и выполнить их на профессиональном уровне.

Такие организации, как «Философт», «Техническая документация», «ABBYY FineReader Engine» предоставляют услуги по разработке технической документации, оценке качества, консультации и сопровождению, методы распознавания текста и структуры документа.

PhiloSoft Technical Communications (или «Философт») – это компания, которая оказывает услуги по разработке технической документации в области информационных технологий. В основном это программные продукты, аппаратно-программные комплексы, автоматизированные системы и другие ИТ-решения. В их услуги по документированию входят консультации по документированию и рецензирование, разработка документации, технический перевод. Так же компания проводит обучение документированию при помощи тренингов и семинаров, издает учебные пособия.

Компания «Техническая документация» предоставляет услуги по разработке технической документации, в которую входит выбор или уточнение состава комплекта документов, изучение предмета документирования, сбор информации, консультации и поиск нормативных документов по предметной области, обратная разработка техдокументации, определение целей и задач документирования, уточнение требований к оформлению комплекта документов, собственно разработка документации, внесение изменений, согласование и передача ее заказчику. Другой предоставляемой услугой компании является оценка качества технической документации, а именно подготовка экспертного заключения, проверка оформления документов, проверка содержимого разделов документов, проверка структуры разделов документов, проверка терминологии и обозначений физических величин. Так же компания обеспечивает сопровождение разработки техдокументации, проводит консультации в форуме проектировщиков и разработчиков технической документации, семинары по техническому документированию.

Компания «ABBYY» разрабатывает решения в области интеллектуальной обработки информации и лингвистики. Одним из решений является обработка бухгалтерских и финансовых документов: извлечение данных из первичных документов и ввод этих данных в корпоративную информационную систему. Компания предоставляет и готовые решения для обработки информации, например, ABBYY FineReader Engine позволяет встраивать в приложения интеллектуальные технологии распознавания данных, с помощью OCR на основе технологий искусственного интеллекта можно создавать приложения с функциями распознавания информации из документов, изображений, мониторов, определения типа документа, конвертации сканированных документов в файлы форматов Word, Excel и PDF с возможностью поиска. ABBYY Comparator – программа, которая позволяет сравнивать электронные и бумажные документы. Пользователю необходимо загрузить отсканированное изображение документа, текст будет автоматически распознан перед поиском различий с эталонным документом. С помощью программы ABBYY Lingvo можно выполнять перевод отраслевых терминов в тематических словарях по праву, экономике, финансам, маркетингу, банковской и деловой лексике, машиностроению, строительству и архитектуре, нефтегазовой отрасли, химии, медицине, биологии и другим направлениям.

Программное обеспечение «Dr. Explain» позволяет создавать файлы справки, справочные системы, руководства пользователя, пособия и техническую документацию к программному обеспечению и техническим системам при помощи аннотирования экранов приложений. Программа способна анализировать пользовательский интерфейс приложений и создавать скриншоты окон, автоматически расставляя на них пояснительные выноски для элементов интерфейса.

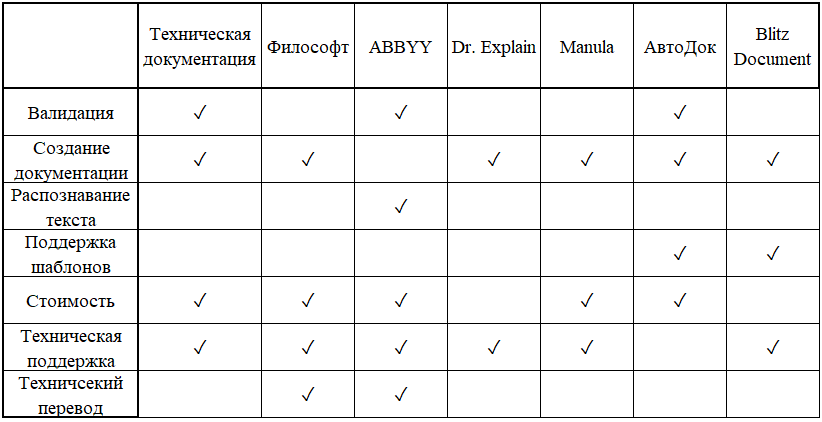
Веб-сервис «Manula» позволяет создавать руководства пользователя, публиковать руководства онлайн, адаптировать макеты для любого устройства или размера экрана. Отсутствует поддержка русского языка.

«АвтоДок» – программа, позволяющая автоматизировать и ускорить процесс заполнения типовых документов посредством созданных с помощью MS Word шаблонов и ведущая архив документов и записей, сделанных при помощи этих шаблонов.

«Blitz Document» – программа для автоматического составления типовых документов посредством шаблонов-сценариев. Позволяет генерировать документы на базе встроенных шаблонов, число которых более 30, или пользовательских сценариев, которые можно разрабатывать на базе встроенных шаблонов или пустых макетов.

Сравнение данных инструментов приведено в таблице 1.

Таблица 1. Сравнение инструментов работы с технической документацией



Все эти инструменты автоматизируют процесс создания технической документации. Одни позволяют упростить эту работу сотрудникам на месте, другие берут весь процесс под свое руководство. Однако, необходимо, чтобы система для работы с документацией давала возможность быстрой обработки и анализа данных, что позволило бы непосредственно взаимодействовать с документацией и оперативно ее редактировать.

## 1.6 Проблемы подготовки технической документации

Существующие программы дают возможность создавать, редактировать, распознавать текст и работать со структурой, но часто требуют глубоких познаний в принципах работы с документацией.

Часто при работе над текстом в силу несовершенств машинных инструментов преобладает ручная обработка, а это кропотливый и долгий процесс, требующий больших трудозатрат.

Существуют и такие проблемы, как несоответствие структуры документации принятым стандартам, требованиям описываемого процесса или предмета, необходимость составления многочисленных однотипных документов, выявление сути документа. Для их решения необходимо проверять структуру на соответствие требованиям или стандартам, создавать документ на основе готового или пользовательского шаблона, реферировать документ.

## 1.7 Выводы

1. Проанализированы виды технической документации: технические задания, конструкторские документации, технические условия на продукцию, обоснования безопасности, паспорта технического устройства, руководства по эксплуатации, технологические инструкции, чертежи, спецификации, пояснительные записки, технические отчеты, ремонтные документы, регламенты, руководства, приказы, акты, стандарты предприятия и др. Выявлены общие проблемы при подготовке документации любого вида.
2. Проанализированы стандарты подготовки документации: ЕСКД, ЕСПД, ЕСТД, которые обеспечивают правильную эксплуатацию изделий, сокращение сроков и снижение трудоемкости подготовки производства. Наличие стандартизированной технической документации необходимо как производителю, так и конечному потребителю для удобства пользования товаром.
3. Проанализированы задачи и инструменты автоматизированной подготовки технической документации. Существующие инструменты сложны для их пользователей, ее подготовка технической документации все еще является трудоемким процессом для технических писателей. Необходимо создание такой информационной системы для работы с технической документацией, которая позволит упростить и ускорить решение типовых задач подготовки документации.

# 2 ГЛАВА ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ РАБОТЫ С ТЕХНИЧЕСКОЙ ДОКУМЕНТАЦИЕЙ

## 2.1 Требования к информационной системе

По результатам проведенного анализа разрабатываемая информационная система для работы с технической документацией должна удовлетворять ряду функциональных и нефункциональных требований.

### 2.1.1 Функциональные требования

Информационная система для работы с технической документацией должна иметь возможности:

1. Поддерживать работу с шаблонами документов.
2. Валидировать структуру документа.
3. Валидировать документ на соответствие шаблону.
4. Поддерживать расширенные правила валидации за счет средств компьютерной лингвистики.
5. Формировать отчет о результатах валидации.

### 2.1.2 Нефункциональные требования

Информационная система для работы с технической документацией должна соответствовать следующим нефункциональным требованиям:

1. Информационная система должна представлять собой оконное приложение для настольных компьютеров.

2. Информационная система должна работать на компьютерах под управлением различных операционных систем.

## 2.2 Формат данных для описания шаблонов документов

### 2.2.1 Описание шаблонов документов

Для описания шаблона используется формат XSD [19]. Формат XSD – это язык описания структуры XML – документа [20]. При использовании XSD многие библиотеки работы с XML имеют возможность автоматически проверять, как правильность синтаксиса XML документа, так и его структуру: соответствие модели документа и заданным типам данных.

В схеме может быть указано несколько пространств имен, например, стандартное пространство имен - "xs", описанное в http://www.w3.org/2001/XMLSchema, для собственного пространства имен используется префикс "mc".

В разрабатываемой ИС корневой элемент шаблона - **"document"** имеет атрибут **"level"** со значением "0" и аннотацию с правилами. Сам элемент имеет составной тип со строгой последовательностью элементов, которые включают в себя ссылку на элемент **"section****"** – описание раздела документа, атрибут **kind** – для связи раздела документа и набора правил, атрибуты "minOccurs" и "maxOccurs" – минимальное и максимальное количество элементов, по умолчанию значение которых равно единице.

Атрибут **"level"**, задающий уровень заголовки имеет простой тип, который описывается глобально, с допустимыми значениям от 0 до 5. Список допустимых значений может быть расширен при необходимости.

Элемент **"section"**, представляющий собой раздел входящего документа, является составным типом, состоящим из последовательности элементов: заголовок **"title"**, содержание **"content"** и подразделов **"section"**. Заголовок должен присутствовать обязательно, а содержание и подраздел опциональны. У данного элемента также необходим атрибут **"level"**.

Для описания типа раздела создан атрибут **"kind"** из собственного пространства имен. Атрибут может иметь любое значение, соответствующее разделу. Так, для раздела «Список терминов и сокращений» используется значение "listOfTermsAndAbbreviations".

Например, описания правил, которые предъявляются к оформлению и структуре выпускной квалификационной работе бакалавра (ВКРБ), для ее различных разделов включают:

1) <xs:element ref="section" mc:kind="intro"/>

где ref="section" – ссылка на описанный глобально элемент "section", kind="intro" – атрибут, указывающий, что раздел является введением.

2) <xs:element ref="section" mc:kind="chapter" minOccurs="3" maxOccurs="3"/>

где kind="chapter" – указывает, что раздел является главой, у которой могут быть подразделы, атрибуты minOccurs и maxOccurs, что таких разделов может встретиться ровно три раза.

3) <xs:element ref="section" mc:kind="conclusion"/>

где mc:kind="conclusion" – атрибут, указывающий, что раздел является заключением.

4) <xs:element ref="section" mc:kind="literature"/>

где mc:kind="literature" – атрибут, указывающий, что раздел является списком использованных источников.

Пример полной схемы XML дипломной работы приведен в Приложении А в Листинге А.1.

### 2.3.2 Описание правил валидации

В аннотации к элементу **"document"** описываются правила для последующей валидации документа. Так как при валидации XML по схеме стандартными средствами языков программирования текст аннотаций, который может быть произвольным, игнорируется, то для извлечения этого текста необходима разработка отдельных алгоритмов.

Шаблон правила состоит из нескольких частей. Сначала указывается тип раздела, затем после ключевого слова **"rules"** – список правил через запятую, название правила и соответствующее значение должны быть разделены знаком равенства, правила для раздела заканчиваются символом «;».

В общем виде правила выглядят следующим образом:

*first\_section\_kind* rules: *rule1*=*value*, *rule2*=*value*;

*second\_section\_kind* rules: *rule1*=*value*, *rule2*=*value;*

Например, правила для разделов ВКРБ имеют вид:

*intro* rules: *title*=*ВВЕДЕНИЕ*, *sectionQuantity*=1;

*conclusion* rules: *title*=*ЗАКЛЮЧЕНИЕ*, *sectionQuantity*=*1*;

*literature* rules: *title*=*СПИСОК**ЛИТЕРАТУРЫ*, *sectionQuantity*=1

Полный список правил для шаблона ВКРБ приведен в Приложении Б в таблице Б.1.

## 2.4 Алгоритм валидации структуры документа

Для успешной валидации структуры документа необходимо пройти несколько этапов проверки.

На первом этапе создается **модель документа**, при этом проверяется, что в исходном документе для заголовков применены стили, а последовательность заголовков непротиворечива, т.е. заголовки имеют последовательную вложенность и в целом, укладываются в древовидную структуру для всего документа.

На втором этапе созданная модель проверяется на соответствие шаблону. Для этого на основе модели генерируется XML-файл, который проверяется на соответствие схеме. Для успешного прохождения необходимо, чтобы документ полностью соответствовал разработанной для него XSD-схеме.

Например, для шаблона ВКРБ, приведенного в листинге А.1 Приложения А, проверяемый документ должен иметь ровно шесть разделов, из которых один предназначен для «Введения», три - для глав с подразделами, один для «Заключения» и один для «Списка использованных источников».

## 2.5 Алгоритм валидации документа по правилам

После проверки модели документа производится считывание из шаблона справочника правил из схемы и списка разделов, к которым необходимо применить правила.

Блок-схема алгоритма проверки представлена на рисунке 1.

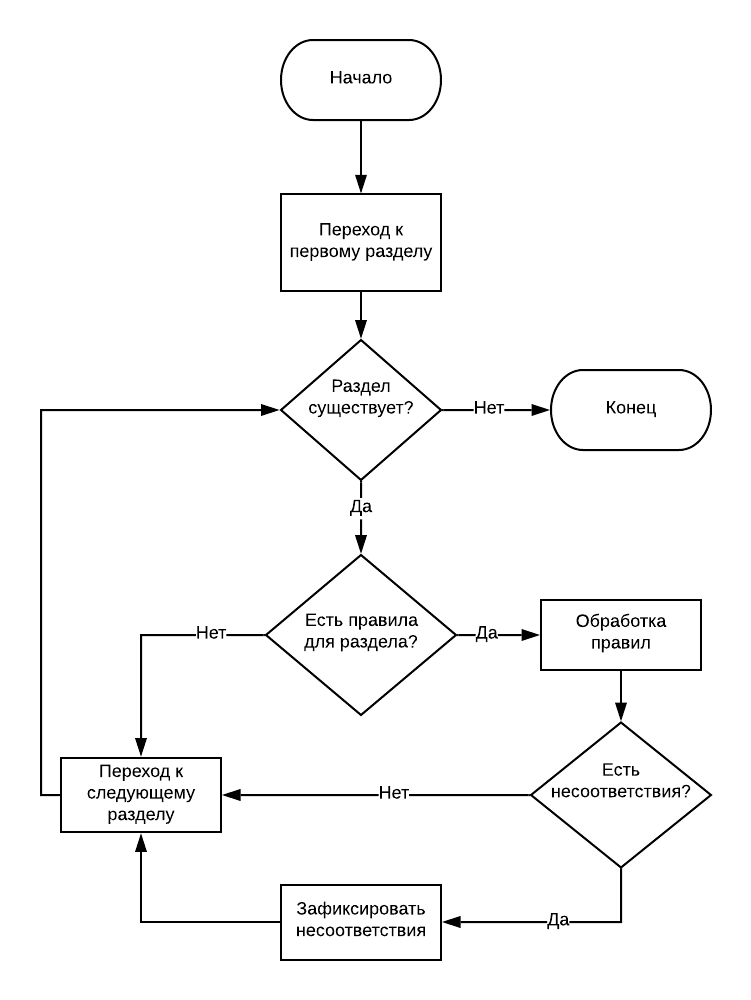


Рисунок 1 – Схема алгоритма валидации алгоритма по правилам

Например, для первого раздела шаблона ВКРБ существуют правила, в соответствии с которыми его заголовок должен быть строкой «ВВЕДЕНИЕ», и в разделе не должно быть подразделов, если значения в документе соответствуют требуемым, осуществляется переход к следующему разделу, иначе несоответствия фиксируются, после чего осуществляется переход к следующему разделу.

Полный список правил для шаблона ВКРБ приведен в Приложении Б в таблице Б.1.

В результате проверки список ошибок содержит описание несоответствий заданным правилам, либо может быть пустым в случае успешной проверки.

## 2.6 Алгоритм валидации аббревиатур

Если в справочнике присутствует правило, требующее проверки аббревиатур, то находится соответствующий раздел, и данная проверка выполняется в следующем порядке:

1. Создается список аббревиатур и их расшифровок, который заполняется значениями из указанного в шаблоне документа раздела, содержащего список аббревиатур и сокращений, используемых в документе.

2. Осуществляется выделение аббревиатур из всего текста документа. Для выделения встретившихся аббревиатур выполняется графематический анализ выделенных аббревиатур и текста,

3. Выполняется морфологический анализ выделенных аббревиатур чтобы отфильтровать написанные заглавными буквами слова, например, «ИЗДЕЛИЕ» или «ПРИЛОЖЕНИЕ».

4. Сравниваются списки выделенных из текста аббревиатур и описанных в соответствующем разделе документа. Все аббревиатуры из текста должны быть найдены в списке аббревиатур, несовпадения заносятся в список ошибок.

## 2.7 Проверка содержимого раздела документа

Если в справочнике есть правило, указывающее на необходимость проверки содержимого разделов документа, то при помощи средств компьютерной лингвистики выполняется соответствующий анализ.

Например, если в шаблоне документа для раздела указаны ключевые слова, то производится автоматическое выделение ключевых слов из текста раздела и их сравнение с заданными в шаблоне.

В итоговый отчет выводится процент совпадения ключевых слов. В дополнение выводятся списки заданных слов в шаблоне и выделенных для того, чтобы пользователь знал что ожидалось и что встретилось в его документе.

Набор ключевых слов позволяет охарактеризовать содержимое документа, что в дальнейшем можно использовать для выявления дублирующихся или сильно пересекающихся по смыслу разделов.

## 2.8 Выводы

1. Определены функциональные и нефункциональные требования к информационной системе для работы с технической документацией. Система должна позволять создавать шаблоны различных типов документов, поддерживать валидацию структуры документа, обеспечивать валидацию документа на соответствие шаблону.

2. Определен формат данных для описания шаблонов документов. Шаблон документа описывается схемой в формате XSD. Схема состоит из описания структуры разделов документа и правил их валидации.

3. Разработана система правил валидации документов на основе справочника правил, расположенного в аннотации XSD элемента, и формат их описания. Правила могут быть применены к определенным разделам, например точное совпадение текста заголовка с требуемым или количество подразделов.

4. Разработаны алгоритмы валидации структуры документов. Для валидации структуры документа создается его модель, на основе которой генерируется XML-файл. Этот файл затем проверяется на соответствие разработанной XSD-схеме, после чего документ обрабатывается по правилам из справочника.

5. Разработан алгоритм для дополнительной проверки аббревиатур документа. Могут быть проверены встретившиеся в документе аббревиатуры на их наличие в списке терминов и сокращений. Списки выделенных из текста аббревиатур и описанных в соответствующем разделе документа сравниваются. Все аббревиатуры из текста должны быть найдены в списке аббревиатур.

6. Разработан алгоритм для дополнительной проверки содержимого раздела документа при помощи средств компьютерной лингвистики. Рассчитывается процент совпадения ключевых слов, который позволяет охарактеризовать содержимое документа, что можно использовать для выявления дублирующихся или сильно пересекающихся по смыслу разделов.

# 3 ГЛАВА. РАЗРАБОТКА СИСТЕМЫ ДЛЯ РАБОТЫ С ТЕХНИЧЕСКОЙ ДОКУМЕНТАЦИЕЙ

## 3.1 Инструменты разработки информационной системы

Для разработки системы были использованы следующие средства:

1. Объектно-ориентрованный язык программирования Java. Программы на этом языке транслируются в байт-код, который выполняется на виртуальной java-машине. Виртуальные java-машины реализованы под многие операционные системы, поэтому код, написанный на языке Java, может выполняться повсеместно.

2. Инструмент для автоматизированной сборки проекта Apache Maven. Maven дает возможность сборки проекта в любой операционной системе, позволяет управлять зависимостями, то есть использовать сторонние библиотеки, разрешать конфликты версий.

3. Набор инструментов для работы с текстом TAWT, поддерживающего основные этапы анализа текста: графематический, морфологический и семантико-синтаксического, объединенных общим интерфейсом инструментов, и имеющих поддержку русского языка.

4. Библиотека Apache POI, которая позволяет обрабатывать документы Microsoft Office. Содержит классы и методы для распознавания файлов документов MS Office. Для чтения и записи файлов расширения docx в MS-Word используется компонент XWPF.

5. Фреймворк JavaFX для создания приложений с графическим интерфейсом. Имеет набор элементов управления, и возможности по работе с мультимедиа, двухмерной и трехмерной графикой, описания интерфейса с помощью FXML, возможность стилизации интерфейса с помощью CSS.

## 3.2 Архитектура системы

Диаграмма компонентов системы представлена на рисунке 2.

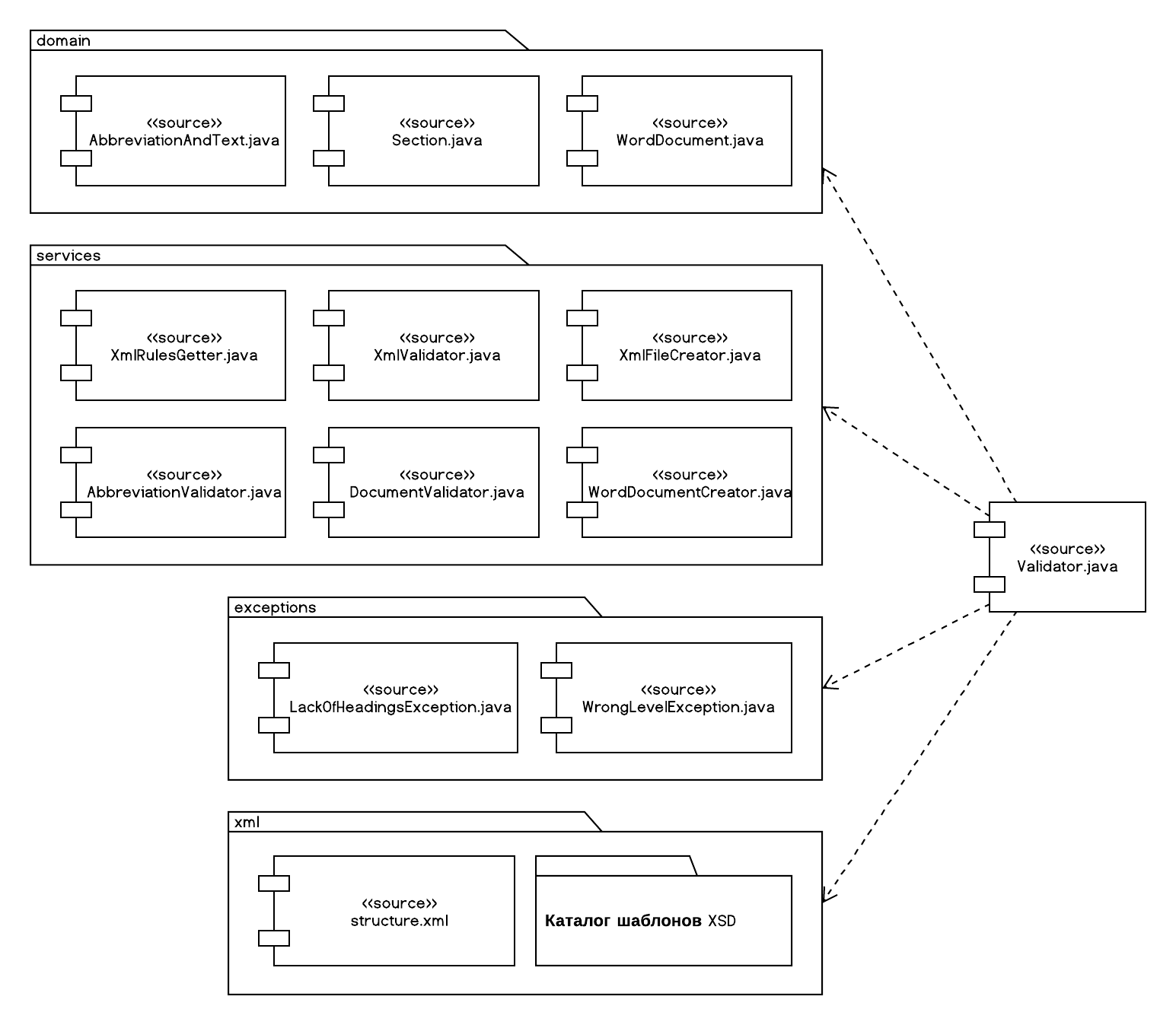


Рисунок 2 – Диаграмма компонентов системы

При первой инициализации документа в формате .docx создается его модель WordDocument, обладающая такими параметрами, как название файла, его описание, корневой раздел и список всех разделов.

Список всех разделов состоит из объектов Section, модели, описывающей раздел, имеет такие параметры, как уровень раздела, заголовок, список подразделов, родительский раздел, содержимое и другие.

Список аббревиатур и их расшифровок формируется моделью AbbreviationAndText.

Класс WordDocumentCreator обеспечивает методом создания модели модели WordDocument.

Класс DocumentValidator предназначен для валидации документа по правилам из схем методом validate.

Класс XmlFileCreator создает структурированную модель XML на основе WordDocument.

Класс AbbreviationValidator обеспечивает методами валидации аббревиатур.

Класс XmlRulesGetter предназначен для работы со схемой XML. Метод getDocumentRules генерирует из аннотации справочник с правилами в виде Map, с ключом, описывающим какому разделу, принадлежит правило, и значением Map с правилами к разделу. Метод getSectionKind генерирует список типов разделов в документе.

Файл structure, на основе которого генерируется xml со структурой документа пользователя, имеет несколько пространств имен и корневой тег "document". Файл schema – схема XML, шаблон документа, включает правила для валидации.

Исключения, которые могут возникнуть при валидации: ошибка, возникающая, если в документе пользователь не использовал стили для заголовков – LackOfHeadingsException и ошибка, возникающая, если в документе пользователь нарушил последовательность заголовков – WrongLevelException.

Диаграмма классов системы представлена в Приложении В.

## 3.3 Реализация алгоритмов валидации

### 3.3.1 Алгоритм создания объектной модели документа

В метод createNewDocument передается путь к проверяемому файлу, а возвращается объект WordDocument со структурой разделов.

1. На основе входящего docx создается документ XWPFDocument, который содержит текст входящего документа и его стили. Текст представляет собой список параграфов XWPFParagraph, а стили – объект XWPFStyles.

2. Для дальнейшего построения структуры объекта WordDocument, определяется текущий уровень – 1, текущий родительский раздел – корневой раздел WordDocument, и пустой текущий раздел.

3. После того, как пройдена подготовка, происходит перебор параграфов. У каждого параграфа проверяется, что у его стиля есть ID, по которому определяется конкретный стиль параграфа.

4. Проверяется, что стиль c данным ID существует и начинается со слова "heading", значит данный параграф – заголовок, иначе – простой текст, содержимое раздела. При условии, что получен заголовок, создается раздел Section, у которого заголовок – это текст текущего параграфа, а уровень – ID стиля.

5. Далее сравнивается полученный уровень раздела и текущий уровень:

– равны, значит это новый раздел того, же уровня, которому присваивается текущий родительский раздел, а ему в список подразделов – текущий раздел.

– текущий уровень меньше на единицу, значит это новый подраздел, если он меньше более, чем на две единицы – в документе ошибка, о чем сообщается пользователю, работа создания структуры прекращается, пока пользователь не исправит ошибку. Если ошибки нет, то у раздела родительским становится текущий раздел, которому присваивается этот подраздел. Текущим родительским разделом становится текущий раздел. Текущий уровень увеличивается на единицу.

– текущий уровень больше, значит подразделы закончились, необходим переход на более высокий уровень. Для этого до тех пор, пока уровень текущего родительского раздела не будет больше либо равен уровню раздела, ему присваивается его корневой раздел, у текущий уровень уменьшается на единицу. После перехода на необходимый уровень выше, у раздела корневым становится текущий родительский, ему раздел добавляется в список подразделов.

6. Раздел добавляется в список разделов документа и становится текущим.

### 3.3.2 Алгоритм создания XML-файлаl

Для создания XML на основе модели WordDocument используются методы класса XmlFileCreator.

В метод createNewXmlTree передается объект WordDocument, из которого извлекаются разделы первого уровня. Из файла structure.xml извлекается корневой элемент "document". Каждый раздел документа Section поочередно преобразовывается в элемент и добавляется в корневой.

Преобразование раздела в элемент осуществляется при помощи метода constructSection. Создается элемент "**section**", указывается его атрибут – уровень раздела. К нему добавляются дочерние элементы "**title**" – заголовок раздела и, если раздел имеет содержимое "**content**". Так преобразуется раздел первого уровня, если у него есть подразделы, то для каждого снова создается элемент, который записывается дочерним разделу первого уровня. То есть метод constructSection вызывается рекурсивно до тех пор, пока у раздела количество подразделов не будет равным нулю.

Пример сгенерированной XML приведен в Приложении Г листинге Г.1.

### 3.3.3 Алгоритм загрузки справочника правил

Класс XmlRulesGetter обеспечивает приложение методами, с помощью которых возможно пользоваться правилами, предназначенными конкретному типу технической документации. Для этого используется XPath из стандартной библиотеки Java и обработчик DOM из библиотеки ORG.W3C.DOM.

XPath – это язык запросов к элементам XML-документа. Для того, чтобы получить интересующие данные, необходимо создать запрос, описывающий эти данные, который выполнит интерпретатор языка XPath.

DOM-обработчик считывает сразу весь XML и сохраняет его, создавая иерархию в виде дерева, по которой можно двигаться и получать доступ к нужным элементам.

Метод getDocumentRules извлекает из аннотации схемы к элементу "document" правила, а возвращает Map, у которой ключ – это раздел, а значение – Map с названием правила и значением. Например, ruleMap с правилами для разделов дипломной работы будет выглядеть так:

key = conclusion

value = {subsectionQuantity=0, title=ЗАКЛЮЧЕНИЕ}

key = literature

value = {subsectionQuantity=0, title=СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ}

key = intro

value = {subsectionQuantity =0, title=ВВЕДЕНИЕ}

где key – раздел, а value – правила.

Для того, чтобы извлечь правила в таком виде, схема приводится к объекту XmlObject, из которого извлекается модель Document. Методом evaluate в XPath передается запрос ".//element[@name='document']/annotation/appinfo/text()", то есть необходимо найти элемент, у которого имя "document", его внутренний элемент с тегом "annotation", у которого найти тег "appinfo" и взять его текст. В результате выполнения запроса создается строка с текстом внутри тега appinfo:

intro rules: title=ВВЕДЕНИЕ, level=1, sectionQuantity=0;

conclusion rules: title=ЗАКЛЮЧЕНИЕ, level=1, sectionQuantity=0;

literature rules: title=СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ, level=1, sectionQuantity=0

Из строки извлекаются правила и распределяются по контейнеру, для чего используется метод *split*. Извлекается массив строк, разделенных по символу ";". Каждый элемента массива снова делится по строке "rules: ", при этом первая подстрока это ключ, который является разделом ("intro") – ruleType, а вторая – список правил нему ("title=ВВЕДЕНИЕ, level=1, sectionQuantity=0").

Список правил делится по строке ", ". Каждый элемент полученного массива делится по символу "=", после чего первый элемент — это конкретное правило ("title"), а второй – его значение ("ВВЕДЕНИЕ"). Полученный массив состоит из двух элементов и заносится в currentMap.

После того, как получен тип раздела и контейнер с правилами к нему, они заносятся как ключ-значение в справочник: ruleMap.put(ruleType, currentMap).

Методом getSectionKind в список строк извлекаются типы разделов из схемы. Для этого запросом ".//element[@kind='chapter']/@minOccurs" в переменную chapterQuantity заносится количество глав, имеющих подразделы. Затем в список sectionKindList запросом ".//element[@ref='section']/@kind" заносятся значения атрибута "kind". Если встретился элемент со значением атрибута "chapter", он добавляется такое количество раз, сколько было указано в переменной chapterQuantity, иначе элемент добавляется только один раз.

После обработки sectionKindList содержит следующий список:

intro

chapter

chapter

chapter

conclusion

literature

### 3.3.4 Валидация XML-файла по XSD-шаблону

Валидация XML по схеме xsd обеспечивается методом validate класса XmlValidator с помощью стандартных средств Java из библиотеки javax.xml.

Метод принимает на вход файл XML и файл xsd. Файл схемы конвертируется в объект класса Schema. Создается объект Validator из полученной схемы методом newValidator. Метод *validate*, который принимает файл XML, осуществляет проверку на соответствие схеме.

В случае успешной проверки будет дан ответ true, иначе возникнет исключение SAXException, в котором указано в какой строке XML ошибка.

### 3.3.5 Валидация по правилам из справочника

Валидация по правилам из справочника обеспечивается методом validate класса DocumentValidator, который использует созданную модель документа, и справочник правил из схемы. Метод возвращает пустой список, если не найдено ошибок, либо список с описанием несоответствий.

Так как на этапе создания модели проверено, что структура документа соответствует схеме, то для каждого раздела первого уровня проверяется, есть ли для него соответствующее правило в справочнике. То есть для структуры дипломной работы будет верно, что его первый раздел «Введение», поэтому если в справочнике найдется правило для этого раздела, то оно будет обрабатываться. Иначе – переход к следующему разделу и поиск правил для него.

Когда найден раздел, для которого есть набор правил, он извлекается из справочника. Каждый пункт правил обрабатывается.

Например, если для раздела «Заключение», указано количество подразделов (conclusion rules: subsectionQuantity=0), то будет проверен размер подразделов «Заключения». Если выявлено несоответствие, в список ошибок заносится его описание в виде "Wrong subsections quantity of section *title*: *current quantity* expected: *expected quantity*".

### 3.3.5 Проверка содержимого раздела документа

Если в справочнике есть правило, указывающее на необходимость проверки содержимого разделов документа, то она реализуется при помощи библиотеки для проверки текста по ключевым словам [21]. Из класса KeyWords используются методы:

getWordsInText – метод для получения всех слов текста. Параметром является исходный текст. Метод возвращает список слов типа List<Word>.

getKeyWords – метод для получения всех ключевых слов текста интегральным методом. Параметром является исходный текст. Метод возвращает список ключевых слов типа List<Word>.

В итоговый отчет выводится процент совпадения ключевых слов. В дополнение выводятся списки заданных слов в шаблоне и выделенных.

## 3.4 Создание шаблона документа

Шаблон для документа создается на основе технического задания или иных нормативных документов. Для создания шаблона документа необходимо:

1. Определить список разделов первого уровня.

2. Выделить одинаковые по структуре и с отдельными требованиями разделы, определить для них значение атрибута kind.

3. Определить возможные значения уровня, указать их в описании типа levelType.

4. Описать раздел в общем виде.

5. Для элемента document добавить в <xs:sequence> разделы в виде последовательности элементов section.

6. Добавить правила для конкретных разделов в <xs:appinfo>.

7. Добавить общие для всего документа правила в <xs:appinfo> .

Заготовка шаблона приведена в Приложении Д листинге Д.1.

## 3.5 Интерфейс системы

На основе требований к системе был реализован следующий интерфейс информационной системы для работы с технической документацией. (Рис. 1-2)



Рисунок 1 – Интерфейс приложения

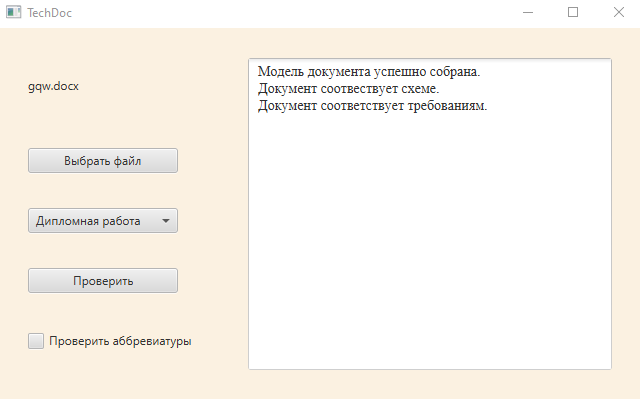


Рисунок 2 – Отчет о проверке

Пользователь может выбрать файл в формате docx для проверки, после чего отображается название файла.

Пользователь может выбрать тип работы, то есть какой шаблон будет применять для валидации, и применить настройку валидации аббревиатур.

По нажатию на кнопку «Проверить» пользователь получит отчет о валидации документа.

**Это все надо явно подробнее, понятно, что реализации сейчас не хватает.**

## 3.6 Применение и развитие системы

Система для работы с технической документацией может применяться для создания шаблонов различных типов документов, валидации структуры документа, валидации документа на соответствие шаблону.

В дальнейшем планируется ввести справочник шаблонов, создание новых на основе предыдущих и новых пользовательских шаблонов. Так же планируется расширить список обрабатываемых правил, включая проверку стилей текста.

Добавить сортировку отчетов об ошибках по их типам, сравнение разделов по ключевым словам, проверку на антиплагиат, применение сравнения текстов, например, реферата и краткого описания содержания в правилах.

## 3.7 Выводы

1. Выбраны инструменты для сборки проекта – Apache Maven, набор инструментов для работы с текстом TAWT, библиотека Apache POI, платформа для реализации интерфейса JavaFX.

2. Разработана архитектура системы. Создана модель документа, раздела, аббревиатур. Реализованы сервисы по созданию моделей и валидации и собственные исключения.

3. Реализованы алгоритмы создания объектной модели документа, создания XML-файла, загрузки справочника правил, валидации XML-файла по XSD-шаблону и правилам из справочника.

4. Спроектирован и реализован интерфейс системы, удовлетворяющий требованиям. Пользователь может выбрать файл формата docx и получить отчет о проверке.

# ЗАКЛЮЧЕНИЕ

# СПИСОК ТЕРМИНОВ И СОКРАЩЕНИЙ

XML – eXtensible Markup Language

XSD – XML Schema definition

ЕСКД – это

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Компания «Философт» [Электронный ресурс]. – https://philosoft-services.com/
2. Компания «Техническая документация» [Электронный ресурс]. – https://tdocs.su/
3. Компания «ABBYY» [Электронный ресурс]. – https://www.abbyy.com/ru-ru/
4. Компания «ПроТекст» [Электронный ресурс]. – https://protext.su/pro/
5. Программное обеспечение Dr.Explain [Электронный ресурс]. – https://www.drexplain.ru/
6. Компания «SWRIT» Разработка технического задания: написание и оформление [Электронный ресурс]. – https://www.swrit.ru/tekhnicheskoe-zadanie.html
7. Центр сертификации «РОСЭКСПЕРТ» [Электронный ресурс]. – https://rosexperts.ru/uslugi/
8. Как происходит распознавание текстов [Электронный ресурс]. – https://www.administrator-pro.ru/articles/kak\_proishodit\_raspoznavanie\_tekstov.html
9. Сервисы для управления задачами и проектами [Электронный ресурс]. – https://vc.ru/services/50333-40-servisov-dlya-upravleniya-zadachami-i-proektami
10. Техническая документация на ремонт автомобилей [Электронный ресурс]. – https://ustroistvo-avtomobilya.ru/to-i-tr/tehnicheskaya-dokumentatsiya-na-remont-avtomobilej/
11. Веб -сервис «Manula» [Электронный ресурс]. – <https://www.manula.com/>
12. Программное обеспечение «АвтоДок» [Электронный ресурс]. – <https://www.auto-doc.ru/>
13. Программное обеспечение «Blitz Document» [Электронный ресурс]. – http://blitz-doc.com/ru/
14. Техническая документация информационных систем: учебное пособие / В. Е. Шикина. – Ульяновск: УлГТУ, 2018. – 92 с.
15. ГОСТ 2.002-72 Единая система конструкторской документации (ЕСКД). Требования к моделям, макетам и темплетам, применяемым при проектировании. – Дата введения 1973-07-01. – Постановлением Государственного комитета стандартов Совета Министров СССР от 30 марта 1972 г. N 655. – 5с.
16. ГОСТ 3.1105-2011 Единая система технологической документации. Формы и правила оформления документов общего назначения. – Дата введения 2012-01-01. – Межгосударственным советом по стандартизации, метрологии и сертификации от 12 мая 2011 г. N 39. – 26с.
17. ГОСТ 19.101-77 Единая система программной документации (ЕСПД). Виды программ и программных документов. – Дата введения 1980-01-01. – Постановлением Государственного комитета стандартов Совета Министров СССР от 20 мая 1977 г. N 1268. – 5с.
18. О техническом регулировании: Федеральный закон [от 27 дек. 2002 г. № 184-ФЗ]. – Принят Государственной Думой 15 декабря 2002 года.
19. XML schema [Электронный ресурс]. – https://www.w3.org/TR/xmlschema-0/
20. XML [Электронный ресурс]. – https://www.w3.org/TR/xml11/
21. Библиотека автоматического реферирования текста на русском языке и выделения ключевых слов (Касаткина А.) [Электронный ресурс]. – https://yadi.sk/d/mEw\_5zg5dl4jhA

# ПРИЛОЖЕНИЕ А ПРИМЕР XSD-ШАБЛОНОВ ДОКУМЕНТОВ

Листинг А.1 – Пример xsd-шаблона ВКРБ

*<?*xml version="1.0" encoding="UTF-8" *?>*<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"  
 xmlns:mc="http://textanalysis.ru"  
 elementFormDefault="qualified">  
  
 <xs:simpleType name="levelType">  
 <xs:restriction base="xs:int">  
 <xs:minExclusive value="-1"/>  
 <xs:maxExclusive value="5"/>  
 </xs:restriction>  
 </xs:simpleType>  
  
 <xs:element name="section">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="title" type="xs:string"/>  
 <xs:element name="content" type="xs:string"

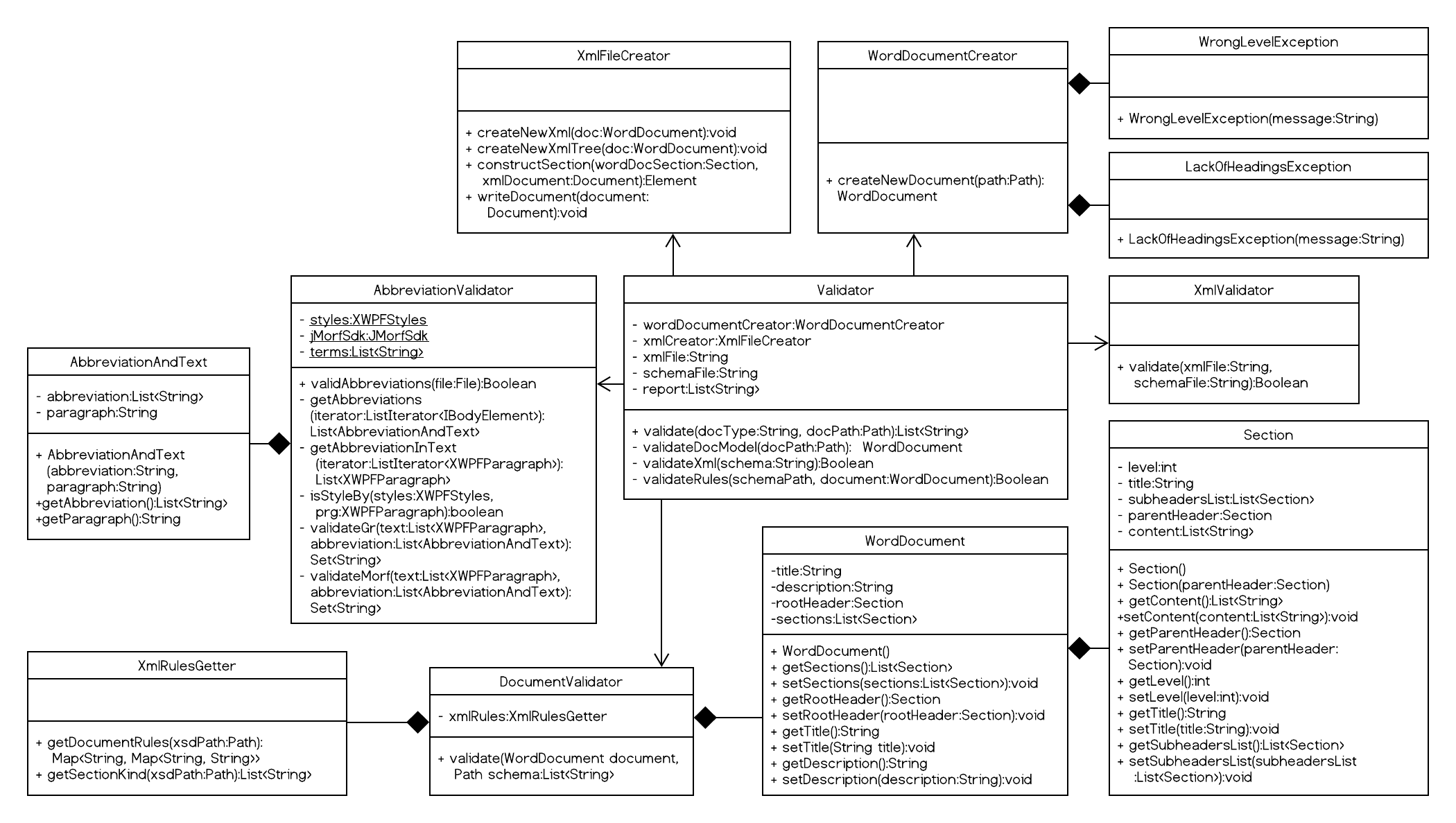
minOccurs="0"/>  
 <xs:element ref="section" minOccurs="0" maxOccurs="10"/>  
 </xs:sequence>  
 <xs:attribute name="level" type="levelType" use="required"/>  
 </xs:complexType>  
 </xs:element>  
  
 <xs:element name="document" mc:kind="document">  
 <xs:annotation>  
 <xs:appinfo>  
 document rules: checkAbbr=true;  
 intro rules: title=ВВЕДЕНИЕ, level=1,

sectionQuantity=0;  
 conclusion rules: title=ЗАКЛЮЧЕНИЕ, level=1,

sectionQuantity=0;  
 literature rules: title=СПИСОК ИСПОЛЬЗОВАННЫХ

ИСТОЧНИКОВ, level=1, sectionQuantity=0  
 </xs:appinfo>  
 </xs:annotation>  
  
 <xs:complexType>  
 <xs:sequence>  
<xs:element ref="section" mc:kind="intro"/><xs:element ref="section" mc:kind="chapter" minOccurs="3" maxOccurs="3"/>  
<xs:element ref="section" mc:kind="conclusion"/>  
  
<xs:element ref="section" mc:kind="literature"/>  
  
 </xs:sequence>  
 <xs:attribute name="level" type="levelType"/>  
 </xs:complexType>  
  
 </xs:element>  
</xs:schema>

# ПРИЛОЖЕНИЕ В ДИАГРАММА КЛАССОВ СИСТЕМЫ



# ПРИЛОЖЕНИЕ Б СПИСОК ПРАВИЛ ДЛЯ РАЗДЕЛОВ

Таблица Б.1 – Правила, которые могут применяться к разделам

|  |  |  |
| --- | --- | --- |
| Правило | Описание | Значения |
| title | Точное значение заголовка | Текст |
| titleContains | Заголовок содержит подстроку | Текст |
| lastSubsectionTitle | Заголовок последнего подраздела содержит подстроку | Текст |
| keySensitive | Чувствительность к регистру | true/false |
| subsectionQuantity | Количество подразделов | Натуральные числа с включенным нулем |
| minSubsectionQuantity | Минимальное количество подразделов | Натуральные числа с включенным нулем |
| maxSubsectionQuantity | Максимальное количество подразделов | Натуральные числа |

# ПРИЛОЖЕНИЕ Г ПРИМЕР СГЕНЕРИРОВАННОЙ XML

Листинг Г.1 – Пример сгенерированной XML

<?xml version="1.0" encoding="utf-8" standalone="no"?>

<document xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" level="0">

<section level="1">

<title>ВВЕДЕНИЕ</title>

<content>Текст раздела</content>

</section>

<section level="1">

<title>1 ГЛАВА</title>

<content>Текст раздела</content>

<section level="2">

<title>1.1 раздел 2 уровня</title>

</section>

<section level="2">

<title>1.2 раздел 2 уровня</title>

<section level="3">

<title>1.2.1 раздел 3 уровня</title>

<content>Текст раздела</content>

</section>

<section level="3">

<title>1.2.2 раздел 3 уровня</title>

<content>Текст раздела</content>

</section>

</section>

<section level="2">

<title>1.3 раздел 2 уровня</title>

<content>Текст раздела</content>

</section>

<section level="2">

<title>1.4 Выводы</title>

<content>Текст раздела</content>

</section>

</section>

<section level="1">

<title>ЗАКЛЮЧЕНИЕ</title>

<content>Текст раздела</content>

</section>

<section level="1">

<title>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ</title>

<content>Текст раздела</content>

</section>

</document>

# ПРИЛОЖЕНИЕ Д ШАБЛОН СХЕМЫ

Листинг Д.1 – Заготовка для создания xsd-шаблона

<?xml version="1.0" encoding="UTF-8" ?>  
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"  
 xmlns:mc="http://textanalysis.ru"  
 elementFormDefault="qualified">  
  
 <xs:simpleType name="levelType">  
 <xs:restriction base="xs:int">  
 <xs:minExclusive value="-1"/>  
 <xs:maxExclusive value="5"/>  
 </xs:restriction>  
 </xs:simpleType>  
  
 <xs:element name="section">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="title" type="xs:string"/>  
 <xs:element name="content" type="xs:string"/>  
 <xs:element ref="section"/>  
 </xs:sequence>  
 <xs:attribute name="level" type="levelType" use="required"/>  
 </xs:complexType>  
 </xs:element>  
  
 <xs:element name="document" mc:kind="document">  
 <xs:annotation>  
 <xs:appinfo>  
 ПРАВИЛА

</xs:appinfo>  
 </xs:annotation>  
  
 <xs:complexType>  
 <xs:sequence>  
  
 <xs:element ref="section" mc:kind="ЗНАЧЕНИЕ 1"/>  
 <xs:element ref="section" mc:kind="ЗНАЧЕНИЕ 2"/>  
 <xs:element ref="section" mc:kind="ЗНАЧЕНИЕ 3"/>  
  
 </xs:sequence>  
 <xs:attribute name="level" type="levelType"/>  
 </xs:complexType>  
  
 </xs:element>  
</xs:schema>