

МИНОБРНАУКИ РОССИИ



Федеральное государственное автономное образовательное учреждение  
высшего образования  
«**Российский государственный гуманитарный университет**»  
(ФГАОУ ВО «РГГУ»)

ИНСТИТУТ ИНФОРМАЦИОННЫХ НАУК И ТЕХНОЛОГИЙ БЕЗОПАСНОСТИ  
ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ СИСТЕМ И БЕЗОПАСНОСТИ  
Кафедра информационных технологий и систем

**ЛИНГВИСТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ**

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

09.03.03 Прикладная информатика

---

*Код и наименование направления подготовки/специальности*

Информационно-коммуникационные технологии цифровой трансформации

---

*Наименование направленности (профиля)*

Уровень высшего образования: *бакалавриат*

Форма обучения: *очная*

РПД адаптирована для лиц  
с ограниченными возможностями  
здоровья и инвалидов

Москва 2026

*Лингвистическое обеспечение информационных систем*  
Рабочая программа дисциплины

Составитель(и):

*старший преподаватель кафедры информационных технологий и систем Е.П. Охупкина*

УТВЕРЖДЕНО

Протокол заседания кафедры ИТС

№5 от 11.12.2025

## ОГЛАВЛЕНИЕ

1.	<u>Пояснительная записка</u> .....	4
1.1.	<u>Цель и задачи дисциплины</u> .....	4
1.2.	<u>Перечень планируемых результатов обучения по дисциплине, соотнесенных с индикаторами достижения компетенций</u> .....	5
1.3.	<u>Место дисциплины в структуре образовательной программы</u> .....	6
2.	<u>Структура дисциплины</u> .....	6
3.	<u>Содержание дисциплины</u> .....	6
4.	<u>Образовательные технологии</u> .....	8
5.	<u>Оценка планируемых результатов обучения</u> .....	8
5.1.	<u>Система оценивания</u> .....	8
5.2.	<u>Критерии выставления оценки по дисциплине</u> .....	8
5.3.	<u>Оценочные средства (материалы) для текущего контроля успеваемости, промежуточной аттестации обучающихся по дисциплине</u> .....	9
6.	<u>Учебно-методическое и информационное обеспечение дисциплины</u> .....	11
6.1.	<u>Список источников и литературы</u> .....	11
6.2.	<u>Перечень ресурсов информационно-телекоммуникационной сети «Интернет»</u> .....	12
6.3.	<u>Профессиональные базы данных и информационно-справочные системы</u> .....	12
7.	<u>Материально-техническое обеспечение дисциплины</u> .....	12
8.	<u>Обеспечение образовательного процесса для лиц с ограниченными возможностями здоровья и инвалидов</u> .....	13
9.	<u>Методические материалы</u> .....	14
9.1.	<u>Планы практических занятий</u> .....	14
9.2.	<u>Методические рекомендации по подготовке письменных работ</u> .....	27
	<u>Приложение 1. Аннотация рабочей программы дисциплины</u> .....	29

## 1. Пояснительная записка

### 1.1. Цель и задачи дисциплины

#### *Цель дисциплины:*

профессиональная подготовка студентов, необходимая для освоения методов и подходов применения информационных технологий, программных систем и технических средств для классификации, кодирования и решения задач автоматизированной обработки лингвистических единиц, описывающих объекты и субъекты предметной области информационных систем. Задачи дисциплины: получение систематизированных знаний о современных компьютерных технологиях, используемых в лингвистическом обеспечении информационных систем; изучение методов защиты информации в автоматизированных системах классификации и кодирования элементов обрабатываемых лингвистическим обеспечением информационных систем; приобретение знаний о структуре и функциях автоматизированных систем лингвистического обеспечения и роли этих систем в процессе создания, эксплуатации и дистрибуции информационных систем; сформировать умения использования нормативно-правовых документов для оценки лингвистического обеспечения информационных систем; владение навыками работы с лингвистическим обеспечением элементов и компонентов электронного документооборота; сформировать умения самостоятельного принятия решения о внедрении тех или иных технологий лингвистического обеспечения для повышения эффективности информационных систем. развитие адекватного системного мышления, умение однозначно излагать свои мысли, развитие способностей к обобщению и анализу элементов лингвистического обеспечения, постановке задач классификации и кодирования объектов обработки информационных систем и выбору путей их достижения.

#### *Задачи дисциплины:*

- получение систематизированных знаний о современных компьютерных технологиях, используемых в лингвистическом обеспечении информационных систем;
- изучение методов защиты информации в автоматизированных системах классификации и кодирования элементов, обрабатываемых лингвистическим обеспечением информационных систем;
- приобретение знаний о структуре и функциях автоматизированных систем лингвистического обеспечения и роли этих систем в процессе создания, эксплуатации и дистрибуции информационных систем;
- сформировать умения использования нормативно-правовых документов для оценки лингвистического обеспечения информационных систем;
- владение навыками работы с лингвистическим обеспечением элементов и компонентов электронного документооборота;
- сформировать умения самостоятельного принятия решения о внедрении тех или иных технологий лингвистического обеспечения для повышения эффективности информационных систем.
- развитие адекватного системного мышления, умение однозначно излагать свои мысли, развитие способностей к обобщению и анализу элементов лингвистического обеспечения, постановке задач классификации и кодирования объектов обработки информационных систем и выбору путей их достижения.

1.

### 1.2. Перечень планируемых результатов обучения по дисциплине, соотнесенных с индикаторами достижения компетенций

Компетенция	Индикаторы	Результаты обучения
-------------	------------	---------------------

	компетенций	
ПК-3 Способен проектировать информационные системы по видам обеспечения	ПК-3.1 Знает модели жизненного цикла информационных систем, основные технологии, стадии и этапы их проектирования	<p><b>Знать:</b> основную терминологию, связанную с использованием компьютерных технологий в лингвистическом обеспечении информационных систем; понятие кодирования электронного документа, его составных элементов и стадий жизненного цикла.</p> <p><b>Уметь:</b> строить структурные модели информационного обеспечения управления; разрабатывать модели информационных и документационных потоков.</p> <p><b>Владеть:</b> методами и программными средствами лингвистического обеспечения информационных систем.</p>
	ПК-3.2 Умеет применять технологии проектирования информационных систем по видам обеспечения	<p><b>Знать:</b> цели и задачи систем информационного обеспечения; методы, технологии и средства автоматизации работ по созданию и эксплуатации лингвистического обеспечения; методы моделирования и анализа предметных областей, охватываемых лингвистическим обеспечением информационных систем; специфику документационных процессов в условиях «электронного офиса».</p> <p><b>Уметь:</b> осуществлять контроллинг и мониторинг за реализацией процессов и документопотоков в соответствии с требованиями модели; проводить анализ эффективности организации системы ДОУ.</p> <p><b>Владеть:</b> навыками применения методов анализа моделей данных, документопотоков и информационных потоков в информационных системах с целью их оптимизации и реорганизации.</p>
	ПК-3.3 Владеет навыками проектирования информационных систем или их частей по видам обеспечения	<p><b>Знать:</b> классификацию функций информационных систем; стандарты и нормативно-правовое обеспечение в области лингвистическим обеспечением информационных систем, элементы математической лингвистики и теории формальных языков.</p> <p><b>Уметь:</b> выбирать информационные технологии и программные системы для решения задач документационного обеспечения управления; разрабатывать требования к ИС в части ЛО; разрабатывать концептуальную модель прикладной области, проводить оценку эффективности внедрения и применения систем управления электронными документами предприятия.</p> <p><b>Владеть:</b> навыками работы с системами</p>

		лингвистического обеспечения различного типа; организации совместной работы с адекватными средствами лингвистического обеспечения в среде корпоративной информационной системы.
--	--	---

### 1.3. Место дисциплины в структуре образовательной программы

Дисциплина «Лингвистическое обеспечение информационных систем» относится к части, формируемой участниками образовательных отношений блока дисциплин учебного плана.

Для освоения дисциплины необходимы знания, умения и владения, сформированные в ходе изучения дисциплин Программная инженерия, Дискретная математика и математическая логика, Теория вероятностей и математическая статистика, Архитектура вычислительных систем, Базы данных.

В результате освоения дисциплины формируются знания, умения и владения, необходимые для изучения следующих дисциплин и прохождения практик: Моделирование бизнес-процессами и проектирование систем, Управление информационными системами, Стандартизация в области информационных технологий, Управление проектами цифровой трансформации.

### 2. Структура дисциплины

Общая трудоёмкость дисциплины составляет 3 з.е., 108 академических часов.

#### Структура дисциплины для очной формы обучения

Объем дисциплины в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении учебных занятий:

Семестр	Тип учебных занятий	Количество часов
5	Лекции	14
5	Практические занятия	28
Всего:		42

Объем дисциплины (модуля) в форме самостоятельной работы обучающихся составляет 66 академических часов.

### 3. Содержание дисциплины

№	Наименование раздела дисциплины	Содержание
1.	Раздел 1. Теоретические основы и общие понятия лингвистического обеспечения. Исторический экскурс	Общая модель коммуникативного взаимодействия. Особенности процесса моделирования применительно к типу коммуникации «пользователь – АИС. Целевая модель коммуникации. Компоненты и предпосылки коммуникативного взаимодействия. Лингвистические и психологические аспекты коммуникации. Компоненты, структура, уровни языка. Лингвистика текста. Целевая теория диалога. Основные понятия семиотики и логики. Синтактика, семантика, прагматика. Треугольник Фреге. Логика классов, высказываний, предикатов. Логические операции. Теория

		<p>речевых актов. Иллокутивный и перлокутивный речевые акты. Пресуппозиция. Семантические и прагматические пресуппозиции.</p> <p>Определение понятия лингвистического обеспечения. История разработки ЛО в России. Классификация средств ЛО</p>
2.	Раздел 2. Электронные библиотеки. Системы метаданных	<p>Классификация информационных систем по функциональному назначению, по используемой технологии, по типу объектов управления. Логическая сложность ЭБ. Электронные библиотеки как специфический вид АИС. Основные функциональные возможности ЭБ. Основы теории информационного поиска. Цели и методы поиска Виды поиска и основные компоненты АИПС. Информационные массивы. Индексирование и индексы. Язык запросов. Механизмы отбора и технологии поиска.</p> <p>Семантические языки разметки текста. Стандартный обобщенный язык разметки SGML. Гипертекстовый язык разметки HTML. Расширяемый язык разметки XML. Диалекты XML.</p> <p>Общие понятия и основные системы метаданных. Языки библиографических данных. Дублинское ядро метаданных</p> <p>МАРК - машиночитаемый каталог. Формат ONIX</p> <p>Организация деятельности по созданию метаданных</p>
3.	Раздел 3. Классификационные и вербальные языки. Лингвистическое обеспечение фактографических и комплексных АИС	<p>Общие понятия классификации. Универсальная десятичная классификация (УДК). Государственный рубрикатор научно-технической информации (ГРНТИ). Проблемы и перспективы применения информационных классификаций в ЭБ</p> <p>Общее описание и история развития вербальных языков. Лексика и организация лексики в вербальных языках.</p> <p>Информационно-поисковый тезаурус</p> <p>Принципы создания и ведения тезауруса УИС «Россия»</p> <p>Грамматика вербальных ИПЯ традиционных АИПС. Методика индексирования средствами вербальных языков в традиционных АИПС. Грамматики вербальных языков современных электронных библиотеках.</p> <p>Организация поиска с использованием вербальных ИПЯ</p>
4.	Раздел 4. Автоматическая обработка текста. Лингвистические банки данных и компьютерная лексикография	<p>Общие понятия фактографии. Интегрированные документально-фактографические АИС. Комбинированные документально-фактографические системы.</p> <p>Обработка запросов на естественном языке к фактографическим базам данных. ЛО ситуационного управления</p>
5.	Раздел 5. Моделирование и оптимизация лингвистического обеспечения ИС	<p>Виды процессов автоматической обработки текста. Морфологический анализ текста. Синтаксический анализ. Позиционные методы анализа текста. Суперсинтаксический анализ. Семантический анализ.</p> <p>Статистические методы. Требования к автоматическому индексированию.</p> <p>Основные типы словарей в АИС. Примеры организации лингвистических банков данных в АИС. Обмен словарями и</p>

	коммуникативные форматы словарей Лингвистические банки данных в Интернете Основы компьютерной лексикографии
--	---

#### 4. Образовательные технологии

Для проведения учебных занятий по дисциплине используются различные образовательные технологии. Для организации учебного процесса может быть использовано электронное обучение и (или) дистанционные образовательные технологии.

#### 5. Оценка планируемых результатов обучения

##### 5.1 Система оценивания

Форма контроля	Макс. количество баллов	
	За одну работу	Всего
Текущий контроль:		
Практическая работа № 1, защита отчета	12 баллов	12 баллов
Практическая работа № 2, защита отчета	12 баллов	12 баллов
Практическая работа № 3, защита отчета	12 баллов	12 баллов
Практическая работа № 4, защита отчета	12 баллов	12 баллов
Практическая работа № 5, защита отчета	12 баллов	12 баллов
Промежуточная аттестация <i>зачет</i>		40 баллов
<b>Итого за семестр</b>		<b>100 баллов</b>

Полученный совокупный результат конвертируется в традиционную шкалу оценок и в шкалу оценок Европейской системы переноса и накопления кредитов (European Credit Transfer System; далее – ECTS) в соответствии с таблицей:

100-балльная шкала	Традиционная шкала	Шкала ECTS	
95 – 100	отлично	A	
83 – 94		B	
68 – 82	хорошо	зачтено	
56 – 67	удовлетворительно		D
50 – 55			E
20 – 49	неудовлетворительно	FX	
0 – 19		не зачтено	F

##### 5.2 Критерии выставления оценки по дисциплине

Баллы/ Шкала ECTS	Оценка по дисциплине	Критерии оценки результатов обучения по дисциплине
100-83/ A,B	отлично/ зачтено	Выставляется обучающемуся, если он глубоко и прочно усвоил теоретический и практический материал, может продемонстрировать это на занятиях и в ходе промежуточной аттестации. Обучающийся исчерпывающе и логически стройно излагает учебный материал, умеет увязывать теорию с практикой, справляется с решением задач профессиональной направленности высокого уровня сложности, правильно

Баллы/ Шкала ECTS	Оценка по дисциплине	Критерии оценки результатов обучения по дисциплине
		<p>обосновывает принятые решения. Свободно ориентируется в учебной и профессиональной литературе.</p> <p>Оценка по дисциплине выставляются обучающемуся с учётом результатов текущей и промежуточной аттестации. Компетенции, закреплённые за дисциплиной, сформированы на уровне – «высокий».</p>
82-68/ С	хорошо/ зачтено	<p>Выставляется обучающемуся, если он знает теоретический и практический материал, грамотно и по существу излагает его на занятиях и в ходе промежуточной аттестации, не допуская существенных неточностей. Обучающийся правильно применяет теоретические положения при решении практических задач профессиональной направленности разного уровня сложности, владеет необходимыми для этого навыками и приёмами. Достаточно хорошо ориентируется в учебной и профессиональной литературе. Оценка по дисциплине выставляются обучающемуся с учётом результатов текущей и промежуточной аттестации. Компетенции, закреплённые за дисциплиной, сформированы на уровне – «хороший».</p>
67-50/ D,E	удовлетво- рительно/ зачтено	<p>Выставляется обучающемуся, если он знает на базовом уровне теоретический и практический материал, допускает отдельные ошибки при его изложении на занятиях и в ходе промежуточной аттестации. Обучающийся испытывает определённые затруднения в применении теоретических положений при решении практических задач профессиональной направленности стандартного уровня сложности, владеет необходимыми для этого базовыми навыками и приёмами. Демонстрирует достаточный уровень знания учебной литературы по дисциплине. Оценка по дисциплине выставляются обучающемуся с учётом результатов текущей и промежуточной аттестации. Компетенции, закреплённые за дисциплиной, сформированы на уровне – «достаточный».</p>
49-0/ F,FX	неудовлет- ворительно/ не зачтено	<p>Выставляется обучающемуся, если он не знает на базовом уровне теоретический и практический материал, допускает грубые ошибки при его изложении на занятиях и в ходе промежуточной аттестации. Обучающийся испытывает серьёзные затруднения в применении теоретических положений при решении практических задач профессиональной направленности стандартного уровня сложности, не владеет необходимыми для этого навыками и приёмами. Демонстрирует фрагментарные знания учебной литературы по дисциплине. Оценка по дисциплине выставляются обучающемуся с учётом результатов текущей и промежуточной аттестации. Компетенции на уровне «достаточный», закреплённые за дисциплиной, не сформированы.</p>

### 5.3 Оценочные средства (материалы) для текущего контроля успеваемости, промежуточной аттестации обучающихся по дисциплине

#### Контрольные вопросы зачета (ПК-3)

1. Теоретические основы ЛО. Общая модель коммуникативного взаимодействия. Лингвистические и психологические аспекты коммуникации
2. Теоретические основы ЛО. Основные понятия семиотики и логики,
3. Теоретические основы ЛО Основные понятия теории речевых актов. Пресуппозиция.
4. Классификация информационных систем. Электронные библиотеки как специфический вид АИС
5. Основы теории информационного поиска
6. Семантические языки разметки текста.
7. Различные подходы к определению ЛО. История разработки ЛО в России

8. Классификация средств ЛО.
9. Общие понятия и основные системы метаданных.
10. Языки библиографических данных. Дублинское ядро метаданных.
11. Форматы MARC и ONIX. Организация деятельности по созданию метаданных
12. Классификационные языки. Общие понятия классификации.
13. Классификационные языки. УДК.
14. Классификационные языки. ГРНТИ.
15. Проблемы и перспективы применения информационных классификаций в ЭБ
16. Вербальные языки. Общее описание и история развития вербальных языков
17. Лексика и организация лексики в вербальных языках.
18. Информационно-поисковый тезаурус. Принципы создания и ведения тезауруса в УИС «Россия».
19. Грамматика вербальных ИПЯ традиционных АИПС. Методика индексирования.
20. Грамматики вербальных языков современных ЭБ. Организация поиска с использованием вербальных ИПЯ
21. ЛО фактографических и комплексных АИС. Общие понятия фактографии.
22. Интегрированные и комбинированные документально-фактографические системы.
23. Автоматическая обработка текста (АОТ). Виды процессов автоматической обработки текста. Морфологический анализ текста.
24. АОТ. Синтаксический анализ.
25. АОТ. Позиционные методы анализа текста. Суперсинтаксический анализ.
26. АОТ. Семантический анализ. Статистические методы.
27. АОТ. Требования к автоматическому индексированию.
28. Лингвистические банки данных и компьютерная лексикография. Основные типы словарей в АИС. Примеры организации лингвистических банков данных в АИС.
29. Обмен словарями и коммуникативные форматы словарей
30. Лингвистические банки данных в Интернете (+ самостоятельный анализ)
31. Основы компьютерной лексикографии

#### **Тематика рефератов (докладов)**

1. Теоретические основы ЛО. Общая модель коммуникативного взаимодействия. Лингвистические и психологические аспекты коммуникации
2. Теоретические основы ЛО. Основные понятия семиотики и логики,
3. Теоретические основы ЛО Основные понятия теории речевых актов. Пресуппозиция.
4. Классификация информационных систем. Электронные библиотеки как специфический вид АИС
5. Основы теории информационного поиска
6. Семантические языки разметки текста.
7. Различные подходы к определению ЛО. История разработки ЛО в России
8. Классификация средств ЛО.
9. Общие понятия и основные системы метаданных.
10. Языки библиографических данных. Дублинское ядро метаданных.
11. Форматы MARC и ONIX. Организация деятельности по созданию метаданных
12. Классификационные языки. Общие понятия классификации.
13. Классификационные языки. УДК.
14. Классификационные языки. ГРНТИ.
15. Проблемы и перспективы применения информационных классификаций в ЭБ
16. Вербальные языки. Общее описание и история развития вербальных языков
17. Лексика и организация лексики в вербальных языках.
18. Информационно-поисковый тезаурус. Принципы создания и ведения тезауруса в УИС «Россия».

19. Грамматика вербальных ИПЯ традиционных АИПС. Методика индексирования.
20. Грамматика вербальных языков современных ЭБ. Организация поиска с использованием вербальных ИПЯ
21. ЛО фактографических и комплексных АИС. Общие понятия фактографии.
22. Интегрированные и комбинированные документально-фактографические системы.
23. Автоматическая обработка текста (АОТ). Виды процессов автоматической обработки текста. Морфологический анализ текста.
24. АОТ. Синтаксический анализ.
25. АОТ. Позиционные методы анализа текста. Суперсинтаксический анализ.
26. АОТ. Семантический анализ. Статистические методы.
27. АОТ. Требования к автоматическому индексированию.
28. Лингвистические банки данных и компьютерная лексикография. Основные типы словарей в АИС. Примеры организации лингвистических банков данных в АИС.
29. Обмен словарями и коммуникативные форматы словарей
30. Лингвистические банки данных в Интернете (+ самостоятельный анализ)
31. Основы компьютерной лексикографии

## **6. Учебно-методическое и информационное обеспечение дисциплины**

### **6.1 Список источников и литературы**

#### Источники Основные

1. Федеральный закон Российской Федерации от 27 июля 2006 г. N 149-ФЗ «Об информации, информационных технологиях и о защите информации».
2. ГОСТ 34.003-90. Автоматизированные системы. Термины и определения.
3. ГОСТ 34.201-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Виды, комплектность и обозначение документов при создании автоматизированных систем.
4. ГОСТ 34.601-90. Автоматизированные системы. Комплекс стандартов на автоматизированные системы. Стадии создания.

#### Литература

##### **Основная**

1. Чернышев, С. А. Принципы, паттерны и методологии разработки программного обеспечения : учебник для среднего профессионального образования / С. А. Чернышев. — Москва : Издательство Юрайт, 2026. — 176 с. — (Профессиональное образование). — ISBN 978-5-534-18705-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/589664>
2. Толстобров, А. П. Управление данными : учебник для вузов / А. П. Толстобров. — 3-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2026. — 272 с. — (Высшее образование). — ISBN 978-5-534-14162-7. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/588522>
3. Казарин, Ю. В. Лингвистический анализ текста : учебник для вузов / Ю. В. Казарин ; под научной редакцией Л. Г. Бабенко. — 2-е изд. — Москва : Издательство Юрайт, 2025. — 119 с. — (Высшее образование). — ISBN 978-5-534-17902-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/564037>
4. Тарланов, З. К. Методы лингвистического анализа : для вузов / З. К. Тарланов. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2026. — 236 с. — (Высшее образование)

образование). — ISBN 978-5-534-07101-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/586616>

### Дополнительная

1. Проектирование информационных систем: учебник и практикум для вузов / Д. В. Чистов, П. П. Мельников, А. В. Золотарюк, Н. Б. Ничепорук. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2024. — 273 с. — (Высшее образование). — ISBN 978-5-534-20361-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/558007>
2. Лаврищева, Е. М. Программная инженерия и технологии программирования сложных систем: учебник для вузов / Е. М. Лаврищева. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2023. — 432 с. — (Высшее образование). — ISBN 978-5-534-07604-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/513067>
3. Тузовский, А. Ф. Объектно-ориентированное программирование: учебное пособие для вузов / А. Ф. Тузовский. — Москва : Издательство Юрайт, 2024. — 213 с. — (Высшее образование). — ISBN 978-5-534-16316-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/537332>

### 6.2 Перечень ресурсов информационно-телекоммуникационной сети «Интернет».

Национальная электронная библиотека (НЭБ) [www.rusneb.ru](http://www.rusneb.ru)

ELibrary.ru Научная электронная библиотека [www.elibrary.ru](http://www.elibrary.ru)

Электронная библиотека Grebennikon.ru [www.grebennikon.ru](http://www.grebennikon.ru)

Cambridge University Press

### 6.3 Профессиональные базы данных и информационно-справочные системы

Доступ к профессиональным базам данных: <https://liber.rsuh.ru/ru/bases>

Информационные справочные системы:

1. Консультант Плюс
2. Гарант

### 7. Материально-техническое обеспечение дисциплины

№п/п	Наименование специальных помещений и помещений для самостоятельной работы	Оснащенность специальных помещений и помещений для самостоятельной работы	Перечень лицензионного программного обеспечения. Реквизиты подтверждающего документа		
			Наименование ПО	Лицензия/сертификат/заказ	Дата лицензии
1.	Лаборатория информатики – ауд. № 203	1 компьютер преподавателя, 12 компьютеров обучающихся, маркерная доска, проектор	Windows 7 Microsoft office 2010 Pro Microsoft Visual Professional 2019 Mozilla Firefox 52.8.1 ESR  Matlab Mathcad Education - University edition	68526624 49420326 63202190 свободный доступ 647526 2996385  17E0181226094912873979	без даты 08.12.2011 без даты свободный доступ без даты 14.06.2019  26.12.2018

			Kaspersky Endpoint Security		
--	--	--	-----------------------------------	--	--

## **8. Обеспечение образовательного процесса для лиц с ограниченными возможностями здоровья и инвалидов**

В ходе реализации дисциплины используются следующие дополнительные методы обучения, текущего контроля успеваемости и промежуточной аттестации обучающихся в зависимости от их индивидуальных особенностей:

- для слепых и слабовидящих: лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением; письменные задания выполняются на компьютере со специализированным программным обеспечением или могут быть заменены устным ответом; обеспечивается индивидуальное равномерное освещение не менее 300 люкс; для выполнения задания при необходимости предоставляется увеличивающее устройство; возможно также использование собственных увеличивающих устройств; письменные задания оформляются увеличенным шрифтом; экзамен и зачёт проводятся в устной форме или выполняются в письменной форме на компьютере.

- для глухих и слабослышащих: лекции оформляются в виде электронного документа, либо предоставляется звукоусиливающая аппаратура индивидуального пользования; письменные задания выполняются на компьютере в письменной форме; экзамен и зачёт проводятся в письменной форме на компьютере; возможно проведение в форме тестирования.

- для лиц с нарушениями опорно-двигательного аппарата: лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением; письменные задания выполняются на компьютере со специализированным программным обеспечением; экзамен и зачёт проводятся в устной форме или выполняются в письменной форме на компьютере.

При необходимости предусматривается увеличение времени для подготовки ответа.

Процедура проведения промежуточной аттестации для обучающихся устанавливается с учётом их индивидуальных психофизических особенностей. Промежуточная аттестация может проводиться в несколько этапов.

При проведении процедуры оценивания результатов обучения предусматривается использование технических средств, необходимых в связи с индивидуальными особенностями обучающихся. Эти средства могут быть предоставлены университетом, или могут использоваться собственные технические средства.

Проведение процедуры оценивания результатов обучения допускается с использованием дистанционных образовательных технологий.

Обеспечивается доступ к информационным и библиографическим ресурсам в сети Интернет для каждого обучающегося в формах, адаптированных к ограничениям их здоровья и восприятия информации:

- для слепых и слабовидящих: в печатной форме увеличенным шрифтом, в форме электронного документа, в форме аудиофайла.

- для глухих и слабослышащих: в печатной форме, в форме электронного документа.

- для обучающихся с нарушениями опорно-двигательного аппарата: в печатной форме, в форме электронного документа, в форме аудиофайла.

Учебные аудитории для всех видов контактной и самостоятельной работы, научная библиотека и иные помещения для обучения оснащены специальным оборудованием и учебными местами с техническими средствами обучения:

- для слепых и слабовидящих: устройством для сканирования и чтения с камерой SARA CE; дисплеем Брайля PAC Mate 20; принтером Брайля EmBraille ViewPlus;

- для глухих и слабослышащих: автоматизированным рабочим местом для людей с нарушением слуха и слабослышащих; акустический усилитель и колонки;
- для обучающихся с нарушениями опорно-двигательного аппарата: передвижными, регулируемые эргономическими партами СИ-1; компьютерной техникой со специальным программным обеспечением.

## 9. Методические материалы

### 9.1 Планы практических занятий

В плане практических занятий выполняются следующие работы;

#### **Практическая работа 1**

Введение в XML. DTD и XML-документы

Изучение DTD и схемы документа на основе словаря SArcicle

#### **Практическая работа 2**

Создание XML-документа, с применением формата SArcicle

#### **Практическая работа 3**

Разработка собственного DTD и схемы документа. Пример документа, описанного в соответствии с созданным DTD (схемой).

#### **Практическая работа 4**

Изучение классификационных языков на примерах конкретных информационных классификаций (УДК, ГРНТИ, МПК)

#### **Практическая работа 5**

Анализ функциональных возможностей и информационных ресурсов русскоязычных ЭБ (УИС «Россия», ЦБОР, ГПНТБ)

## *Практические работы №1 - 3*

### *Основы XML*

Цели практической работы:

- 1) Получить навыки создания XML-документов различных типов;
- 2) Научиться определять собственные типы документов.

Задачи практической работы:

- 1) Получить представление об уже существующих XML-форматах описания данных (на примере формата описания сериальных изданий SArcicle);
- 2) Ознакомиться с основными этапами создания DTD-описаний и схем документов;
- 3) Научиться создавать собственные DTD (или схемы) и документы.

Этапы выполнения:

- 1) Ознакомиться с форматом описания сериальных изданий SArcicle (см. примеры в файлах Sarcticle.dtd, SarcticleSample.xml): изучить DTD, ознакомиться с документом в формате SArcicle.
- 2) Создать документ в формате SArcicle и проверить его соответствие DTD.
- 3) Создать собственное DTD или схему для описания определенного типа ресурсов.
- 4) Создать несколько документов в соответствии с описанной структурой.

Теоретические сведения, необходимые для выполнения работы

Язык XML

**XML (Extensible Markup Language)** - это язык разметки, описывающий целый класс объектов данных, называемых XML- документами. Этот язык используется в качестве средства

для описания грамматики других языков и контроля за правильностью составления документов. Т.е. сам по себе XML не содержит никаких тэгов, предназначенных для разметки, он просто определяет порядок их создания.

Для описания данных XML использует DTD (Document Type Definition - Определение типа документа) или схему документа.

### Синтаксис XML

Пример и структура XML-документа

Простейший XML- документ может выглядеть так:

```
<?xml version="1.0"?>
<list_of_items>
<item id="1"><first>Первый</item>
<item id="2">Второй <sub_item>подпункт 1</sub_item></item>
<item id="3">Третий</item>
<item id="4"></last>Последний</item>
</list_of_items>
```

Этот документ очень похож на обычную HTML-страницу. Также, как и в HTML, инструкции, заключенные в угловые скобки называются тэгами и служат для разметки основного текста документа. В XML существуют открывающие, закрывающие и пустые тэги (в HTML понятие пустого тэга тоже существует, но специального его обозначения не требуется).

Тело документа XML состоит из элементов разметки и непосредственно содержимого документа - данных. XML-тэги предназначены для определения элементов документа, их атрибутов и других конструкций языка.

Любой XML- документ должен всегда начинаться с инструкции `<?xml?>`, внутри которой также можно задавать номер версии языка, номер кодовой страницы и другие параметры, необходимые программе-анализатору в процессе разбора документа.

### Конструкции языка

Содержимое XML- документа представляет собой набор элементов, секций CDATA, директив анализатора, комментариев, спецсимволов, текстовых данных. Рассмотрим каждый из них подробнее.

### Элементы данных

Элемент - это структурная единица XML- документа. Заключая слово `rose` в в тэги, мы определяем непустой элемент, называемый `<flower>`, содержимым которого является `rose`. В общем случае в качестве содержимого элементов могут выступать как просто какой-то текст, так и другие, вложенные, элементы документа, секции CDATA, инструкции по обработке, комментарии, - т.е. практически любые части XML- документа.

Любой непустой элемент должен состоять из начального, конечного тэгов и данных, между ними заключенных: `<flower> rose </flower>`.

### Атрибуты

Если при определении элементов необходимо задать какие-либо параметры, уточняющие его характеристики, то имеется возможность использовать атрибуты элемента. Атрибут - это пара "название" = "значение", которую надо задавать при определении элемента в начальном тэге.

Пример:

```
<color RGB="true">#ff08ff</color>
```

```
<color RGB="false">white</color>
```

### Сущности и специальные символы

Для того, чтобы включить в документ символ, используемый для определения каких-либо конструкций языка (например, символ угловой скобки) и не вызвать при этом ошибок в процессе разбора такого документа, нужно использовать его специальный символьный либо числовой идентификатор. Например, &lt; , &gt; &quot; или &#036;(десятичная форма записи), &#x1a (шестнадцатеричная) и т.д.

Если вы поместите внутрь XML-элемента символ "<", это вызовет ошибку, потому что парсер проинтерпретирует его как начало нового элемента. Поэтому писать вот так нельзя:

```
<message>if salary < 1000 then</message>
```

Чтобы избежать ошибки парсера, нужно заменить символ "<" ссылкой на сущность вот так:

```
<message>if salary &lt; 1000 then</message>
```

В XML есть пять изначально заданных сущностей:

ссылка на сущность	символ
&lt;	< - меньше, чем
&gt;	> - больше, чем
&amp;	& - амперсанд
&apos;	` - апостроф
&quot;	" - двойная кавычка

Ссылки на сущности всегда начинаются с амперсанда "&" и заканчиваются точкой с запятой: ";".

Обратите внимание: только символы "<" и "&" строго неприемлемы в XML. Апострофы, кавычки и символы "больше, чем" приемлемы, но лучше их все же заменять ссылками на сущности.

### Комментарии

Комментариями является любая область данных, заключенная между последовательностями символов

```
<!-- -->
```

Комментарии пропускаются анализатором и поэтому при разборе структуры документа в качестве значащей информации не рассматриваются.

### Директивы анализатора

Инструкции, предназначенные для анализаторов языка, описываются в XML документе при помощи специальных тэгов - <? и ?>;. Программа клиента использует эти инструкции для управления процессом разбора документа. Наиболее часто инструкции используются при определении типа документа (например, <? Xml version="1.0"?>) или создании пространства имен.

### CDATA

Чтобы задать область документа, которую при разборе анализатор будет рассматривать как простой текст, игнорируя любые инструкции и специальные символы, но, в отличие от комментариев, иметь возможность использовать их в приложении, необходимо использовать тэги <![CDATA] и ]>. Внутри этого блока можно помещать любую информацию, которая может понадобиться программе- клиенту для выполнения каких-либо действий (в область

CDATA, можно помещать, например, инструкции JavaScript). Естественно, надо следить за тем, чтобы в области, ограниченной этими тэгами не было последовательности символов ]].

### Правила создания XML- документа

#### 1) В XML все элементы должны иметь закрывающий тэг

В XML не разрешается опускать конечные тэги элементов.

В HTML некоторые элементы могут не иметь закрывающего тэга, например так:

```
<p>Параграф 1
```

В XML все элементы обязательно должны иметь закрывающий тэг:

```
<p>Параграф 1</p>
```

#### 2) В тэгах XML учитывается регистр

В отличие от HTML, в XML-тэгах учитывается регистр

В XML тэг <Book> отличается от тэга <book>.

Таким образом, начальные и конечные тэги должны писаться в одном регистре:

```
<Book>Вот так неправильно</book>
```

```
<book>Вот так правильно</book>
```

#### 3) Элементы XML должны быть правильно вложены друг в друга

Неверное вложение элементов считается в XML неправильным

В HTML некоторые элементы могут быть вложены друг в друга неправильно, например:

```
<b><i>Этот текст пишется полужирным курсивом</b></i>
```

В XML все элементы должны быть правильно вложены друг в друга, например:

```
<b><i>Этот текст пишется полужирным курсивом</i></b>
```

#### 4) XML-документы должны иметь единственный корневой элемент

Первый тэг XML-документа является корневым тэгом

Все XML-документы должны иметь единственную пару тэгов, задающую корневой элемент. Все остальные элементы должны быть вложены в корневой элемент.

Все элементы могут иметь под-элементы (дочерние элементы). Под-элементы должны быть правильно вложены внутри родительского элемента.

```
<root>
```

```
<child>
```

```
<subchild>.....</subchild>
```

```
</child>
```

```
</root>
```

#### 5) Значения атрибутов всегда должны быть заключены в кавычки

В XML не разрешается опускать кавычки вокруг значения атрибутов

В XML элементы могут обладать атрибутами в парах имя атрибута/его значение, также, как HTML. Но в XML значения атрибутов всегда должны быть заключены в кавычки.

```
<?xml version="1.0"?>
```

```
<message date="21/02/02">
```

```
<theme>Тема сообщения</theme>
```

```
<body>Тело сообщения</body>
```

```
</message>
```

Атрибут `date` в элементе `message` должен быть заключен в кавычки.

## 6) В XML пробелы сохраняются

В XML пробелы в вашем документе не будут устраняться. Этим XML отличается от HTML.

Если XML-документ не нарушает приведенные правила, то он называется формально-правильным и все анализаторы, предназначенные для разбора XML- документов, смогут работать с ним корректно.

Однако кроме проверки на формальное соответствие грамматике языка, в документе могут присутствовать средства контроля над содержанием документа, за соблюдением правил, определяющих необходимые соотношений между элементами и формирующих структуру документа.

Для того, чтобы обеспечить проверку корректности XML- документов, необходимо использовать анализаторы, производящие такую проверку и называемые верифицирующими.

На сегодняшний день существует два способа контроля правильности XML- документа: DTD-определения(Document Type Definition) и схемы данных (Semantic Schema). Более подробно об использовании DTD и схемах мы поговорим в следующих разделах. В отличие от SGML, определение DTD- правил в XML не является необходимостью, и это обстоятельство позволяет нам создавать любые XML- документы, не ломая пока голову над весьма непростым синтаксисом DTD.

### DTD – описания

В XML-документах DTD определяет набор действительных элементов, идентифицирует элементы, которые могут находиться в других элементах, и определяет действительные атрибуты для каждого из них.

Для того, чтобы использовать DTD в нашем документе, мы можем или описать его во внешнем файле и при описании DTD просто указать ссылку на этот файл или же непосредственно внутри самого документа выделить область, в которой определить нужные правила. В первом случае в документе указывается имя файла, содержащего DTD-описания:

```
<?xml version="1.0" standalone="yes" ?>
<!DOCTYPE journal SYSTEM "journal.dtd">
```

...

Внутри же документа DTD- декларации включаются следующим образом:

...

```
<!DOCTYPE journal [
<!ELEMENT journal (contacts, issues, authors)>
```

...

```
]>
```

...

В том случае, если используются одновременно внутренние и внешние описания, то программой-анализатором будут сначала рассматриваться внутренние, т.е. их приоритет выше. При проверке документа XML- процессор в первую очередь ищет DTD внутри документа. Если правила внутри документа не определены и не задан атрибут `standalone = "yes"` , то программа загрузит указанный внешний файл и правила, находящиеся в нем, будут считаны оттуда. Если же атрибут `standalone` имеет значение `"yes"`, то использование внешних DTD описаний будет запрещено.

### Определение элемента

Элемент в DTD определяется с помощью дескриптора **!ELEMENT**, в котором указывается название элемента и структура его содержимого.

Например, для элемента `<flower>` можно определить следующее правило:

```
<!ELEMENT flower PCDATA>
```

Ключевое слово **ELEMENT** указывает, что данной инструкцией будет описываться элемент XML. Внутри этой инструкции задается название элемента (`flower`) и тип его содержимого.

В определении элемента мы указываем сначала название элемента (`flower`), а затем его модель содержимого - определяем, какие другие элементы или типы данных могут встречаться внутри него. В данном случае содержимое элемента `flower` будет определяться при помощи специального маркера **PCDATA** (что означает *parseable character data* - любая информация, с которой может работать программа-анализатор). Существует еще две инструкции, определяющие тип содержимого: **EMPTY**, **ANY**. Первая указывает на то, что элемент должен быть пустым (например, `<red/>`), вторая - на то, что содержимое элемента специально не описывается.

Последовательность дочерних для текущего элемента объектов задается в виде списка разделенных запятыми названий элементов. При этом для того, чтобы указать количество повторений включений этих элементов могут использоваться символы `+`, `*`, `?` :

- `"+"` означает, что соответствующий элемент может встречаться один или более раз;
- `"?"` означает, что может быть не более одного элемента, или же элемент может отсутствовать вообще;
- `"*"` означает, что элемент может отсутствовать или появляться один или более раз.

```
<!ELEMENT issue (title, author+, table-of-contents?)>
```

В этом примере указывается, что внутри элемента `<issue>` должны быть определены элементы `title`, `author` и `table-of-contents`, причем элемент `title` является обязательным элементом и может встречаться лишь однажды, элемент `author` может встречаться несколько раз, а элемент `table-of-contents` является опциональным, т.е. может отсутствовать. В том случае, если существует несколько возможных вариантов содержимого определяемого элемента, их следует разделять при помощи символа `"|"` :

```
<!ELEMENT flower (PCDATA | title)*>
```

Символ `*` в этом примере указывает на то, что определяемая последовательность внутренних элементов может быть повторена несколько раз или же совсем не использоваться.

### Определение атрибутов

Списки атрибутов элемента определяются с помощью ключевого слова **!ATTLIST**. Внутри него задаются названия атрибутов, типы их значений и дополнительные параметры. Например, для элемента `<article>` могут быть определены следующие атрибуты:

```
<!ATTLIST article
id ID #REQUIRED
about CDATA #IMPLIED
type (actual | review | teach ) 'actual' "
```

&gt;

В данном примере для элемента *article* определяются три атрибута: *id*, *about* и *type*, которые имеют типы ID (идентификатор), CDATA и список возможных значений соответственно. Всего существует шесть возможных типов значений атрибута:

- CDATA - содержимым документа могут быть любые символьные данные
- ID - определяет уникальный идентификатор элемента в документе
- IDREF( IDREFS )- указывает, что значением атрибута должно выступать название(или несколько таких названий, разделенных пробелами во втором случае) уникального идентификатора определенного в этом документе элемента
- ENTITY( ENTITIES) - значение атрибута должно быть названием(или списком названий, если используется ENTITIES) компонента (макροопределения), определенного в документе
- NMTOKEN (NMTOKENS) - содержимым элемента может быть только одно отдельное слово(т.е. этот параметр является ограниченным вариантом CDATA)
- Список допустимых значений - определяется список значений, которые может иметь данный атрибут.

Также в определении атрибута можно использовать следующие параметры:

- #REQUIRED - определяет обязательный атрибут, который должен быть задан во всех элементах данного типа
- #IMPLIED - атрибут не является обязательным
- #FIXED "значение" - указывает, что атрибут должен иметь только указанное значение, однако само определение атрибута не является обязательным, но в процессе разбора его значение в любом случае будет передано программе-анализатору
- Значение - задает значение атрибута по умолчанию

### Определение сущностей

Сущность (entity) представляет собой определения, содержимое которых может быть повторно использовано в документе. В других языках программирования подобные элементы называются макроопределениями. Создаются DTD-сущности при помощи инструкции !ENTITY:

```
<!ENTITY hello ' Мы рады приветствовать Вас!' >
```

Программа-анализатор, просматривая в первую очередь содержимое области DTD-определений, обработает эту инструкцию и при дальнейшем разборе документа будет использовать содержимое DTD-сущности в том месте, где будет встречаться его название. Т.е. теперь в документе мы можем использовать выражение &hello;, которое будет заменено на строчку *"Мы рады приветствовать Вас"*

В общем случае, внутри DTD можно задать три типа сущностей:

**Внутренние сущности** - предназначены для определения строковой константы, с их помощью можно организовывать ссылки на часто изменяемую информацию, делая документ более читабельным. Внутренние компоненты включаются в документ при помощи амперсанта & .

**Внешние сущности** - указывают на содержимое внешнего файла, причем этим содержимым могут быть как текстовые, так и двоичные данные. В первом случае в месте использования

макроста будут вставлены текстовые строки, во втором - бинарные данные, которые анализатором не рассматриваются и используются внешними программами

```
<!ENTITY logotype SYSTEM "/image.gif" NDATA GIF87A>
```

**Макроопределения правил** - макроопределения параметров могут использоваться только внутри области DTD и обозначаются специальным символом %, вставляемым перед названием макроста. При этом содержимое компонента будет помещено непосредственно в текст DTD-правила

### XML-схемы

Схемы данных (Schema) являются альтернативным способом создания правил построения XML-документов. По сравнению с DTD, схемы обладают более мощными средствами для определения сложных структур данных, обеспечивают более понятный способ описания грамматики языка, способны легко модернизироваться и расширяться. Безусловным достоинством схем является также то, что они позволяют описывать правила для XML-документа средствами самого же XML.

### Область схемы данных

Создавая схемы данных, мы определяем в документе специальный элемент, **<schema>**, внутри которого содержатся описания правил:

```
<schema id="OurSchema">
<!-- последовательность инструкций -->
</schema>
```

Если использовать отдельное пространство имен, то полный XML-документ, содержащий в себе схему данных, будет выглядеть следующим образом:

```
<?XML version='1.0' ?>
<?xml:namespace href="http://www.mrcpk.nstu.ru/schemas/" as="s"/?>
<s:schema id="OurSchema">
<!-- последовательность инструкций -->
</s:schema>
```

### Описание элементов

Для определения класса элемента, к которому в дальнейшем будут применяться инструкции, описывающие его содержимое и структуру, предназначен специальный элемент схемы **elementType**,

```
<elementType id="issue">
<descript>Элемент содержит информацию об очередном выпуске журнала</descript>
</elementType>
```

Название элемента задается атрибутом **id**. Все дальнейшие инструкции, которые относятся к описываемому классу, определяют его внутреннюю структуру и набор допустимых данных, содержатся внутри блока, заданного тэгами **<elementType>** и **</elementType>**. Мы рассмотрим эти инструкции чуть позже.

Как видно из примера, при определении класса элемента, можно также использовать комментарии к нему, которые заключаются в тэги **<descript></descript>**

### Атрибуты элемента

Для того, чтобы в описании элемента определить его атрибуты и описать свойства этих атрибутов мы должны использовать элемент **attribute**:

```
<elementType id="photo">
```

```
<attribute name="src"/>
<empty/>
</elementType>
```

В данном примере элементу *<photo>* определяется атрибут *src*, значением которого может быть любая последовательность разрешенных символов:

```
<photo src="0"/>
<photo src="some text">
```

Подобно DTD, схемы данных позволяют устанавливать ограничения на значения и способ использования атрибутов. Для этого в дескрипторе *<attribute>* необходимо использовать параметр **atttype**.

Например, если мы хотим указать, что значение атрибута должно использоваться программой-анализатором как уникальный идентификатор, то нам необходимо создать следующее правило:

```
<elementType id="bouquet">
<attribute name="id" atttype="ID">
</elementType>
```

Если же требуется задать список возможных значений атрибута, то пример будет выглядеть следующим образом:

```
<attribute name = "flower" atttype = "ENUMERATION" values = "red green" default="red">
```

Для приведенных примеров корректным будет являться следующий фрагмент XML-документа:

```
<bouquet id="0">
  <flower color="red">rose</flower>
  <flower color="green">leaf</flower>
</bouquet>
```

### Модель содержимого элемента

Под моделью содержимого в схеме данных понимают описание всех допустимых объектов XML- документа, использование которых внутри данного элемента является корректным. Модель содержимого определяется инструкциями, расположенными внутри блока *<elementType>*.

```
<elementType id="article">
<attribute name="id" atttype="ID">
<element type="#title">
<string/>
</elementType>
```

Вложенные элементы описываются при помощи инструкции **element**, в которой параметром *type* указывается класс объекта - ссылка на его определение:

```
<elementType id="article">
<element type="#title"/>
<element type="#author"/>
</elementType>
```

Если требуется указать режим использования вложенного элемента, то надо определить параметр **occurs**:

```
<elementType id="article">
<element type="#title" occurs="REQUIRED"/>
<element type="#author" occurs="OPTIONAL"/>
```

```
<element type="#subject" occurs="ONEORMORE"/>
</elementType>
```

Возможные значения этого параметра таковы:

- **REQUIRED** - элемент должен быть обязательно определен
- **OPTIONAL** - использование элемента не является обязательным
- **ZEROORMORE** - вложенный элемент может встречаться несколько раз или ни разу
- **ONEORMORE** - элемент должен встречаться хотя бы один раз

Кроме элементов, содержащим XML-документа могут также является обычный текст и области CDATA. Для обозначения типов содержимого текущего элемента в схемах используются следующие инструкции:

- **<string/>** - указывает на то, что содержимым элемента является только свободная текстовая информация (секция PCDATA) :
 

```
<elementType id="flower">
  <string/>
</elementType>
```
- **<any/>** - указывает на то, что содержимым элемента должны являться только элементы, без текста, незаключенного ни в один элемент:
 

```
<elementType id="issue">
  <any/>
</elementType>
```
- **<mixed>** - любое сочетание элементов и текста
 

```
<elementType id="contacts">
  <mixed/>
</elementType>
```
- **<empty>** - пустой элемент

Пример:

```
<elementType id="title">
  <string/>
</elementType>
<elementType id="chapter">
  <string/>
</elementType>
<elementType id="chapters-list">
  <any/>
</elementType>
<elementType id="content">
  <element type="#chapters-list" occurs="OPTIONAL">
</elementType>
```

### Группировка элементов

Элемент **group** используется для того, чтобы задать некоторую последовательность вложенных объектов:

```
<elementType id="contacts">
  <element type="#tel" occurs="ONEORMORE">
  <group occurs="OPTIONAL">
```

```

<element type="#email">
<element type="#url">
</group>
</elementType>

```

Группировка объектов позволяет определять сразу группу объектов различных типов, которые могут находиться внутри данного объекта. В приведенном примере мы указали, что внутри объекта типа `contacts` могут быть включены элементы `tel`, `email`, и `url`, причем атрибутом `occurs` мы указали, что элементы в группе являются необязательными. Корректным для таких схем будут являться следующие фрагменты документов:

```

<contacts>
<tel>123-12-12</tel>
<email>info@j.com</email>
<url>http://www.j.com</url>
</contacts>
...
<contacts>
<tel>123-12-12</tel>
</contacts>
...
<contacts>
<tel>123-12-12</tel>
<email>info@j.com</email>
</contacts>

```

При помощи атрибута **groupOrder** можно также задавать режим использования группированных элементов. При установленном значении `OR` возможно использование не всех элементов группы, а лишь некоторых из них. Если задано значение `AND`, то оба элемента должны быть включены в обязательном порядке. Например:

```

<elementType id="contacts">
<element type="#tel" occurs="ONEORMORE">
<group groupOrder="AND" occurs="OPTIONAL">
<element type="#email">
<element type="#url">
</group>
</elementType>

```

### Использование правил из внешних схем

Схема может использовать элементы и атрибуты из других схем. Для этого надо использовать атрибут `href`, в котором указывается название внешней схемы. Например:

```

<?XML version='1.0' ?>
<?xml:namespace name="urn:uuid:BDC6E3F0-6DA3-11d1-A2A3-00AA00C14882/" as="s"/?>
<s:schema>
<elementType id="author">
<string/>
</elementType>

```

```

<elementType id="title">
<string/>
</elementType>
<elementType id="Book">
<element type="#title" occurs="OPTIONAL"/>
<element type="#author" occurs="ONEORMORE"/>
<element href="http://mrcpk.org/" /> </elementType></s:schema>
</elementType>
</s:schema>

```

### Сущности схем

Сущности, или макроопределения, используются в схемах точно также, как и в DTD. Для их определения предназначены тэги **<intEntityDcl/>** и **<extEntityDcl/>**;

```
<intEntityDcl name="RSUH">
```

Российский Государственный Гуманитарный Университет

```
</intEntityDcl>
```

```
<extEntityDcl name="logo" notation="#gif" systemId="http://www.rsuh.ru/logo.gif"/>
```

### Пример DTD, схемы и документа

Примеры DTD, схемы и документа хранятся в файлах SAarticle.dtd, SAarticleMS.xdr, SAarticleSample.xml.

### Отчет

Отчет обязательно должен содержать следующие пункты:

- 1) Название работы
- 2) Цели работы
- 3) Этапы выполнения работы
- 4) Приложения:
  - пример документа, созданного с применением формата SAarticle,
  - собственное DTD или схема документа, созданные в процессе выполнения работы,
  - пример документа, описанного в соответствии с созданным DTD (схемой).
- 5) Выводы о проделанной работе

### *Практическая работа №4*

«Изучение классификационных языков на примерах конкретных информационных классификаций (УДК, ГРНТИ, МПК)»

Цели практической работы:

1. Изучить более подробно один из классификационных языков (например, УДК)
2. Научиться описывать ПОД документа с использованием индексов УДК

Задачи практической работы.

1. Изучить методику индексирования, применяемую в УДК.
2. Ознакомиться со структурой, свойствами и принципами УДК.
3. Научиться создавать индексы УДК для документов.

Этапы выполнения:

1. Загрузить локальную копию сайта «УДК» с рабочего стола
2. Изучить разделы «Структура, свойства и принципы УДК», «Методические указания к применению УДК» из вступительной части.
3. Ответить письменно на следующие вопросы:
  1. Что такое УДК и для чего она используется?
  2. Свойства УДК?
  3. УДК построена по систематическому принципу. Что это означает?
  4. Как обеспечивается многоаспектное индексирование документов в УДК?
  5. Сколько классов в УДК?
  6. Как достигается детализация индекса?
  7. Знаки УДК?
  8. Составные части УДК?
  9. Специальные определители?
  10. Общие определители?
  11. Что понимают под методикой индексирования?
  12. Основные правила индексирования по УДК?
4. Письменно объяснить состав индекса для двух произвольно выбранных печатных работ.
5. Проиндексировать используя сложный индекс и знаки УДК два собственных «произведения печати».
6. Представить отчет, в соответствии с принятыми правилами.

### ***Практическая работа №5***

Анализ функциональных возможностей и информационных ресурсов русскоязычных ЭБ (УИС «Россия», ЦБОР, ГПНТБ)

Цели практической работы:

1. Изучить функциональные возможности информационных ресурсов русскоязычных ЭБ (на примере УИС «Россия», ЦБОР, ГПНТБ)
2. Научиться эффективно пользоваться возможностями предоставляемыми ЭБ.

Задачи практической работы.

1. Ознакомиться со структурой и возможностями ЭБ.
2. Выполнить сравнительный анализ указанных в задании библиотек.  
Представить отчет, в соответствии с принятыми правилами.

#### 9.2. Методические рекомендации по подготовке письменных работ.

Письменными работами по данной дисциплине являются рефераты, а также отчеты о практических работах, которые обучающиеся выполняют и оформляют в соответствии с требованиями, изложенными в Практикуме по дисциплине «Лингвистическое обеспечение информационных систем».

### **9.2 Методические рекомендации по подготовке письменных работ**

#### **Порядок составления и оформления отчета о практической работе**

В значительной мере эффективность решения задачи по выполнению практической работы зависит от качества соответствующего отчета. Для этого необходимо соблюдать следующие основные требования по составлению и оформлению отчета, обусловленные соответствующими нормативными документами. Текст отчета должен быть лаконичным и вместе с тем информативным. Текст должен быть изложен с соблюдением правил грамматики. Отчет составляется с обязательным составлением следующих разделов:

1. Заголовок отчета.
2. Цели работы.
3. Методика работы.
4. Порядок выполнения работы (этапы работы).
5. Выводы по работе.

1. В **заголовке отчета** приводятся наименования идентифицирующих признаков: **Отчет о практической работе № 1** по теме, например, «**Введение в XML. DTD и XML-документы**», ниже указываются данные студента (фамилия и инициалы, вид обучения, специальность, курс, группа).

2. В разделе **Цель работы** формулируется цели работы студента в соответствии с содержанием раздела «Постановка задачи» данной работы и индивидуального задания студенту на работу.

3. В разделе **Методика работы** указывается методика работы в соответствии с имеющейся формулировкой в разделе «Методика работы» данной работы и при необходимости уточняется в зависимости от содержания конкретного варианта задания студенту на практическую работу.

4. **Порядок выполнения работы.** Приводятся номера и наименования этапов работы, предусмотренные для работы данного Практикума. По каждому из этапов приводится описание выполненных студентом работ, направленных на достижение цели работы. Пропуск какого-либо из этапов работы Практикума не допускается. В рамках этапов помещается соответствующий иллюстративный материал - таблицы, рисунки (графики), полученные по ходу решения задачи работы. Обозначение иллюстративного материала выполняется в соответствии с правилами, принятыми для публикаций. Обозначение каждой таблицы и рисунка должно иметь номер и наименование. Внутри каждого отчета таблицы и рисунки обозначаются соответственно сквозными номерами. Обозначение таблицы указывается над таблицей, а обозначение рисунка под рисунком. Приводимые в тексте данной работы примеры включать в отчет не разрешается. Применяется только материал, полученный в ходе работы студентом по соответствующему заданию, полученному от преподавателя.

5. Последним разделом отчета являются **выводы** по работе. Это самая сложная и трудная часть работы. Очень важно, чтобы выводы отражали методику, технологию, применяемые программно-аппаратные средства решения задачи. Полезно каждому из этапов работы формулировать не менее одного вывода. Вывод может содержать от одного до трех предложений. Формулировки выводов должны быть конкретными, информативными, лаконичными, по возможности подкрепляться количественными данными.

Оформление отчета выполняется с учетом общепринятых правил. Графическая часть отчетов должна соответствовать правилам графического оформления. Текст отчета набирается в редакторе Word через 1,5 интервала, 14 кегль. Следует использовать шрифт Times New Roman. Заголовки разделов и подразделов выделяются жирным шрифтом. После окончания оформления отчета он проверяется студентом на предмет качество содержания и формы. При условии обнаружения ошибок последние исправляются. После устранения дефектов отчета его экранная форма, или принтерная распечатка предьявляется преподавателю. При условии обнаружения преподавателем ошибок в отчете студент их исправляет и предьявляет отчет преподавателю повторно. Если ошибок нет, то отчет принимается и сохраняется на жестком диске.

Отчет по работе сохраняется студентом в виде отдельного файла. В имени файла указывается фамилия студента и номер выполненной работы. Файл сохраняется в папке с фамилией студента в папке соответствующей студенческой группы. Папка группы создается на

первом занятии. В имени папки группы должен присутствовать индекс группы. Папка группы включается в папку «Мои документы».

## АННОТАЦИЯ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ

Цель дисциплины: профессиональная подготовка студентов, необходимая для освоения методов и подходов применения информационных технологий, программных систем и технических средств для классификации, кодирования и решения задач автоматизированной обработки лингвистических единиц, описывающих объекты и субъекты предметной области информационных систем. Задачи дисциплины: получение систематизированных знаний о современных компьютерных технологиях, используемых в лингвистическом обеспечении информационных систем; изучение методов защиты информации в автоматизированных системах классификации и кодирования элементов обрабатываемых лингвистическим обеспечением информационных систем; приобретение знаний о структуре и функциях автоматизированных систем лингвистического обеспечения и роли этих систем в процессе создания, эксплуатации и дистрибуции информационных систем; сформировать умения использования нормативно-правовых документов для оценки лингвистического обеспечения информационных систем; владение навыками работы с лингвистическим обеспечением элементов и компонентов электронного документооборота; сформировать умения самостоятельного принятия решения о внедрении тех или иных технологий лингвистического обеспечения для повышения эффективности информационных систем. развитие адекватного системного мышления, умение однозначно излагать свои мысли, развитие способностей к обобщению и анализу элементов лингвистического обеспечения, постановке задач классификации и кодирования объектов обработки информационных систем и выбору путей их достижения.

Задачи дисциплины:

1. получение систематизированных знаний о современных компьютерных технологиях, используемых в лингвистическом обеспечении информационных систем;
2. изучение методов защиты информации в автоматизированных системах классификации и кодирования элементов, обрабатываемых лингвистическим обеспечением информационных систем;
3. приобретение знаний о структуре и функциях автоматизированных систем лингвистического обеспечения и роли этих систем в процессе создания, эксплуатации и дистрибуции информационных систем;
4. сформировать умения использования нормативно-правовых документов для оценки лингвистического обеспечения информационных систем;
5. владение навыками работы с лингвистическим обеспечением элементов и компонентов электронного документооборота;
6. сформировать умения самостоятельного принятия решения о внедрении тех или иных технологий лингвистического обеспечения для повышения эффективности информационных систем.
7. развитие адекватного системного мышления, умение однозначно излагать свои мысли, развитие способностей к обобщению и анализу элементов лингвистического обеспечения, постановке задач классификации и кодирования объектов обработки информационных систем и выбору путей их достижения.

В результате освоения дисциплины обучающийся должен:

**Знать:** основную терминологию, связанную с использованием компьютерных технологий в лингвистическом обеспечении информационных систем; цели и задачи систем информационного обеспечения; методы, технологии и средства автоматизации работ по созданию и эксплуатации лингвистического обеспечения; методы моделирования и анализа предметных областей, охватываемых лингвистическим обеспечением информационных систем; специфику документационных процессов в условиях «электронного офиса»; понятие кодирования электронного документа, его составных элементов и стадий жизненного цикла;

классификацию функций информационных систем; стандарты и нормативно-правовое обеспечение в области лингвистическим обеспечением информационных систем, элементы математической лингвистики и теории формальных языков.

**Уметь:** строить структурные модели информационного обеспечения управления; разрабатывать модели информационных и документационных потоков; осуществлять контроллинг и мониторинг за реализацией процессов и документопотоков в соответствии с требованиями модели; проводить анализ эффективности организации системы ДОУ; выбирать информационные технологии и программные системы для решения задач документационного обеспечения управления; разрабатывать требования к ИС в части ЛО; разрабатывать концептуальную модель прикладной области, проводить оценку эффективности внедрения и применения систем управления электронными документами предприятия.

**Владеть:** методами и программными средствами лингвистического обеспечения информационных систем, навыками применения методов анализа моделей данных, документопотоков и информационных потоков в информационных системах с целью их оптимизации и реорганизации; навыками работы с системами лингвистического обеспечения различного типа; организации совместной работы с адекватными средствами лингвистического обеспечения в среде корпоративной информационной системы.