Федеральное государственное бюджетное образовательное учреждение высшего образования
«РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА И ГОСУДАРСТВЕННОЙ СЛУЖБЫ ПРИ ПРЕЗИДЕНТЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»

Нижегородский институт управления – филиал РАНХиГС
(наименование института (факультета))
Кафедра информатики и информационных технологий
(наименование кафедры)

УТВЕРЖДЕНО
И.о. директора Нижегородского института управления – филиала РАНХиГС

Лебедева Екатерина Александровна
Подписано цифровой подписью: Лебедева Екатерина Александровна
Дата: 2025.06.18 09:51:33 +03'00'

ПРОГРАММА МАГИСТРАТУРЫ

«Информационные технологии в государственном и муниципальном управлении»
(наименование образовательной программы)

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ, реализуемой с применением электронного (онлайн) курса

Б1.В.01.04 ПРОЕКТИРОВАНИЕ WEB-СЕРВИСОВ ИНФОРМАЦИОННЫХ СИСТЕМ
(индекс и наименование дисциплины, в соответствии с учебным планом)

09.04.03 Прикладная информатика
(код и наименование направления подготовки)
очная, заочная
(форма обучения)

Год набора – 2025

Нижний Новгород, 2025 г.

Автор-составитель:

Заведующий кафедрой информатики и информационных технологий, канд. техн. наук, доцент	И.И. Гребенюк
(ученая степень и(или) ученое звание, должность, наименование кафедры)	*(Ф.И.О.)*
Доцент кафедры информатики и информационных технологий, канд. физ.-мат. наук,	М.Л. Орлов
(ученая степень и(или) ученое звание, должность, наименование кафедры)	*(Ф.И.О.)*
Заведующий кафедрой информатики и информационных технологий, канд. техн. наук, доцент	И.И. Гребенюк
(ученая степень и(или) ученое звание, должность, наименование кафедры)	*(Ф.И.О.)*

РПД Б1.В.01.04 Проектирование web-сервисов информационных систем одобрена на заседании кафедры информатики и информационных технологий.
Протокол №13 от 04.02.2025 г.

Содержание

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения программы 4

2. Объем и место дисциплины в структуре ОП ВО 5

3. Содержание и структура дисциплины 5

4. Материалы текущего контроля успеваемости обучающихся 7

5. Оценочные материалы промежуточной аттестации по дисциплине 27

6. Методические материалы по освоению дисциплины с применением ЭК 30

7. Учебная литература и ресурсы информационно-телекоммуникационной сети Интернет 32

7.1 Основная литература 32

7.2 Дополнительная литература 32

7.3 Учебно-методическое обеспечение самостоятельной работы. 33

7.4 Нормативные правовые документы. 33

7.5. Интернет-ресурсы, справочные системы. 34

8. Материально-техническая база, информационные технологии, программное обеспечение и информационные справочные системы 34

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения программы

1.1 Дисциплина «Проектирование web-сервисов информационных систем» обеспечивает овладение следующими компетенциями:

Таблица 1.

Код компетенции	Наименование компетенции	Код компонента компетенции	Наименование компонента освоения компетенции
ПК-6	Способен выбирать методологию и технологию проектирования ИС с учетом проектных рисков	ПК-6.1	Способен использовать современные методологии проектирования информационных систем.
ПК-11	Способен проектировать архитектуру и сервисы ИС предприятий и организаций в прикладной области	ПК-11.1	Способен применять способы интеграции сервисов и баз данных информационных систем.
ПК-13	Способен принимать эффективные проектные решения в условиях неопределенности и риска	ПК-13.2	Способен определять показатели эффективности проектных решений.
ПК-17	Способен организовывать и проводить переговоры с представителями заказчика и профессиональные консультации на предприятиях и в организациях	ПК-17.2	Способен создавать проекты по разработке и внедрению информационных систем и технологий.

1.2 В результате освоения дисциплины у студентов должны быть сформированы:

Таблица 2.

ОТФ/ТФ/ профессиональные действия	Код этапа компетенции	Планируемые результаты обучения
ОТФ: управление работами по сопровождению и проектами создания (модификации) ИС, автоматизирующих задачи организационного управления и бизнес-процессы (D); ТФ: • разработка инструментов и методов адаптации бизнес-процессов заказчика к возможностям ИС в рамках управления работами по сопровождению и проектами создания (модификации) ИС (D/09.7); • организационное и технологическое обеспечение разработки баз данных ИС в рамках управления работами по сопровождению и проектами создания (модификации) ИС (D/17.7).	ПК-11.1	На уровне знаний: – возможности ИС; – архитектура, устройство и функционирование вычислительных систем; – программные средства и платформы инфраструктуры информационных технологий организаций; – инструменты и методы моделирования бизнес-процессов организации На уровне навыков: – проектирования и разработки сервисов ИС в прикладных областях.
ОТФ: Управление работами по сопровождению и проектами создания (модификации) ИС, автоматизирующих задачи организационного управления и бизнес-процессы (D); ТФ: • организационное и технологическое обеспечение разработки	ПК-13.2	На уровне знаний: – оценка (прогнозирование) бюджетов и графиков: метод аналогов, экспертные оценки. На уровне навыков: – управления командой разработчиков.

ОТФ/ТФ/ профессиональные действия	Код этапа компетенции	Планируемые результаты обучения
баз данных ИС (D/17.7).		
ОТФ: управление работами по сопровождению и проектами создания (модификации) ИС, автоматизирующих задачи организационного управления и бизнес-процессы (D); ТФ: • организационное и технологическое обеспечение разработки баз данных ИС в рамках управления работами по сопровождению и проектами создания (модификации) ИС (D/17.7).	ПК-17.2	На уровне знаний: – Методы управления содержанием проекта: документирование требований, анализ продукта, моделируемые совещания. На уровне навыков: – создания проектов по разработке и внедрению информационных систем и технологий.

2. Объем и место дисциплины в структуре ОП ВО

Дисциплина Б1.В.01.04. «Проектирование web-сервисов информационных систем» является дисциплиной части, формируемой участниками образовательных отношений.

Общая трудоемкость дисциплины составляет 3 зачетных единицы, 108 академических часов / 81 астрономический час.

Освоение дисциплины опирается на необходимый объем теоретических знаний и практических навыков в области проектирования информационных систем и программирования.

Дисциплины, на которых базируется данная дисциплина: Технологии межведомственного электронного взаимодействия, шифр дисциплины Б1.В.01.05; Тенденции развития Web-пространства, шифр дисциплины Б1.О.07; Управление внедрением IT-проектов, шифр дисциплины Б1.О.06.

Дисциплины, для которых данная дисциплина является предшествующей: Научно-исследовательская работа, шифр Б2.В.01(Н), Преддипломная практика, шифр Б2.В.03(Пд).

При очной форме обучения на контактную работу с преподавателем выделено 48 ч, в том числе на лекционные занятия – 8 ч, на практические (семинарские) занятия – 40 ч; на самостоятельную работу обучающихся выделено 56 ч.

При заочной форме обучения на контактную работу с преподавателем выделено 12 ч, в том числе на лекционные занятия – 4 ч, на практические (семинарские) занятия – 8 ч; на самостоятельную работу обучающихся выделено 92 ч.

Форма промежуточной аттестации в соответствии с учебным планом – зачет.

3. Содержание и структура дисциплины

3.1 Структура дисциплины

Таблица 3.

№ п/п	Наименование тем (разделов)	Объем дисциплины (модуля), час.						Форма текущего контроля успеваемости**, и промежуточной аттестации
		Всего	Контактная работа обучающихся с преподавателем по видам учебных занятий				СР	
			Л/ЭО, ДОТ*	ЛР/ ЭО, ДОТ*	ПЗ/ ЭО, ДОТ*	КСР		
Очная форма обучения								
Тема 1	Введение в веб-сервисы	20	2		6		12	О, ПЗ

№ п/п	Наименование тем (разделов)	Объем дисциплины (модуля), час.						Форма текущего контроля успеваемости**, и промежуточной аттестации
		Всего	Контактная работа обучающихся с преподавателем по видам учебных занятий				СР	
			Л/ЭО, ДОТ*	ЛР/ ЭО, ДОТ*	ПЗ/ ЭО, ДОТ*	КСР		
Тема 2	Основы XML: семантика и синтаксис	20	2		6		12	О, ПЗ
Тема 3	Язык описания веб-сервисов	20	2		6		12	О, ПЗ
Тема 4	Универсальный интерфейс распознавания, описания и интеграции	24	2		10		12	О, ПЗ
Тема 5	Реализация веб-сервисов на различных платформах	24			12		12	О, ПЗ
Промежуточная аттестация								Зачет
Всего (в з.е./акад.часах/астр.часах):		**3/108/81**	**8/6**		**40/30**		**56/42**	4/3
Заочная форма обучения								
Тема 1	Введение в веб-сервисы	20	1		1		18	О, ПЗ
Тема 2	Основы XML: семантика и синтаксис	20	1		1		18	О, ПЗ
Тема 3	Язык описания веб-сервисов	20	1		2		17	О, ПЗ
Тема 4	Универсальный интерфейс распознавания, описания и интеграции	22	1		2		19	О, ПЗ
Тема 5	Реализация веб-сервисов на различных платформах	22			2		20	О, ПЗ
Промежуточная аттестация		**4**						Зачет
Всего (в з.е./акад.часах/астр.часах):		**3/108/81**	**4/3**		**8/6**		**92/69**	**4/3**

Примечание:
** – при применении электронного обучения, дистанционных образовательных технологий в соответствии с учебным планом*
*** – формы текущего контроля успеваемости: опрос (О), тестирование (Т),), оценка доклада (Д), оценка работы с документами в микрогруппе (М),оценка участия в диспуте (Д)с, отчет о выполненнии практического задания (ПЗ), контрольная работа (КР), расчетно-графическая работа (РГР)).*

3.2 Содержание дисциплины

Тема 1 Введение в веб-сервисы

Архитектура электронного правительства. Реализация «Стратегии развития информационного общества».

Предоставление государственных и муниципальных услуг органами власти средствами веб-сервисов и мобильных приложений. Веб-сервисы как способы вывода информации и данных.

Тема 2 Основы XML: семантика и синтаксис

Язык XML как универсальное средство обеспечения интеграции ИС. Связь XML с другими языками текстовой разметки и его особенности. Создание XML-документов. XML теги и XML элементы.

Тема 3 Язык описания веб-сервисов

Сервисно-ориентированная архитектура приложений. Протоколы и стандарты веб-сервисов: SOAP, WSDL, UDDI. Среда разработки Visual Studio.net, «АСМО-графический редактор».

Тема 4 Универсальный интерфейс распознавания, описания и интеграции

Интеграция данных и массивов элементов в информационных системах. Автоматизация бизнес-схем и бизнес-процессов в различных информационных системах. Проектирование веб-сервиса для связи различных ИС

Тема 5 Реализация веб-сервисов на различных платформах

Анализ порталов предоставления государственных и муниципальных услуг органов власти различных уровней. Проектирование веб-сервиса и разработка мобильного приложения для предоставления государственной (муниципальной) услуги.

4. Материалы текущего контроля успеваемости обучающихся

4.1. В ходе реализации дисциплины Б1.В.01.04. «Проектирование web-сервисов информационных систем» используются следующие **методы текущего контроля** успеваемости обучающихся:

Таблица 4.

Тема (раздел)	Методы текущего контроля успеваемости
Тема 1 Введение в веб-сервисы	Опрос, контроль выполнения заданий
Тема 2 Основы XML: семантика и синтаксис	Опрос, контроль выполнения заданий
Тема 3 Язык описания веб-сервисов	Опрос, контроль выполнения заданий
Тема 4 Универсальный интерфейс распознавания, описания и интеграции	Опрос, контроль выполнения заданий
Тема 5 Реализация веб-сервисов на различных платформах	Опрос, контроль выполнения заданий

4.2. Типовые материалы текущего контроля успеваемости обучающихся

Вопросы для текущего контроля успеваемости (опросов) по тема 1-5:

1. Что такое web-сервисы? Приведите примеры.
2. Назовите основные показатели реализации «Стратегии развития информационного общества» в Нижегородском регионе.
3. Какие web-сервисы доступны на портале государственных и муниципальных услуг Нижегородской области?
4. Что такое XML тег?
5. Назовите XML элементы. Какие существуют правила именования XML элементов?
6. Опишите функционал среды разработки Visual Studio.net.
7. Опишите функционал среды разработки «АСМО-графический редактор»
8. Что такое протоколы web-сервисов?
9. Назовите стандарты для разработки web-сервисов.
10. Какие инструментальные средства можно использовать для автоматизации бизнес-процессов?
11. Назовите средства проектирования web-сервисов.
12. Какие инструментальные средства и среды разработки используются для создания мобильных приложений?
13. Назовите особенности эксплуатации и продвижения мобильных приложений.

Вопросы (задания) для самостоятельной выполнения и подготовки к занятиям (пример)

1. Изучить функционал интернет-порталов органов государственной и муниципальной власти. Найти основные услуги, предоставляемые в электронном виде через порталы.
2. Описать структуру портала предоставления государственных услуг www.gosuslugi.ru. Найти и описать функционал веб-сервисов портала.
3. Найти информацию об использовании мобильных приложений, как средства интеграции с базами данных информационных систем органов власти.
4. Создать сравнительную таблицу функционала порталов предоставления услуг с использованием веб-сервисов нескольких муниципальных образований (реестр услуг, административный регламент, используемая платформа и т.д.).

Примерное практическое задание

Практическая работа №1. Введение в веб-сервисы

Учебные вопросы:

1. Архитектура электронного правительства.
2. Реализация «Стратегии развития информационного общества».
3. Предоставление государственных и муниципальных услуг органами власти средствами веб-сервисов и мобильных приложений.

Задания для самостоятельной выполнения и подготовки к занятиям:

1. Изучить функционал интернет-порталов органов государственной и муниципальной власти. Найти основные услуги, предоставляемые в электронном виде через порталы.
2. Описать структуру портала предоставления государственных услуг www.gosuslugi.ru. Найти и описать функционал веб-сервисов портала.
3. Найти информацию об использовании мобильных приложений, как средства интеграции с базами данных информационных систем органов власти.
4. Создать сравнительную таблицу функционала порталов предоставления услуг с использованием веб-сервисов нескольких муниципальных образований (реестр услуг, административный регламент, используемая платформа и т.д.).

Практическая работа №2 Язык XML

Учебные вопросы:

1. Язык XML как универсальное средство обеспечения интеграции ИС.
2. Связь XML с другими языками текстовой разметки и его особенности.
3. Создание XML-документов.
4. XML теги и XML элементы.

Задания для самостоятельной выполнения и подготовки к занятиям:

1. Сделайте обзор и анализ наиболее популярных инструментальных средств построения веб-сервисов информационных систем.
2. Предложите возможные проблемные области для создания веб-сервиса.
3. Проанализируйте с точки зрения эффективности работы любой информационный ресурс, содержащий элементы веб-сервисов. Предложите систему критериев оценки эффективности работы ресурса.
4. Предложите свою модель этого информационного ресурса на основе современных информационных технологий и инструментальных сред.
5. Разработайте структуру информационного ресурса и веб-сервиса.

Практическая работа №3 Статистическая и динамическая модели XML документ.

Учебные вопросы:

1. Изучить конструкции языка XML, способы описания структуры документа средствами DTD, описания схемы документа на языке XSD. Изучить способы обработки документов XML.
2. Написать по созданной в практической работе № 2 статистической и динамической модели XML документ.
3. Описать структуру документа средствами DTD.
4. Описать схему документа на языке XSD
5. Преобразовать документ средствами XSLT
6. Преобразовать документ с помощью DOM
7. Оформить отчет, включающий постановку задачи, тексты XML, DTD, XSD, XSLT документов, демонстрацию работы программ.
8. Защитить лабораторную работ

Практическая работа №4 Среда разработки Visual Studio.net

Учебные вопросы:

1. Протоколы и стандарты веб-сервисов: SOAP, WSDL, UDDI.
2. Среда разработки Visual Studio.net.

Задания для самостоятельной выполнения и подготовки к занятиям:

Изучить среду разработки Visual Studio.net.

Практическая работа №5 Среда разработки «АСМО-графический редактор»

Учебные вопросы:

1. Протоколы и стандарты веб-сервисов: SOAP, WSDL, UDDI.
2. Среда разработки «АСМО-графический редактор».

Задания для самостоятельной выполнения и подготовки к занятиям:
Изучить среду разработки «АСМО-графический редактор».

Практическая работа №6 Создание Web-сервисов

Содержание работы

1. Изучить основные принципы создания Web-сервисов.
2. Создать web-сервис, который будет использовать XML документ в качестве источника данных. То есть создать Web- сервис, возвращающий обычный набор данных содержащий информацию, взятую из XML-файла.
3. Дополнить созданный web- сервис процедурой, позволяющей сохранять данные в XML файле.
4. Создать клиентское приложение.
5. Оформить отчет, включающий постановку задачи, текст программ, демонстрацию работы программ.
6. Защитить лабораторную работу

Контрольная работа

Вариант 1

1. Дано *DTD*:

<!DOCTYPE base_of_students

```
[
<!ELEMENT base_of_students (university+)>
<!ELEMENT university (student+)>
<!ATTLIST university nameU ID #REQUIRED>
<!ELEMENT student (name,course,group,mark,form,faculty,adress)>
<!ELEMENT name (surname,first,second)>
<!ELEMENT surname (#PCDATA)>
<!ELEMENT first (#PCDATA)>
<!ELEMENT second (#PCDATA)>
<!ELEMENT course (#PCDATA)>
<!ELEMENT group (#PCDATA)>
<!ELEMENT mark (#PCDATA)>
<!ELEMENT form (#PCDATA)>
<!ELEMENT fuculty (#PCDATA)>
<!ELEMENT adress (#PCDATA)>
]
>
```

Написать соответствующий данному *DTD* XML-*файл*.

2. Дан XML-*файл*:

```
<?xml version="1.0" encoding="windows-1251" ?>
<?xml-stylesheet type='text/xsl' href=7apos;1.xsl'?>
<documents>
  <book>
        <title>Языки информационного обмена </title>
        <type> Лекции </type>
        <file>Лекции ЯИО.doc</file>
  </book>
  <book>
        <title>Языки информационного обмена </title>
        <type>Лабораторные работы</type>
        <file>Лабораторные работы ЯИО.doc</file>
  </book>
  <book>
        <title>Администрирование ИС</title>
        <type>Лекции</type>
        <file>Лекции АИС.doc</file>
  </book>
  <book>
        <title>Администрирование ИС</title>
        <type>Лабораторные работы</type>
        <file>Лабораторные работы АИС.doc</file>
  </book>
</documents>
```

Написать *файл* 1.xsl для отображения содержимого XML-файла. Содержимое тега <file> должно отображаться как *ссылка*.

Вариант 2

1. Дана часть схемы документа:

```
<xs:element name="adress" minOccurs="0" maxOccurs="unbounded">
<xs:complexType>
<xs:sequence>
```

```
<xs:element name="city" type="xs:string" minOccurs="0" />
<xs:element name="street" type="xs:string" minOccurs="0" />
<xs:element name="house" type="xs:string" minOccurs="0" />
<xs:element name="zip" type="xs:string" minOccurs="0" />
<xs:element name="phone-list" minOccurs="1" maxOccurs="unbounded">
<xs:complexType>
<xs:sequence>
<xs:element name="phone" nillable="true" minOccurs="1" maxOccurs="unbounded">
<xs:complexType>
<xs:simpleContent msdata:ColumnName="phone_Text" msdata:Ordinal="0">
<xs:extension base="xs:string"></xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
```

Написать фрагмент XML-файла удовлетворяющий данному фрагменту.

2. Дан XML-*файл*:

```
<?xml version="1.0" encoding="windows-1251" ?>
<?xml-stylesheet type='text/xsl' href='1.xsl'?>
<documents>
  <book>
        <title>Языки информационного обмена </title>
        <type> Лекции </type>
        <file>
<id>1</id>
<name>Лекции ЯИО.doc</name>
</file>
  </book>
  <book>
        <title>Языки информационного обмена </title>
        <type>Лабораторные работы</type>
        <file>
<id>3</id>
<name> Лабораторные работы ЯИО.doc </name>
</file>
  </book>
  <book>
        <title>Администрирование ИС</title>
        <type>Лекции</type>
        <file>
<id>2</id>
<name>Лекции АИС.doc</name>
</file>
  </book>
  <book>
        <title>Администрирование ИС</title>
        <type>Лабораторные работы</type>
```

```
            <file>
<id>4</id>
<name> Лабораторные работы АИС.doc </name>
</file>

  </book>
</documents>
```

Написать *файл* 1.xsl для отображения содержимого XML-файла. Содержимое тега <file> должно только в случае если id<3.

Вариант 3

1. Дано *DTD*:

```
<!DOCTYPE notebook
[
<!ELEMENT notebook (person+)>
<!ELEMENT person (name, birthday?, address*, phone-list)>
<!ELEMENT name EMPTY>
<!ATTLIST name
first CDATA #IMPLIED
second CDATA #IMPLIED
surname CDATA #REQUIRED>
<!ELEMENT birthday (#PCDATA)>
<!ELEMENT address (street,city,zip)?>
<!ELEMENT street (#PCDATA)>
<!ELEMENT city (#PCDATA)>
<!ATTLIST city
type (город | поселок | деревня) "город">
<!ELEMENT zip (#PCDATA)>
<!ELEMENT phone-list (work-phone*, home-phone*)>
<!ELEMENT work-phone (#PCDATA)>
<!ELEMENT home-phone (#PCDATA)>
]
>
```

Написать соответствующий данному *DTD* XML-*файл*.

2. Дан XML-*файл*:

```
<?xml version="1.0" encoding="windows-1251" ?>
<?xml-stylesheet type='text/xsl' href='1.xsl'?>
<documents>
  <book>
            <title>Языки информационного обмена </title>
            <type> Лекции </type>
            <file>
<id>1</id>
<name1>Лекции ЯИО.doc</name1>
<name2>Лекции ЯИО.html</name2>
</file>
  </book>
  <book>
            <title>Языки информационного обмена </title>
            <type>Лабораторные работы</type>
            <file>
```

```
<id>3</id>
<name1> Лабораторные работы ЯИО.doc </name1>
<name2> Лабораторные работы ЯИО.html </name2>
</file>
  </book>
  <book>
        <title>Администрирование ИС</title>
        <type>Лекции</type>
        <file>
<id>2</id>
<name1>Лекции АИС.doc</name1>
<name2>Лекции АИС.html</name2>

</file>
  </book>
  <book>
        <title>Администрирование ИС</title>
        <type>Лабораторные работы</type>
        <file>
<id>4</id>
<name1> Лабораторные работы АИС.doc </name1>
<name2> Лабораторные работы АИС.html </name2>
</file>

  </book>
</documents>
```

Написать *файл* 1.xsl для отображения содержимого XML-файла. Содержимое тега <file> отображается следующим образом: если id>3 отображается только содержимое тега <name1>, если id=3 отображается только содержимое тега <name2>, если id>3 отображается содержимое тегов <name1> и <name2>.

Вариант 4

1. Дана часть схемы документа:

```
<xs:element name="good" minOccurs="0" maxOccurs="unbounded">
<xs:complexType>
<xs:sequence>
<xs:element name="description" minOccurs="0" maxOccurs="unbounded">
<xs:complexType>
<xs:sequence>
<xs:element name="model" type="xs:string" minOccurs="0" />
<xs:element name="parametrs" type="xs:string" minOccurs="0" />
<xs:element name="price" type="xs:string" minOccurs="0" />
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="name" form="unqualified" type="xs:string" />
</xs:complexType>
</xs:element>
```

Написать фрагмент XML-файла удовлетворяющий данному фрагменту.

2. Дан XML-*файл*:

```
<?xml version="1.0" encoding="windows-1251" ?>
<documents>
  <book>
        <title>Языки информационного обмена </title>
        <type> Лекции </type>
        <file>Лекции ЯИО.doc</file>
  </book>
  <book>
        <title>Языки информационного обмена </title>
        <type>Лабораторные работы</type>
        <file>Лабораторные работы ЯИО.doc</file>
  </book>
  <book>
        <title>Администрирование ИС</title>
        <type>Лекции</type>
        <file>Лекции АИС.doc</file>
  </book>
  <book>
        <title>Администрирование ИС</title>
        <type>Лабораторные работы</type>
        <file>Лабораторные работы АИС.doc</file>
  </book>
</documents>
```

Отобразить содержимое XML-файла используя интерфейсы *DOM*.

Вариант 5

1. Дана часть схемы документа:

```
<xs:element name="student" maxOccurs="unbounded">
<xs:complexType>
<xs:sequence>
<xs:element name="name">
<xs:complexType>
<xs:sequence>
<xs:element name="surname" type="xs:string" />
<xs:element name="first" type="xs:string" />
<xs:element name="second" type="xs:string" />
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="faculty" type="xs:string" minOccurs="1" maxOccurs="unbounded"/>
<xs:element name="group" type="xs:string" minOccurs="1" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
</xs:element>
```

Написать фрагмент XML-файла удовлетворяющий данному фрагменту.

2. Дан XML-*файл*:

```
<?xml version="1.0" encoding="windows-1251" ?>
<?xml-stylesheet type='text/xsl' href='1.xsl'?>
<documents>
  <book>
        <title>Языки информационного обмена </title>
        <type> Лекции </type>
```

```
                <file>Лекции ЯИО.doc</ file >
< file >Лекции ЯИО.pdf</ file >
< file >Лекции ЯИО.html</ file >
  </book>
  <book>
                <title>Языки информационного обмена </title>
                <type>Лабораторные работы</type>
                < file > Лабораторные работы ЯИО.doc </file >
< file > Лабораторные работы ЯИО.html </ file >
  </book>
</documents>
```

Написать *файл* 1.xsl для отображения содержимого XML-файла в следующем виде:

Название: Языки информационного обмена
Описание: Лекции
Электронная версия: Лекции ЯИО.doc, Лекции ЯИО.pdf, Лекции ЯИО.html.

Название: Языки информационного обмена
Описание: Лабораторные работы
Электронная версия: Лабораторные работы ЯИО.doc, Лабораторные работы ЯИО.html.

Практическая работа №7 Разработка Web-службы в ASPNET

Контрольное задание

Создайте *ASP.NET* веб-службу, поддерживающую три метода, возвращающие соответственно название текущего дня недели, номер текущего дня в месяце, номер текущего дня в году.

Для получения необходимых данных на стороне сервера можно использовать свойства и методы:

```
Label1.Text = DateTime.Now.DayOfWeek.ToString();
Label1.Text = DateTime.Now.Day.ToString();
Label1.Text = DateTime.Now.DayOfYear.ToString();
```

Практическая работа №8 Разработка веб-службы в ASP.NET. Создание прокси-сборки для веб-службы

Порядок выполнения работы

1. Создание веб-службы

2.Создание прокси-сборки.

Контрольное задание

Создайте *ASP.NET* прокси-сборку на языке C++, Python, JavaScript для веб-службы, поддерживающей три метода, возвращающие соответственно название текущего дня недели, номер текущего дня в месяце, номер текущего дня в году.

Для получения необходимых данных на стороне сервера можно использовать свойства и методы:

```
Label1.Text = DateTime.Now.DayOfWeek.ToString();
Label1.Text = DateTime.Now.Day.ToString();
Label1.Text = DateTime.Now.DayOfYear.ToString();
```

Выбор метода веб-службы должен быть реализован через форму с элементом типа ListBox или RadioButton.

Практическая работа №9 Примеры разработки RSS-источников и RSS-ридеров

Порядок выполнения работы

1. Создания RSS документа с помощью PHP сценария и с использованием интерфейса DOM XML.

Подготовьте файл, содержащий сценарий на языке PHP:

```
<?php
// Создается новый XML документ.
$dfeed = new DOMDocument('1.0', 'utf-8');

// Создание корневого элемента  <rss>
$erss = $dfeed->createElement('rss');
$erss->setAttribute('version', '2.0');

//  И добавление его в дерево документа
$dfeed->appendChild($erss);

// Создание элемента <channel> и добавление его к <rss>
$echannel = $dfeed->createElement('channel');
$erss->appendChild($echannel);

// Создание и добавление в channel
// узлов <title>, <link>, <description>, <language>, <pubDate>
$echannel->appendChild( $dfeed->createElement('title', 'RSS-channel title') );
$echannel->appendChild( $dfeed ->createElement('link', 'http://www.myhp.edu') );
$echannel->appendChild( $dfeed ->createElement('description', 'my RSS demo') );
$echannel->appendChild( $dfeed ->createElement('language', 'en') );
$echannel->appendChild( $dfeed ->createElement('pubDate', date('r')) );

// Добавление к узлу <channel> 2 узлов <item>
for ( $i = 1; $i <= 2; $i++ )
{
   $eitem = $dfeed->createElement('item');
   $echannel->appendChild($eitem);

   // Создание дочерних элементов для <item>
   $eitem->appendChild( $dfeed->createElement('title', 'Item'.$i) );
   $eitem->appendChild(                                    $dfeed->createElement('link',
'http://www.myhp.edu/rss/'.$i.'.xml') );
   $eitem->appendChild( $dfeed->createElement('description', 'Description for '.$i.' item')
);
}

// Сохранение документа в файле demo.rss
$dfeed->save('demo.rss');
?>
```

Созданный файл разместите на веб-сервере, настройте права доступа (право на запись) для веб-сервера к директории, в которой размещается сценарий, либо к специальной директории, в которой будет создан файл demo.rss (это более безопасно).

Выполните сценарий по запросу с веб-браузера.

В результате выполнения сценария получится следующий документ:

В результате выполнения сценария получится следующий документ:

```
<?xml version="1.0" encoding="utf-8" ?>
<rss version="2.0">
 <channel>
  <title>RSS-channel title</title>
  <link>http://www.myhp.edu</link>
  <description>my RSS demo</description>
  <language>en</language>
  <pubDate>Thu, 06 Nov 2008 01:50:53 +0300</pubDate>
<item>
  <title>Item1</title>
  <link>http://www.myhp.edu/rss/1.xml</link>
  <description>Description for 1 item</description>
  </item>
 <item>
  <title>Item2</title>
  <link>http://www.myhp.edu/rss/2.xml</link>
  <description>Description for 2 item</description>
  </item>
  </channel>
  </rss>
```

2.Чтение RSS документа с помощью PHP сценария и с использованием интерфейса DOM XML.

Подготовьте файл, содержащий сценарий на языке PHP:

```
<?php
// Загрузка документа
$dfeed = DOMDocument::load('demo.rss');

// Чтение элемента <channel>
$echannel = $dfeed->firstChild->firstChild;

// Чтение первого дочернего элемента узла <channel> т.е.< title>
$cchild = $echannel->firstChild;

// Вывод содержимого дочерних элементов узла <channel> до элемента <item>:

print "<strong><br>";
while ( $cchild->tagName != 'item' )
{
   print $cchild->nodeValue. "<br>";
   $cchild = $cchild->nextSibling;
}
print "</strong><br>";

// Получение списка узлов <item>
$litems = $echannel->getElementsByTagName('item');

print "<br>";
foreach ( $litems as $eitem )
{
   $title = $eitem->firstChild->nodeValue;
   $link = $eitem->firstChild->nextSibling->nodeValue;
   $descr = $eitem->lastChild->nodeValue;
```

```
print "<a href=".$link.">".$title."</a> :: ".$descr."<br>";
}
print "<br>";

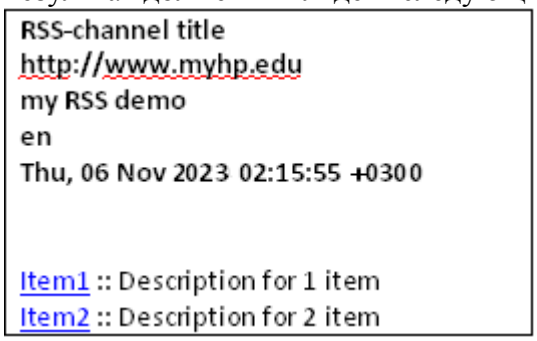
?>
```

В методе load() в качестве параметра укажите путь к RSS-файлу.

Созданный файл разместите на веб-сервере, настройте права доступа (право на запись) для веб-сервера к директории, в которой размещается сценарий, либо к специальной директории, в которой будет создан файл demo.rss (это более безопасно).

Выполните сценарий по запросу с веб-браузера.

Результат должен выглядеть следующим образом:



Контрольное задание

Разработайте PHP-сценарий, генерирующий RSS документ, содержащий описание веб-страниц на вашем локальном веб-сайте. В качестве источника информации для сценария можно использовать либо заранее подготовленный текстовый файл, либо функции PHP из библиотеки для работы с файловой системой (см., например: http://www.softtime.ru/bookphp/help.php)

Практическая работа №10 Реализация асинхронного взаимодействия веб-браузера с веб-сервером с помощью технологии AJAX

Порядок выполнения работы

В данной работе рассматривается пример системы, имитирующей работу сервиса *Google Suggest* на основе *AJAX*.

1. Реализация клиентской части.

Предполагается, что *пользователь* может вводить в текстовое *поле* формы название автомобильной марки, получая при этом динамически *список* вариантов названий, соответствующих уже введенным символам, без перезагрузки страницы.

1. Создайте следующую веб-страницу:

```
<html>
<head>
 <script src="chint.js"></script>
  </head>
    <body>
     <form>
      First Name:
     <input type="text" id="txt1" onkeyup="showHint(this.value)">
              </form>
              <p>Suggestions: <span id="txtHint"></span></p>
            </body>
```

```
</html>
```

Как видно из кода, при наступлении события onkeyup (отжатие клавиши) вызывается обработчик showHint().

13. В файле chint.js сохраните следующий код обработчика

Код обработчика загружается из файла chint.js:

```
var xmlHttp;
function showHint(str)
{
if (str.length==0)
  {
  document.getElementById("txtHint").innerHTML="";
  return;
  }
xmlHttp=GetXmlHttpObject();
if (xmlHttp==null)
  {
  alert ("Your browser does not support AJAX!");
  return;
  }
var url = "ghint.php";
url = url + "?q=" + str;
url = url + "&sid=" + Math.random();
xmlHttp.onreadystatechange = stateChanged;
xmlHttp.open("GET", url, true);
xmlHttp.send(null);
}
function stateChanged()
{
if (xmlHttp.readyState==4)
{
document.getElementById("txtHint").innerHTML = xmlHttp.responseText;
}
}
function GetXmlHttpObject()
{
var xmlHttp=null;
try
  {
  // Firefox, Opera 8.0+, Safari
  xmlHttp = new XMLHttpRequest();
  }
catch (e)
  {
  // Internet Explorer
  try
    {
    xmlHttp=new ActiveXObject("Msxml2.XMLHTTP");
    }
  catch (e)
    {
    xmlHttp=new ActiveXObject("Microsoft.XMLHTTP");
    }
```

```
  }
return xmlHttp;
}
```

Из кода видно, что каждый раз, когда вводится символ, вызывается функция-обработчик. Если при этом содержимое текстового поля формы непустое (str.length > 0), функция выполняет следующие действия:

- Формируется *url* для отправки веб-серверу
- Добавляется значение параметра q, равное содержимому текстового поля, к *url*
- Добавляется к *url* случайное число для предотвращения *кеширования*
- Создается объект XMLHTTP, при этом указывается функция (stateChanged) подлежащая исполнению при наступлении события ввода символа
- Открывается объект XMLHTTP с указанным значением *url*
- Отправляется HTTP запрос веб-серверу

Если поле ввода пустое, происходит очистка содержимого раздела txtHint на веб-странице.

Ключевым моментом в данной системе является использование объекта *XMLHttpRequest*.

Данный объект по-разному создается в различных браузерах. Так, *Internet Explorer* для этого использует *ActiveXObject*, в то время как остальные браузеры используют встроенный в JavaScript объект *XMLHttpRequest*.

Для поддержки работы системы в разных браузерах использован оператор " try-catch ".

- Сначала делается попытка создать объект *XMLHttpRequest* для браузеров *Firefox*, *Opera* или *Safari*:

```
xmlHttp = new XMLHttpRequest().
```

- В случае неудачи, делается следующая попытка создания объекта для *Internet Explorer 6.0 +* :

```
xmlHttp = new ActiveXObject("Msxml2.XMLHTTP").
```

- Если это также не удается, то делается попытка создания объекта уже для *Internet Explorer 5.5 +* :

```
xmlHttp = new ActiveXObject("Microsoft.XMLHTTP").
```

- В случае, если ни одна из этих попыток не принесла успеха, выдается сообщение об отсутствии поддержки AJAX браузером.

2. Реализация серверной части.

Разместите на веб-сервере в файле с именем ghint.php следующий *PHP сценарий*::

```
<?php
header("Cache-Control: no-cache, must-revalidate");
 // Прошедшая дата
header("Expires: Mon, 1 Sep 2008 07:30:00 GMT");
// Инициализация массива названий
$a[]="Audi";
$a[]="BMW";
$a[]="Buick";
$a[]="Chevrolet";
$a[]="Citroen";
$a[]="Dodge";
$a[]="Ferrari";
$a[]="Fiat";
$a[]="Ford";
```

```
$a[]="Honda";
$a[]="Hyundai";
$a[]="Cherokee";
$a[]="Cherry";
$a[]="Lada";
$a[]="Lamborghini";
$a[]="Lincoln";
$a[]="Mazda";
$a[]="Mercedes";
$a[]="Mitsubishi";
$a[]="Nissan";
$a[]="Opel";
$a[]="Peugeot";
$a[]="Plymoth";
$a[]="Pontiac";
$a[]="Renault";
$a[]="Rover";
$a[]="Saab";
$a[]="Subaru";
$a[]="Suzuki";
$a[]="Toyota";
$a[]="Volkswagen";
$a[]="Volvo";
//получение параметра q из URL
$q = $_GET["q"];
//поиск соответствий из массива если длина q > 0
if (strlen($q) > 0)
{
  $hint = "";
  for($i = 0; $i<count($a); $i++)
  {
  if (strtolower($q) == strtolower(substr($a[$i],0,strlen($q))))
    {
    if ($hint == "")
      {
      $hint=$a[$i];
      }
    else
      {
      $hint=$hint." , ".$a[$i];
      }
    }
  }
}
// Возврат строки "нет вариантов" если соответствий не найдено
// либо найденное соответствие
if ($hint == "")
{
$response = "no suggestion";
}
else
{
```

```
$response = $hint;
}
//вывод результата
echo $response;
?>
```

3. Проверка работоспособности системы.

Проверьте с помощью подходящего веб-браузера работу системы.

На скриншоте показана работа в браузере с веб-страницей ghint.html, использующей *AJAX*:

Контрольное задание

С использованием PHP-сценария №7 из практической работы 9 разработайте модифицированный вариант калькулятора для четырех арифметических операций, в котором происходит асинхронная передачи данных между браузером и сервером (с помощью AJAX).

На скриншоте выше показана работа в браузере с веб-страницей ghint.html, использующей AJAX:

Практическая работа №11 Создание мобильных веб-приложений при помощи ASP.NET

Порядок выполнения работы

Для работы с примерами, приводимыми в данной работе помимо среды разработки *Visual Studio .NET* потребуется установка компонентов *.NET Mobile* и *WAP* эмулятора.

В качестве *WAP* эмулятора был использован Series 40 5th Edition SDK (на основе *Nokia 6267*). Для установки эмулятора необходимо скачать с сайта http://www.*forum*.nokia.com/ (требуется предварительная *регистрация*) и установить программу Nokia *Mobile Internet Toolkit* и сам эмулятор Nokia *Series* 40 5th *Edition SDK*.

Общая последовательность разработки веб-приложения с использованием *.NET Mobile* выглядит следующим образом:

- Создание проекта веб-приложения ASP.NET.
- Подключение интерфейса System.Mobile.UI.
- Вставка в веб-страницу управляющих элементов .NET Mobile.

Рассмотрим пример создания простейшей веб-странице, отображающей на экране мобильного устройства сообщение " Hello world!".

Создайте новый проект типа ASP.NET Web Application.

Активируйте с помощью меню окно "Solution Explorer". Из списка файлов проекта удалите файлы Default.aspx.cs и Default.aspx.designer.cs.

Страница Default.aspx должна содержать следующий код:

```
<%@ Page Inherits= "System.Web.UI.MobileControls.MobilePage " Language= "C# "%>
<%@ Register TagPrefix= "mobile" Namespace= "System.Web.UI.MobileControls "
Assembly= "System.Web.Mobile " %>

<mobile:Form runat="server">
  <mobile:Label runat="server">Hello world!</mobile:Label>
</mobile:Form>
```

Директива в первой строке сообщает обработчику ASP о необходимости обработки данной страницы как предназначенной для мобильных устройств.

Директива во второй строке регистрирует префикс "mobile" для обращения к элементами управления для мобильных устройств.

Тэг <mobile:Form> указывает серверу на необходимость вставки формы.

Тэг <mobile: Label> указывает серверу на необходимость вставки метки с текстом " Hello world!".

Откомпилируйте проект.

Создайте новый виртуальный каталог, связанный с директорией текущего проекта. Добавьте в список документов по умолчанию страницу Default.aspx.

Запустите эмулятор Nokia *Series* 40 5th Edition SDK и программу Nokia *Mobile Internet* Toolkit. В программе Nokia *Mobile Internet* Toolkit укажите полный URL созданной с помощью ASP.NET страницы Default.aspx.

После обработки данной страницы для мобильного устройства, поддерживающего протокол *WAP*, будет сгенерирована страница, вид которой в эмуляторе приведен ниже:

В эмуляторе выберите раздел меню "Tools > Diagnostics". В появившемся окне диагностики можно будет увидеть следующий код страницы, загруженной в эмуляторе:

```
<?xml version='1.0'?>
 <!DOCTYPE wml PUBLIC '-//WAPFORUM//DTD WML 1.1//EN'
 'http://www.wapforum.org/DTD/wml_1.1.xml'>
 <wml>
  <head>
   <meta http-equiv="Cache-Control" content="max-age=0" />
  </head>
  <card>
   <p>Hello world!</p>
  </card>
 </wml>
```

Следующий пример демонстрирует использование простого элемента формы для ввода текста (в режиме пароля):

```
<%@ Page Inherits= "System.Web.UI.MobileControls.MobilePage " Language= "C# "%>
<%@ Register TagPrefix= "mobile " Namespace= "System.Web.UI.MobileControls "
Assembly= "System.Web.Mobile " %>

<script runat="server" language="C#">

string uname;

protected void AgeClick(Object sender, EventArgs e)
{
   uname=text1.Text;
   ActiveForm=form2;
}

protected void  Form2_Activate(Object sender, EventArgs e)
{
 message.Text="Wellcome " + uname + "!";
}
</script>

<Mobile:Form id="form1" runat="server">
<Mobile:Label runat="server">Your name, please</Mobile:Label>
<Mobile:TextBox runat="server" id="text1" password="true"/>
<Mobile:Command runat="server" OnClick="AgeClick" Text="Submit" />
</Mobile:Form>
```

```
<Mobile:Form id="form2" runat="server" OnActivate="Form2_Activate">
<Mobile:Label runat="server" id="message" />
</Mobile:Form>
```

Если выполнить как в предыдущем примере все шаги с 1 по 8, то после загрузки получившейся страницы в эмуляторе на экране отобразится первая форма:



После ввода строки и нажатия кнопки " *Submit* " активируется вторая форма:



Код страницы для первой формы, загруженный в мобильное устройство, выглядит следующим образом:

```
<?xml version='1.0'?>
<!DOCTYPE wml PUBLIC '-//WAPFORUM//DTD WML 1.1//EN'
'http://www.wapforum.org/DTD/wml_1.1.xml'>
<wml>
  <head>
    <meta http-equiv="Cache-Control" content="max-age=0" />
```

```
  </head>
  <card>
    <do type="prev" label="PќP°P•P°Pr"><prev /></do>
    <p>Your name, please<br/>
    <input name="mcsvprtws0" type="password" value="" />
    <anchor title="P'PїPµCЂPµPr">Submit
    <go href="Default.aspx?__ufps=058985" method="post">
    <postfield name="__EVENTTARGET" value="ctl01" />
    <postfield name="text1" value="$(mcsvprtws0)" />
    </go>
    </anchor>
    </p>
  </card>
</wml>
```

Контрольное задание

Разработайте с помощью MS Visual Studio .NET серверное веб-приложение, использующее управляющий ASP.NET Mobile элемент: <Mobile:Calendar> Добавьте для этого элемента обработчик события OnSelectionChanged, который должен отображать выбранный пользователем мобильного устройства день календаря с помощью элемента <Mobile:Label>.

Код обработчика события может выглядеть следующим образом:

```
<script runat="server">

protected void CalChanged(Object sender, EventArgs e)
{
   label1.Text = "You selected: " + calendar1.SelectedDate;
}
</script>
```

Создайте необходимый виртуальный каталог на веб-сервере и проверьте работу приложения с помощью эмулятора мобильного устройства.

Практическая работа №12 Инструменты создания Web-ресурсов

Цели: Ознакомиться с технологией создания веб-сайтов, создать сайт по теме «Компьютерные вирусы. Антивирусные программы».

Задачи:

- ознакомиться с основными терминами электронного обучения
- на основе карты сайта из практического занятия 6, создайте иерархическую структуру сайта (разделы и страницы)
- поберите и разместите материалы на страницах сайта

Содержание сайта

В рамках данной работы вам предстоит разработать электронный образовательный ресурс по теме «Компьютерные вирусы. Антивирусные программы». По карте сайта из работы №6 вам необходимо создать набор связанных друг с другом страниц (сайт) при помощи инструментов Google. Материалы для сайта можно найти в Интернете (презентации, видеоролики, изображения, текст...), однако ваша главная задача – правильно структурировать и отформатировать материал для сайта.

! Абсолютно не допускается бездумное копирование текста с различных сайтов без обработки.

Примерный план работы над сайтом может быть таким:

1. Продумайте содержание и структуру сайта (см. карту сайта)
2. Создайте новый сайт в среде Google.
3. Используя интернет-ресурсы, подберите материал и разместите его на страницах.
4. Опубликуйте видеоролики и презентацию в блоге.

План работы:

1) Создание сайта
2) Редактирование страниц
3) Добавление страниц, создание иерархической структуры
4) Настройка

Требования к содержанию сайта

1. На главной странице сайта разместить ссылки на страницы с презентацией и видео-роликом и краткую информацию об авторе и назначении сайта.
2. На сайте обязательно должны быть встроены **рисунки**, **видео** и **презентация.**
3. Встроить в одну из страниц форму из работы №8. На главную страницу сайта добавить ссылку, ведущую к странице с формой.
4. На странице "Об авторе" разместить логотип.

Практическая работа №13 Интеграция ASP.NET Web Forms, MVC и веб-API

В этой практической работе вы узнаете, как выполнять следующие задачи:

- Создание веб-сайта на основе одного типа проекта **ASP.NET.**
- Использование различных платформ **ASP.NET** , таких как **MVC** и **веб-API** , в одном проекте.
- Поиск основных компонентов приложения **ASP.NET.**
- Воспользуйтесь преимуществами платформы **формирования шаблонов ASP.NET** , чтобы автоматически создавать контроллеры и представления для выполнения операций CRUD на основе классов модели.
- Предоставление одинакового набора сведений в форматах машинного и удобного для чтения данных с помощью правильного инструмента для каждого задания.

Эта практическая работа включает в себя следующие упражнения:

1. Создание нового проекта веб-форм
2. Создание контроллера MVC с помощью формирования шаблонов
3. Создание контроллера веб-API с помощью формирования шаблонов

Упражнение 1. Создание нового проекта веб-форм

В этом упражнении вы создадите новый сайт Web Forms в Visual Studio 2013 с помощью единого унифицированного проекта **ASP.NET** , который позволит легко интегрировать веб-формы, MVC и компоненты веб-API в одно приложение. Затем будет рассмотрено созданное решение и его части, и, наконец, вы увидите, что веб-сайт в действии.

Задача 1. Создание нового сайта с помощью одного интерфейса ASP.NET

В этой задаче вы приступите к созданию нового веб-сайта в Visual Studio на основе одного типа проекта **ASP.NET** . **Один ASP.NET** объединяет все технологии ASP.NET и дает возможность смешивать и сопоставлять их по своему усмотрению. Затем вы узнаете о различных компонентах веб-форм, MVC и веб-API, которые параллельно находятся в приложении.

Упражнение 2. Создание контроллера MVC с помощью формирования шаблонов

В этом упражнении вы будете использовать преимущества платформы формирования шаблонов ASP.NET, предоставляемой Visual Studio для создания контроллера ASP.NET MVC 5 с действиями и представлениями Razor для выполнения операций CRUD, не написав ни одной строки кода. Процесс формирования шаблонов будет использовать Entity Framework Code First для создания контекста данных и схемы базы данных в базе данных SQL.

Задача 1. Создание новой модели

Задача 2. Создание контроллера MVC

Задача 3. Запуск решения

Упражнение 3. Создание контроллера веб-API с помощью формирования шаблонов

Платформа веб-API является частью стека ASP.NET и предназначена для упрощения реализации служб HTTP, которая обычно отправляет и получает данные в формате JSON или XML через API RESTFUL.

В этом упражнении для создания контроллера веб-API будет использоваться формирование шаблонов ASP.NET. Вы будете использовать те же классы **Person** и **персонконтекст** из предыдущего упражнения, чтобы предоставить одни и те же данные пользователя в формате JSON. Вы увидите, как можно предоставить одни и те же ресурсы различными способами в рамках одного приложения ASP.NET.

Задача 1. Создание контроллера веб-API

Задача 2. Запуск решения

Задача 3. Добавление страниц справки по веб-API

5. Оценочные материалы промежуточной аттестации по дисциплине

5.1. Зачет проводится с применением следующих методов (средств): проведение устного зачета, включающего теоретическую часть и практико-ориентированное задание, требующее демонстрации уровня владения умениями/навыками.

По дисциплине выполняется курсовой проект.

5.2. Оценочные материалы промежуточной аттестации

Таблица 5.

Показатели и критерии оценивания

Компонент компетенции	Промежуточный индикатор оценивания	Критерий оценивания
ПК-6.1 Способен использовать современные методологии проектирования информационных систем.	ИПК-6.1 Использует современные методологии проектирования информационных систем.	Применяет современные методологии проектирования информационных систем.
ПК-11.2 Способен использовать инструментарий проектирования архитектуры и сервисов ИС.	ИПК-11.1 Применяет способы интеграции сервисов и баз данных информационных систем.	Использует способы интеграции сервисов и баз данных информационных систем.
ПК-13.2 Способен определять показатели эффективности проектных решений.	ИПК-13.2 Определяет показатели эффективности проектных решений.	Имеет навыки принятия эффективных проектных решений
ПК-17.2 Способен создавать проекты по разработке и внедрению информационных	ИПК-17.2 Создает проекты по разработке и внедрению информационных систем и	Формирует проекты по разработке и внедрению информационных систем и технологий.

Компонент компетенции	Промежуточный индикатор оценивания	Критерий оценивания
систем и технологий.	технологий.	

Вопросы к зачету

1. Представление данных в информационных системах.
2. Способы вывода информации через веб-приложения и веб-сервисы.
3. Инструментальные средства в области проектирования и управления ИС.
4. Архитектура веб-приложений. Протокол HTTP. Браузеры. Веб-серверы.
5. Основные языки программирования. W3C. Языки разметки SGML, HTML, XML, RSS, KML. CSS.
6. Статические и динамические веб-страницы. Средства отладки.
7. Универсальные скриптовые языки программирования. Базовые конструкции. Блоки. Массивы. Классы. Модули. Метапрограммирование.
8. Принципы работы веб-серверов. Принципы построения веб-приложений.
9. Фильтры и обработчики запросов. Переменные окружения. Передача параметров и файлов серверным модулям приложения.
10. Шаблоны проектирования веб-сервисов.
11. Особенности управление проектами при создании web-сервисов для информационных систем предприятий и организаций.
12. Синхронный и асинхронный принципы построения веб-интерфейса. Языки и форматы обмена данными.
13. Хранение состояния на стороне браузера и на стороне сервера. Cookies, WebStorage, Sessions.
14. Способы обращения к базам данных.
15. Проблемы создания и использования информационных сервисов для автоматизации прикладных и информационных процессов. Пути их решения.
16. Проблемы безопасности веб-приложений и веб-сервисов.
17. Методы преобразования данных для интеграции веб-сервисов и информационных систем.

Примеры практико-ориентированных (практических) заданий к зачету

ПК-6.1 - способен использовать современные методологии проектирования информационных систем:

№ 1. Спроектировать web-сервис «электронная приемная». Определить необходимые поля и типы данных. Разработать формы для ввода данных.

ПК-11.2 - способен использовать инструментарий проектирования архитектуры и сервисов ИС:

№ 2. Создать страницу с формой регистрации пользователя на сайте и написать скрипт, реализующий добавление пользователя в БД сайта.

ПК-13.2 - способен определять показатели эффективности проектных решений:

№ 3. Опишите основные этапы проектирования web-сервисов для информационных систем предприятий и организаций. Определите ресурсы проекта и инструментальные средства его реализации

ПК-17.2 - способен создавать проекты по разработке и внедрению информационных систем и технологий:

№ 4. Спроектировать web-сервис для системы учета и согласования счетов к оплате с отслеживанием статуса

Шкала оценивания

Промежуточная аттестация проводится в форме зачета без применения балльно-рейтинговой системы оценивания - посредством аттестационных испытаний в форме традиционного зачета (в том числе защиты курсового проекта).

Результаты промежуточной аттестации оцениваются: по бинарной системе оценивания («зачтено», «не зачтено»).

Наряду с оцениванием учебных достижений обучающихся по принятой образовательной организации системе оценок и использованием ее при оформлении официальных дипломов Академии о высшем образовании, вводят повышенный («Excellent» и «Good») и базовый («Very good» и «Satisfactory») уровень оценок «отлично» и «хорошо» для использования при оформлении ЕПД, согласно Положению об оформлении и выдаче ЕПД.

Таблица 6.

Соответствие академической оценке

Академическая оценка		**Оценка по шкале ESTS**	
зачтено	отлично	Passed	A (Excellent)
			B (Very Good)
	хорошо		C (Good)
			D (Satisfactory)
	удовлетворительно		E (Sufficient)
не зачтено	неудовлетворительно	-	-

Критерии оценивания курсовых проектов

Оценка «отлично» выставляется обучающемуся, если он глубоко и всесторонне раскрывал тему письменной работы, последовательно, четко и логически выстраивал материал, продемонстрировал тесную связь теории и практики, использовал при написании письменной работы дополнительные научные источники и информационные образовательные ресурсы, четко сформулировал выводы по работе там, где это необходимо, предложил свои рекомендации. Оформил работу в соответствии с требованиями к письменным работам, работа прошла проверку в системе «Антиплагиат»

Оценка «хорошо» выставляется обучающемуся, если он раскрыл тему работы последовательно, четко и логично выстроил материал, продемонстрировал теоретический и практический материал, использовал рекомендованные источники, сформулировал выводы по работе, работа прошла проверку в системе «Антиплагиат»

Оценка «удовлетворительно» выставляется обучающемуся, если в работе продемонстрированы знания основного материала по теме, но при выполнении работы допущены неточности, недостаточно правильно сформулированы и нарушены логические последовательности, нечетко сформулированы выводы по работе, работа прошла проверку в системе «Антиплагиат»

Оценка «неудовлетворительно» выставляется за работу, в которой не раскрыто содержание темы, отсутствуют логика изложения материала и выводы автора, работа не прошла проверку в системе «Антиплагиат»

Критерии оценивания на зачете

Оценка **«зачтено»** выставляется обучающемуся, если он твердо знает материал, четко и логически стройно излагает его на зачете, умеет тесно увязывать теорию с практикой, правильно применяет теоретические положения при решении практических вопросов и задач, владеет необходимыми навыками и приемами их выполнения, умеет тесно увязывать теорию с практикой, свободно справляется с задачами, вопросами и другими видами заданий, причем не затрудняется с ответом при видоизменении заданий, правильно обосновывает принятое решение.

Оценка **«не зачтено»** выставляется обучающемуся, если он не знает значительной части программного материала, допускает неточности, недостаточно правильные формулировки, выполнение практических заданий неуверенно, с большими затруднениями

6. Методические материалы по освоению дисциплины с применением ЭК

Обучение по дисциплине «Проектирование web-сервисов информационных систем» предполагает изучение курса на аудиторных занятиях (лекции, практические) и самостоятельной работы студентов. Практические занятия дисциплины «Проектирование web-сервисов информационных систем» предполагают их проведение в различных формах с целью выявления полученных знаний, умений, навыков и компетенций.

Подготовка к лекции студентами заключается в следующем:

– повторить материал предыдущей лекции, прочитав его повторно;

– ознакомиться с темой предстоящей лекции (в рабочей программе учебной дисциплины);

– ознакомиться с учебными материалами по данной теме в соответствии с предложенным списком литературы в рабочей программе учебной дисциплины или с электронными материалами, предложенными лектором;

– записать возможные вопросы, которые можно будет задать лектору.

Подготовка к практическим (семинарским) занятиям:

– внимательно прочитать материал лекций относящихся к данному занятию, ознакомиться с учебными материалами, включая электронные в соответствии с предложенным списком литературы в рабочей программе учебной дисциплины;

– подготовить развернутые ответы на вопросы, предложенные в рабочей программе дисциплины для обсуждения;

– выполнить задания, если они предусмотрены в письменной форме;

– понять, что для вас осталось неясными и постараться получить на них ответ заранее;

– готовиться к практическим/семинарским занятиям можно как индивидуально, так и в составе малой группы;

– рабочую программу учебной дисциплины необходимо использовать в качестве основного ориентира в организации обучения;

– электронная версия рабочей программы по дисциплине размещена на сайте Нижегородского института управления, к ней предоставлен авторизованный доступ.

Самостоятельная работа обучающегося:

Самостоятельная работа магистрантов призвана закрепить теоретические знания и практические навыки, полученные магистрантами на лекциях, практических и семинарских занятиях.

Самостоятельная работа магистрантов проводится с целью:

– систематизации и закрепления полученных теоретических знаний и практических умений обучающихся;

– углубления и расширения теоретических знаний;

– формирования умений использовать справочную документацию и специальную литературу;

– развития познавательных способностей и активности магистрантов: творческой инициативы, самостоятельности, ответственности и организованности;

– формирования самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации;

– развития исследовательских умений.

Эффективность лекционных, семинарских и практических занятий по дисциплине во многом зависит от качества самостоятельной работы магистрантов, от их самоподготовки.

Часть времени, отведенного на самостоятельную работу должна использоваться на подготовку к аудиторным занятиям, другая часть на выполнение домашней работы, осмысление и оформление результатов практических занятий.

При подготовке к занятиям магистранту полезно:

– изучить теоретический материал по данной теме (конспект занятия);

– ознакомиться с литературой, рекомендованной преподавателем;

– выполнить задания, предложенные преподавателем, к занятию;

– составить перечень вопросов, вызывающих затруднения, неясности или сомнения, обсудить их с преподавателем или на занятии;

– заниматься самостоятельным поиском дополнительной литературы по изучаемой теме.

Подготовка к зачету. К зачету необходимо готовиться целенаправленно, регулярно, систематически и с первых дней обучения по данной дисциплине. В самом начале учебного курса познакомьтесь со следующей учебно-методической документацией:

– программой дисциплины;

– перечнем знаний и умений, которыми студент должен владеть;

– тематическими планами лекций, семинарских занятий;

– учебником, учебными пособиями по дисциплине, а также электронными ресурсами;

– перечнем и тематикой письменных работ, а также методическими рекомендациями по их выполнению;

– перечнем вопросов.

Систематическое выполнение всех видов заданий на лекциях, практических занятиях, а также самостоятельная работа позволит успешно освоить дисциплину и создать хорошую базу для сдачи зачета**.**

Вопросы для самостоятельной подготовки к занятиям

1. Что такое JAXB? Как аннотации JAXB можно применять в контексте SOAP-сервисов?

2. В чем ценность паттерна DTO применительно к веб-сервисам? Можно ли использовать этот паттерн в разработанном в соответствии с заданием веб-сервисе?

3. Предположим, что разработчик веб-сервиса не предоставил WSDL для разработки клиента к веб-сервису, а предоставил только XSD-схему. Достаточно ли XSD-схемы для разработки клиента к такому веб-сервису?

4. В чем заключается роль утилиты wsimport, и можно ли обойтись без нее?

5. В чем заключается роль утилиты wsgen, и можно ли обойтись без нее?

6. Модифицируйте REST и SOAP сервисы так, чтобы для CREATE, DELETE, UPDATE операций требовалась Basic-аутентификация. Внесите соответствующие изменения в клиентские приложения.

7. Предположим, что при реализации SOAP веб-сервиса необходимо передавать двоичные данные (аватары пользователей, архивы и др.). Какие стратегии можно предложить для работы с binary attachments? Реализуйте одну из них.

8. Обычно взаимодействие с внешним сервисом является дорогостоящей операцией и может привести к значительным задержкам. В таких случаях имеет смысл взаимодействовать с веб-сервисом асинхронно. Модифицируйте клиентское приложение таким образом, чтобы взаимодействие с веб-сервисом было асинхронным.

9. Предположим, что одним из нефункциональных требований к вашему сервису является поддержка throttling – умышленное ограничение количества одновременно

выполняемых запросов, то есть если поступает новый запрос, а в это время уже выполняется максимально разрешенное количество, то необходимо прервать выполнение запроса, например, путем выброса ThrottlingException (класс исключения нужно создать самостоятельно). Реализуйте это в вашей работе.

10. Для чего нужны аннотации @Produces и @Consumes при реализации RESTсервиса?

11. В чем принципиальное различие при реализации обработки ошибок в REST и SOAP-сервисах с точки зрения разработки сервиса, а также с точки зрения реализации клиента?

12. Реализуйте endpoint, позволяющий пользователю загружать файлы (для простоты не нужно делать никаких проверок корректности имен файлов, наличия дубликатов файлов, и других подобных операций). Реализуйте клиентскую часть, взаимодействующую с сервисом.

13. Какие методы являются идемпотентными в отношении REST? В чем заключается свойство идемпотентности?

14. Для чего используются такие компоненты сервис-ориентированной архитектуры, как ESB? Можно ли обойтись без них?

15. В чем заключается смысл использования UDDI-реестров в промышленных SOA-системах? Можно ли обойтись без реестров?

16. В каком случае вы предпочтете сервис-ориентированную архитектуру вместо монолитной? Приведите ваши рассуждения по данному вопросу.

17. Предположим, что нужно спроектировать public API для предоставления программного доступа сторонним клиентам к ресурсам корпоративной информационной системы. Что вы выберете, REST или SOAP? Какие будут основные критерии выбора? Приведите ваши рассуждения по данному вопросу.

7. Учебная литература и ресурсы информационно-телекоммуникационной сети Интернет

7.1 Основная литература

1. Коваленко, В. В. Проектирование информационных систем : учебное пособие / В.В. Коваленко. — 2-е изд., перераб. и доп. — Москва : ИНФРА-М, 2023. — 357 с. — (Высшее образование: Бакалавриат). — DOI 10.12737/987869. - ISBN 978-5-00091-783-1. - Текст : электронный. - URL: https://znanium.com/catalog/product/1894610 (дата обращения: 12.04.2025). – Режим доступа: по подписке.

2. Гвоздева, Т. В. Проектирование информационных систем: технология автоматизированного проектирования. Лабораторный практикум : учебное пособие / Т. В. Гвоздева, Б. А. Баллод. — 2-е изд., стер. — Санкт-Петербург : Лань, 2020. — 156 с. — ISBN 978-5-8114-5147-0. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/133477 (дата обращения: 12.04.2025). — Режим доступа: для авториз. пользователей.

3. Тузовский, А. Ф. Проектирование и разработка web-приложений : учебник для вузов / А. Ф. Тузовский. — Москва : Издательство Юрайт, 2025. — 219 с. — (Высшее образование). — ISBN 978-5-534-16300-1. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/561176 (дата обращения: 02.06.2025). - Режим доступа: по подписке.

7.2 Дополнительная литература

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, 2025. — 273 с. — (Высшее образование). — ISBN 978-5-534-20361-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/560485 (дата обращения: 02.06.2025).- Режим доступа: по подписке.

7.3 Учебно-методическое обеспечение самостоятельной работы.

1. Гумерова, Л. З. Основы web-программирования : учебное пособие / Л. З. Гумерова. — Красноярск : Научно-инновационный центр, 2019. — 104 c. — ISBN 978-5-6042232-6-0. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/97112.html (дата обращения: 02.03.2025). - Режим доступа: по подписке.
2. Окулич, В.И. Разработка WEB-приложений : учебно-методическое пособие . Часть 1 : Логика функционирования Web приложений. Конструирование Web страниц с использованием языка HTML и каскадных таблиц стилей (CSS) / В.И. Окулич, Н.М. Трубилов. - Нижний Новгород : Издательство ВВАГС, 2011. - 184 с. - 125-71; 125-72. - Текст : непосредственный.
3. Окулич, В.И. Разработка WEB-приложений : учебно-методическое пособие . Часть 2 : Основы PHP - основного языка создания Web-приложений (введение в процедурное программирование на PHP) / В.И. Окулич, Н.М. Трубилов. - Нижний Новгород : Издательство ВВАГС, 2011. - 298 с. - 680-00. - Текст : непосредственный.
4. Меркулова, А. Ш. Формирование баз данных : учебно-методическое пособие / А. Ш. Меркулова. — Кемерово : КемГИК, 2013. — 104 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/49651 (дата обращения: 02.06.2025). — Режим доступа: для авториз. пользователей.
5. Сергеенко, С. В. Разработка Web-приложений в Oracle Forms : учебное пособие / С. В. Сергеенко. — 2-е изд. — Москва : ИНТУИТ, 2016. — 221 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/100475 (дата обращения: 02.06.2025). — Режим доступа: для авториз. пользователей.
6. Электронные информационно-образовательные ресурсы Нижегородского института управления – филиала РАНХиГС - http://sdo.niu.ranepa.ru/

7.4 Нормативные правовые документы.

1. Федеральный закон от 29 декабря 2012 г. № 273-ФЗ "Об образовании в Российской Федерации"
2. Федеральный закон от 27 июля 2006 г. N 149-ФЗ "Об информации, информационных технологиях и о защите информации" (с изменениями и дополнениями)
3. Федеральный закон от 27 июля 2006 г. N 152-ФЗ "О персональных данных" (с изменениями и дополнениями)
4. Электронные информационно-образовательные ресурсы Нижегородского института управления – филиала РАНХиГС - http://sdo.niu.ranepa.ru/
5. Федеральный государственный образовательный стандарт высшего образования - магистратура по направлению подготовки 09.04.03 Прикладная информатика, утвержденного приказом Министерства образования и науки Российской Федерации от 19.09.2017 №916 (зарегистрировано в Министерстве юстиции Российской Федерации 10.10.2017, регистрационный номер № 48495)
6. Приказ Министерства науки и высшего образования Российской Федерации от 06.04.2021 №245 «Об утверждении Порядка организации и осуществления образовательной деятельности по образовательным программам высшего образования - программам бакалавриата, программам специалитета, программам магистратуры»

7.5. Интернет-ресурсы, справочные системы.

1. Научная электронная библиотека eLibrary.ru и информационно-аналитическая система Science index.
2. Международный электронный репозиторий публикаций ssrn.com.
3. Азбука Internet [Электронный ресурс]: интерактивный курс обучения. Электрон. дан. и прогр. М.: Новый Диск [и др.], 2009.
4. Официальный сайт RDF. URL: https://www.w3.org/RDF/.
5. Официальный сайт RDFS. URL: https://www.w3.org/TR/rdf-schema/.
6. Официальный сайт OWL. URL: https://www.w3.org/TR/owl2-quick- reference/.
7. Официальный сайт редактора онтологий Protégé. URL: https://protege.stanford.edu/.
8. Официальный сайт проекта Open Ontology Repository. URL: http://ontolog.cim3.net/wiki/OpenOntologyRepository.html.
9. Официальный сайт проекта Linked Open Vocabularies (LOV). URL: http://lov.okfn.org/dataset/lov/.

8. Материально-техническая база, информационные технологии, программное обеспечение и информационные справочные системы

8.1. Материально-техническая база

Перечень материально-технического обеспечения:

1. Учебные аудитории, оборудованные для проведения учебных занятий лекционного и семинарского типа, групповых и индивидуальных консультаций, коллоквиумов, мероприятий текущего контроля и промежуточной аттестации, в том числе мультимедийным оборудованием для демонстрации электронных презентаций и аудио- и видеоматериалов;
2. Компьютерные классы для выполнения групповых тестовых и иных заданий, а также для самостоятельной работы обучающихся оснащенные компьютерной техникой и обеспечением доступа к сети «Интернет» и доступа в электронную информационно- образовательную среду организации;
3. Специализированные аудитории и лаборатории;
4. Библиотека с обеспечением печатными изданиями или электронно-библиотечная система обеспечивающая доступ к электронным изданиям (электронная библиотека);
5. Читальный зал;
6. Технические средства обучения: персональные компьютеры; компьютерные проекторы; звуковые динамики; программные средства, обеспечивающие просмотр видеофайлов в форматах AVI, MPEG-4, DivX, RMVB, WMV и др.

8.2. Информационные технологии, программное обеспечение:

При осуществлении образовательного процесса по дисциплине используется информационные технологии и программное обеспечение:

1. Современная операционная система.
2. Kaspersky Endpoint Security (или аналог).
3. Средство просмотра файлов формата pdf.
4. Современные офисные средства (текстовые и табличные редакторы, средстваработы с презентационными материалами и т.д.).
5. Архиватор 7-Zip.
6. Система дистанционного обучения.
7. Автоматизированная библиотечная система.

8.3. Информационные справочные системы:

1. https://www.urait.ru –Электронно-библиотечная система [ЭБС] Юрайт;
2. https://www.iprbookshop.ru – Электронно-библиотечная система [ЭБС] «IPRSMART» (ранее – IPRBooks)
3. https://e.lanbook.com - Электронно-библиотечная система [ЭБС] «Лань».
4. https://znanium.ru - Электронно-библиотечная система [ЭБС] «Znanium.com».
5. https://www.book.ru - Электронно-библиотечная система [ЭБС] «Book.ru».
6. https://ibooks.ru - Электронно-библиотечная система [ЭБС] «ibooks.ru».
7. https://ranepalib.miflib.ru - Электронная библиотека издательства «МИФ».
8. https://eivis.ru – Ивис (East View). Полные тексты российских научных и практических журналов, а также газет центральной прессы России. Доступ с IP-адресов локальной сети Института.
9. https://grebennikon.ru - Полные тексты 38 научно-практических журналов по маркетингу, менеджменту, финансам и управлению персоналом ИД «Гребенников». Доступ с IP-адресов локальной сети Института.
10. http://www.consultant.ru/ - Справочно-правовая система «Консультант».
11. http://www.garant.ru/ Справочно-правовая система «Гарант».