

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Воронежский государственный технический университет»

УТВЕРЖДАЮ



И.о. декана факультета информационных
технологий и компьютерной безопасности

А.В. Бредихин /

19.03.2024

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Технологии разработки клиент-серверных приложений»

Направление подготовки 09.03.02 Информационные системы и технологии

Профиль Технологии искусственного интеллекта

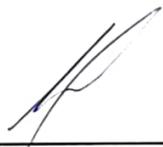
Квалификация выпускника бакалавр

Нормативный период обучения 4 года

Форма обучения очная

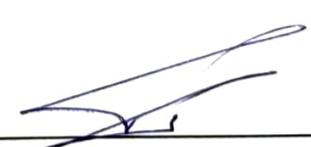
Год начала подготовки 2024

Автор программы


Е.Н. Королев

И.о. заведующего кафедрой
систем

автоматизированного
проектирования и
информационных систем


П.Ю. Гусев

Руководитель ОПОП


Д.В. Иванов

Воронеж 2024

1. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

1.1. Цели дисциплины

Формирование у студентов глубоких теоретических знаний в области разработки клиент-серверных приложений, а также практических навыков по технологиям создания клиент-серверных приложений на языке Java.

1.2. Задачи освоения дисциплины

- изучить технологии написания клиентской части клиент-серверного приложения;
- изучить технологии написания серверной части клиент-серверного приложения;
- освоить разработку двухуровневых клиент-серверных приложений;
- освоить разработку трехуровневых клиент-серверных приложений.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина «Технологии разработки клиент-серверных приложений» относится к дисциплинам части, формируемой участниками образовательных отношений блока Б1.

3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Процесс изучения дисциплины «Технологии разработки клиент-серверных приложений» направлен на формирование следующих компетенций:

ПК-1 - Способен анализировать предметную область, определять современные подходы и стандарты автоматизации в процессе проектирования и разработки информационных систем

ПК-2 - Способен разрабатывать структуры программного кода, пользовательский интерфейс и проводить анализ качества кода в процессе разработки информационных систем

ПК-3 - Способен осуществлять сбор и подготовку данных, анализировать структуры данных, проектировать и разрабатывать базы данных в рамках создания (модификации) и сопровождения информационных систем

Компетенция	Результаты обучения, характеризующие сформированность компетенции
ПК-1	знать современные подходы и стандарты автоматизации в процессе проектирования и разработки клиент-серверных приложений
	уметь выбирать современные подходы и стандарты автоматизации в процессе проектирования и разработки информационных систем
	владеть инструментальными средствами проектирования и разработки клиент-серверных

	приложений
ПК-2	знать основы разработки структуры программного кода и пользовательского интерфейса
	уметь разрабатывать структуры программного кода, пользовательский интерфейс и проводить анализ качества кода в процессе разработки информационных систем
	владеть инструментальными средствами для проектирования и разработки клиент-серверных приложений
ПК-3	знать методы сбора и подготовки данных, анализа структур данных, проектирования и разработки баз данных
	уметь выбирать и использовать современные средства сбора и подготовки данных, анализа структур данных, проектирования и разработки баз данных
	владеть инструментальными средствами сбора и подготовки данных, анализа структур данных, проектирования и разработки баз данных при разработке клиент-серверных приложений

4. ОБЪЕМ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины «Технологии разработки клиент-серверных приложений» составляет 8 з.е.

Распределение трудоемкости дисциплины по видам занятий
очная форма обучения

Виды учебной работы	Всего часов	Семестры	
		5	6
Аудиторные занятия (всего)	162	72	90
В том числе:			
Лекции	72	36	36
Лабораторные работы (ЛР)	90	36	54
Самостоятельная работа	90	36	54
Курсовой проект	+		+
Часы на контроль	36		36
Виды промежуточной аттестации зачет, экзамен	+	+	+
Общая трудоемкость:			
академические часы	288	108	180
зач.ед.	8	3	5

5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

5.1 Содержание разделов дисциплины и распределение трудоемкости по видам занятий очная форма обучения

№ п/п	Наименование темы	Содержание раздела	Лекц	Лаб. зан.	СРС	Всего, час
1.	Основы Java для клиент-серверных приложений	Особенности языка Java. Типы данных, ключевые слова, классы. Работа с файлами. Интерфейсы. Реализация интерфейсов. Исключительные ситуации. Обработка исключений. Работа с потоками. Коллекции. Stream. Лямбда выражения.	12	12	12	36
2.	Шаблоны, паттерны, рефлексия и аннотации	Адаптеры, конструкторы. Рефлексия. Объект Class. Аннотации в Java. Аннотации, используемые компилятором. Собственные аннотации. generics (дженерики)	8	8	8	24
3.	Пакет java.net	Работа с сетью. Протоколы. Работа с сетью. Классы InetAddress, URL. Работа с сетью. Классы Socket, ServerSocket. Протокол UDP. Работа с сетью. Классы DatagramSocket, DatagramPacket.	4	4	4	12
4.	Работа с удаленными базами данных	Библиотека JDBC. Объекты Connection, Statement, PreparedStatement. Библиотека JDBC, обработка SQL запросов. Объект ResultSet, ResultSetMetaData.	8	8	8	24
5.	Технологии написания серверных компонент	Servlet, JSP. Объекты HttpServletRequest, HttpServletResponse. Понятие сессии и куки. Аутентификация, обработка запросов. Основы технологии JSP. Выражения, директивы. Использование JSP с применением JavaBeans.	4	4	4	12
6.	Технологии создания клиентской части приложения	Java FX	4	4	4	12
7.	Технологии создания WEB ориентированной клиентской части приложения	JS, Ajax	4	8	8	20
8.	Технологии создания 3-х звенных клиент серверных приложений	Jakarta EE, EJB, JPA	10	10	10	30
9.	REST архитектура	JS, AJAX, Servlet, JSON	4	4	4	12
10.	SOA архитектура	XML, SOAP, XML парсеры	4	8	8	20
11.	Современные фреймворки	Spring Framework	10	20	20	50
Итого			72	90	90	252

5.2 Перечень лабораторных работ

1. Особенности языка Java.
2. Работа с потоками.
3. Работа с коллекциями и Stream.
4. Паттерны.
5. Аннотации и рефлексия.

6. Сетевые приложения. Пакет java.net.
7. Работа с БД с использованием JDBC.
8. Написание сервлетов и JSP.
9. JS, Ajax.
10. JPA.
11. Разработка приложений с REST архитектурой.
12. SOA.
13. Разработка приложений с использованием фреймворка Spring.

6. ПРИМЕРНАЯ ТЕМАТИКА КУРСОВЫХ ПРОЕКТОВ (РАБОТ) И КОНТРОЛЬНЫХ РАБОТ

В соответствии с учебным планом освоение дисциплины предусматривает выполнение курсового проекта в 6 семестре для очной формы обучения.

Примерная тематика курсового проекта: «Разработка компонент клиент-серверного приложения»

Задачи, решаемые при выполнении курсового проекта:

- выбрать технологии Java для решения поставленной задачи;
- выбрать технологии разработки клиентской части;
- выбрать технологии разработки серверной части;
- спланировать перечень работ по проекту;
- разработать соответствующие компоненты клиент-серверного приложения.

приложения.

Курсовой проект включает в себя графическую часть и расчетно-пояснительную записку.

7. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

7.1. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

7.1.1 Этап текущего контроля

Результаты текущего контроля знаний и межсессионной аттестации оцениваются по следующей системе:

«аттестован»;

«не аттестован».

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Аттестован	Не аттестован
ПК-1	знать современные подходы и стандарты автоматизации в процессе проектирования и разработки клиент-серверных приложений	Лабораторные работы 1-5	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах

	уметь выбирать современные подходы и стандарты автоматизации в процессе проектирования и разработки информационных систем	Лабораторные работы 1-5	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	владеть инструментальными средствами проектирования и разработки клиент-серверных приложений	Пояснительная записка к курсовому проекту. Лабораторные работы 1-13	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
ПК-2	знать основы разработки структуры программного кода и пользовательского интерфейса	Лабораторная работа 3,4	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	уметь разрабатывать структуру программного кода, пользовательский интерфейс и проводить анализ качества кода в процессе разработки информационных систем	Лабораторные работы 8-13	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	владеть инструментальными средствами для проектирования и разработки клиент-серверных приложений	Пояснительная записка к курсовому проекту. Лабораторные работы 1-13	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
ПК-3	знать методы сбора и подготовки данных, анализа структур данных, проектирования и разработки баз данных	Лабораторные работы 8-13	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	уметь выбирать и использовать современные средства сбора и подготовки данных, анализа структур данных, проектирования и разработки баз данных	Лабораторные работы 7-10	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	владеть инструментальными средствами сбора и подготовки данных, анализа структур данных, проектирования и	Пояснительная записка к курсовому проекту. Лабораторные работы 1-13	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах

разработки баз данных при разработке клиент-серверных приложений			
--	--	--	--

7.1.2 Этап промежуточного контроля знаний

Результаты промежуточного контроля знаний оцениваются в 5 семестре для очной формы обучения по четырехбалльной системе:

«отлично»;

«хорошо»;

«удовлетворительно»;

«неудовлетворительно».

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Отлично	Хорошо	Удовл.	Неудовл.
ПК-1	знать современные подходы и стандарты автоматизации в процессе проектирования и разработки клиент-серверных приложений	Тест	Выполнение теста на 90-100%	Выполнение теста на 80- 90%	Выполнение теста на 70-80%	В тесте менее 70% правильных ответов
	уметь выбирать современные подходы и стандарты автоматизации в процессе проектирования и разработки информационных систем	Решение стандартных практических задач	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	владеть инструментальными и средствами проектирования и разработки клиент-серверных приложений	Решение прикладных задач в конкретной предметной области	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
ПК-2	знать основы разработки структуры программного кода и пользовательского интерфейса	Тест	Выполнение теста на 90-100%	Выполнение теста на 80- 90%	Выполнение теста на 70-80%	В тесте менее 70% правильных ответов
	уметь разрабатывать структуры программного кода, пользовательский интерфейс и проводить анализ качества кода в процессе разработки	Решение стандартных практических задач	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены

	информационных систем					
	владеть инструментальным и средствами для проектирования и разработки клиент-серверных приложений	Решение прикладных задач в конкретной предметной области	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
ПК-3	знать методы сбора и подготовки данных, анализа структур данных, проектирования и разработки баз данных	Тест	Выполнение теста на 90-100%	Выполнение теста на 80-90%	Выполнение теста на 70-80%	В тесте менее 70% правильных ответов
	уметь выбирать и использовать современные средства сбора и подготовки данных, анализа структур данных, проектирования и разработки баз данных	Решение стандартных практических задач	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	владеть инструментальным и средствами сбора и подготовки данных, анализа структур данных, проектирования и разработки баз данных при разработке клиент-серверных приложений	Решение прикладных задач в конкретной предметной области	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены

7.2 Примерный перечень оценочных средств (типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности)

7.2.1 Примерный перечень заданий для подготовки к тестированию

1. Что такое сервлет?

1. Сервлет является интерфейсом Java, реализация которого расширяет функциональные возможности сервера.
2. Сервлет является интерфейсом Java, реализация которого расширяет функциональные возможности клиента.

2. Что такое контейнер сервлетов?

1. Контейнер сервлетов — программа, представляющая собой сервер, который занимается системной поддержкой сервлетов и обеспечивает их жизненный цикл в соответствии с правилами, определёнными в спецификациях.

2. Контейнер сервлетов — программа, представляющая собой клиентскую часть, которая занимается системной поддержкой сервлетов и обеспечивает их жизненный цикл в соответствии с правилами, определёнными в спецификациях.

3. Что такое дескриптор развертывания?

1. Дескриптор развертывания — это конфигурационный файл артефакта, который будет развернут в контейнере сервлетов.
2. Дескриптор развертывания — это конфигурационный файл апплета, который будет развернут в контейнере апплетов.

4. Что представляет собой объект ServletConfig?

1. Интерфейс `javax.servlet.ServletConfig` используется для передачи конфигурационной информации сервлету.
2. Интерфейс `javax.servlet.ServletConfig` используется для управления жизненным циклом сервлета.

5. Что представляет собой объект ServletContext?

1. Интерфейс `javax.servlet.ServletContext` предоставляет доступ к параметрам веб приложения сервлету.
2. Интерфейс `javax.servlet.ServletContext` предоставляет доступ к параметрам сервлета.

6. В чем отличия ServletContext и ServletConfig?

1. `ServletConfig` является уникальным объектом для каждого сервлета, в то время как `ServletContext` уникальный для всего приложения.
2. `ServletContext` является уникальным объектом для каждого сервлета, в то время как `ServletConfig` уникальный для всего приложения.

7. Что такое Request Dispatcher?

1. Интерфейс `RequestDispatcher` используется для передачи запроса другому ресурсу
2. Интерфейс `RequestDispatcher` используется для завершения работы сервлета.

8. Как получить адрес сервлета на сервере?

1. Для получения актуального пути сервлета на сервере можно использовать эту конструкцию: `getServletContext().getRealPath(request.getServletPath())`
2. Для получения актуального пути сервлета на сервере можно использовать эту конструкцию: `getServletContext().getAddress(request.getAddress())`

9. В каком случае вы будете переопределять метод `service()`?

1. Метод `service()` переопределяется, когда мы хотим, чтобы сервлет обрабатывал как GET так и POST запросы в одном методе.
2. Метод `service()` переопределяется, когда мы хотим, чтобы сервлет обрабатывал как только GET запросы в одном методе.
3. Метод `service()` переопределяется, когда мы хотим, чтобы сервлет обрабатывал как только POST запросы в одном методе.

10. Что вы знаете и в чем отличия методов `forward()` и `sendRedirect()`?

1. `RequestDispatcher forward()` используется для проброски того же самого запроса к другому ресурсу, в то время как `ServletResponse sendRedirect()` это двухшаговый метод.
2. `ServletResponse sendRedirect()` используется для проброски того же самого запроса к другому ресурсу, в то время как `RequestDispatcher forward()` это двухшаговый метод.

7.2.2 Примерный перечень заданий для решения стандартных задач

1. Что такое JPA?

1. JPA (Java Persistence API) это спецификация Java EE и Java SE, описывающая систему управления сохранением java объектов в таблицы реляционных баз данных в удобном виде.

2. JPA (Java Persistence API) это спецификация Java EE и Java SE, описывающая систему управления сохранением java объектов в XML файлы в удобном виде.

2. Может ли Entity класс наследоваться от не Entity классов (non-entity classes)?

1. Может.
2. Не может.

3. Может ли Entity класс наследоваться от других Entity классов?

1. Может.
2. Не может

4. Может ли не Entity класс наследоваться от Entity класса?

1. Может.
2. Не может

5. Может ли Entity быть абстрактным классом?

1. Может, при этом он сохраняет все свойства Entity, за исключением того, что его нельзя непосредственно инициализировать.

2. Не может.

6. Может ли встраиваемый (Embeddable) класс содержать другой встраиваемый (Embeddable) класс?

1. Может.
2. Не может.

7. **Может ли встраиваемый (Embeddable) класс содержать связи (relationship) с другими Entity или коллекциями Entity?**

1. Может, но только в случае если такой класс не используется как первичный ключ или ключ map'ы.
2. Не может.

8. **Что такое JAXB.**

1. Java Architecture for XML Binding (JAXB) позволяет Java разработчикам ставить в соответствие Java классы и XML представления.

2. Java Architecture for XML Binding (JAXB) позволяет Java разработчикам ставить в соответствие Java классы и таблицы базы данных.

9. **Какие HTTP методы поддерживаются в REST?**

1. GET; POST; PUT; DELETE; OPTIONS; HEAD;
2. GET; POST;
3. PUT; DELETE; OPTIONS; HEAD.

10. **Какая разница между GET и POST запросами?**

1. GET передает данные серверу используя URL, когда POST передает данные, используя тело HTTP запроса.

2. POST передает данные серверу используя URL, когда GET передает данные, используя тело HTTP запроса.

7.2.3 Примерный перечень заданий для решения прикладных задач

1. **Что означает аннотация @ManagedProperty?**

1. Аннотация @ManagedProperty позволяет внедрять один управляемый бин в другой.

2. Аннотация @ManagedProperty позволяет ручное управление бином.

2. **На что указывает аннотация @ApplicationScoped?**

1. Аннотация @ApplicationScoped указывает на то, что управляемый бин будет доступен на протяжении времени жизни приложения.

2. Аннотация @ApplicationScoped указывает на то, что управляемый бин будет доступен на протяжении времени жизни сессии.

3. **Объясните разницу между required и requiredMessage атрибутов в теге <h:inputText>.**

1. Эти два тега объявляют необходимость обязательного заполнения текстового поля, а так же сообщения, которое будет выполнено при нарушении этого правила.

2. Эти два тега объявляют необходимость обязательной отправки формы на обработку.

4. Объясните назначение тега <h:form>.

1. Тег <h:form> генерирует элемент формы HTML, который использует «POST» запрос для подтверждения формы и пересылки данных из формы.

2. Тег <h:form> генерирует элемент формы HTML, который использует «GET» запрос для подтверждения формы и пересылки данных из формы.

5. Какое назначение тега facelets?

1. JSF предоставляет специальный набор тегов, которая дает большую гибкость для управления общими тегами / частями в одном месте для нескольких приложений.

2. Набор тегов для прорисовки окон

6. Что такое конфигурационный файл Hibernate?

1. Файл конфигурации Hibernate содержит в себе данные о базе данных и необходим для инициализации SessionFactory.

2. Файл конфигурации JSF содержит в себе данные о базе данных и необходим для инициализации SessionFactory.

7. Что такое Hibernate mapping file?

1. Файл отображения (mapping file) используется для связи entity бинов и колонок в таблице базы данных.

2. Файл отображения (mapping file) используется для связи xml файлов и колонок в таблице базы данных.

8. Является ли Hibernate SessionFactory потокобезопасным?

1. Так как объект SessionFactory immutable (неизменяемый), то да, он потокобезопасный.

2. Нет, так как объект SessionFactory mutable (изменяемый)

9. Какие типы коллекций в Hibernate вы знаете?

1. Bag, Set, List, Array, Map

2. Vector, List, Array, Map

10. Что такое Managed Bean?

1. В технологии JSF управляемые бины применяются для сохранения состояния web-страниц.

2. В технологии JSF управляемые бины применяются для сохранения

состояния базы данных.

7.2.4 Примерный перечень вопросов для подготовки к зачету

1. Особенности и основы языка Java. Основы синтаксиса. Типы данных. Преобразование типов. Классы, работа с объектами. Основные операторы, логические операторы. Побитовые операторы.
2. Java. Основные конструкции. Ключевые слова `this`, `super`, `static`, `final`. Классы, конструкторы, объекты. Спецификаторы доступа. Инициализация.
3. Java. Реализация принципов ООП в Java. Абстрактные классы. Интерфейсы. Адаптеры. Пакеты. Внутренние классы. Анонимные классы.
4. Java. Лямбда-выражения. Потоки (`stream`) для работы с коллекциями. Ссылки на методы.
5. Обработка исключительных ситуаций. Иерархия исключений. Создание собственных исключений. Цепочки исключений, `finally`. Работа с файлами.
6. Java. Коллекции. Интерфейсы коллекций. Интерфейс `List`. Интерфейс `Map`. Интерфейс `Set`. Классы `ArrayList`, `Vector`, `LinkedList`, `HashMap`, `LinkedHashMap`, `TreeMap`, `HashSet`, `LinkedHashSet`, `TreeSet`.
7. Java. Потоки. Создание потока. Класс `Thread`. Интерфейс `Runnable`. Интерфейс `java.util.concurrent.Callable`. Процессы демоны.
8. Синхронизация потоков. Мониторы. Методы класса `Object`: `wait()`, `notify()`, `notifyAll()`. Ключевое слово `synchronized`. Семафоры. Классы `java.util.Timer` и `java.util.TimerTask`.
9. Сетевое программирование. Пакет `java.net`.
10. Java. Рефлексия. Объект `Class`. Ключевое слово `instanceof`. Получение имени класса, исследование модификаторов класса. Исследование, получение и установка значений полей класса. Исследование информации о методе, вызов метода. Загрузка и динамическое создание экземпляра класса.
11. Аннотации в Java. Аннотации, используемые компилятором. Собственные аннотации. `generics` (дженерики)
12. JDBC (Java DataBase Connectivity). Соединение с базой данных. Регистрация драйвера. Реализация интерфейса `java.sql.Connection`.
13. Объекты `Statement` и `PreparedStatement`. Обработка результатов запроса, объект `ResultSet`, навигация. Пакеты запросов. Транзакции. Класс `ResultSetMetaData`.

7.2.5 Примерный перечень заданий для подготовки к экзамену

1. Типы клиент-серверной архитектуры, технологии реализации.
2. Трехзвенная клиент-серверная архитектура, технологии реализации на Java.

3. Что такое Java EE? Какие модули входят в Java EE? Какая разница между .jar, .war и .ear файлами?
4. Что такое сервлет? Какова структура веб-проекта? Что такое контейнер сервлетов?
5. Что представляет собой объект ServletConfig? Что представляет собой объект ServletContext? В чем отличия ServletContext и ServletConfig?
6. Что такое Request Dispatcher? Как получить адрес сервлета на сервере? Как получить ip адрес клиента на сервере?
7. Каков жизненный цикл сервлета и когда какие методы вызываются? Какие методы необходимо определить при создании сервлетов?
8. Что вы знаете и в чем отличия методов forward() и sendRedirect()? Стоит ли волноваться о “многопоточной безопасности” работая с сервлетами? В чем разница между методами GET и POST?
9. Как применяются Cookies в сервлетах? Написать сервлет, реализующий загрузку файла на сервер.
10. Что такое JPA? В чем её отличие JPA от Hibernate?
11. Что такое Entity? Какие требования JPA к Entity классам вы можете перечислить (не менее шести требований)?
12. Что такое EntityManager и какие основные его функции вы можете перечислить? Какой аннотацией можно управлять кешированием JPA для данного Entity?
13. Что такое веб сервисы? В чем разница между SOA и web service?
14. Что такое SOAP? Что такое REST? В чем разница между REST и SOAP веб сервисами?
15. Расскажите о JAXB.
16. Что такое JSF? Что такое Managed Bean? Каковы три типа тегов для текстовых полей существуют в JSF?
17. Какие фазы жизненного цикла в JSF вы знаете?
18. Каковы различные типы событий в JSF?
19. Какие теги для валидации существуют в JSF?
20. Объясните архитектуру JSF.
21. Что такое jsp и зачем он нужен? Расскажите об этапах (фазах) жизненного цикла jsp.
22. Расскажите о методах жизненного цикла jsp. Какие методы жизненного цикла JSP могут быть переопределены?
23. Какие неявные, внутренние объекты и методы есть на jsp странице?
24. Что вы знаете о PageContext и какие преимущества его использования?
25. Что такое JSTL (Jsp Standard tag library)?
26. Что вы знаете о написании пользовательских jsp тегов?
27. Как можно обработать ошибки jsp страниц?
28. Что такое Hibernate Framework? Какие важные преимущества дает использование Hibernate Framework?
29. Каковы преимущества Hibernate над JDBC? Назовите некоторые важные интерфейсы Hibernate.
30. Enterprise Java Beans - EJB

- 31. Session bean, stateless
- 32. Session bean, stateful
- 33. Session bean, singleton
- 34. Основные аннотации EJB3
- 35. REST архитектура
- 36. SOA

7.2.6. Методика выставления оценки при проведении промежуточной аттестации

Экзамен проводится по билетам, каждый из которых содержит 2 вопроса.

1. Оценка «Неудовлетворительно» ставится в случае, если студент отвечает менее 50% необходимой информации.

2. Оценка «Удовлетворительно» ставится в случае, если студент отвечает более 50%, но менее 70% необходимой информации.

3. Оценка «Хорошо» ставится в случае, если студент отвечает более 70%, но менее 90% необходимой информации.

4. Оценка «Отлично» ставится, если студент отвечает более 90% необходимой информации.

7.2.7 Паспорт оценочных материалов

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции	Наименование оценочного средства
1.	Основы Java для клиент-серверных приложений	ПК-1, ПК-2, ПК-3	Тест, защита лабораторных работ, требования к курсовому проекту
2.	Шаблоны, паттерны, рефлексия и аннотации	ПК-1, ПК-2, ПК-3	Тест, защита лабораторных работ, требования к курсовому проекту
3.	Пакет java.net	ПК-1, ПК-2, ПК-3	Тест, защита лабораторных работ, требования к курсовому проекту
4.	Работа с удаленными базами данных	ПК-1, ПК-2, ПК-3	Тест, защита лабораторных работ, требования к курсовому проекту
5.	Технологии написания серверных компонент	ПК-1, ПК-2, ПК-3	Тест, защита лабораторных работ, требования к курсовому проекту
6.	Технологии создания клиентской части приложения	ПК-1, ПК-2, ПК-3	Тест, защита лабораторных работ,

		ПК-3	требования к курсовому проекту
7.	Технологии создания WEB ориентированной клиентской части приложения		Тест, защита лабораторных работ, требования к курсовому проекту
8.	Технологии создания 3-х звенных клиент серверных приложений		Тест, защита лабораторных работ, требования к курсовому проекту
9.	REST архитектура		Тест, защита лабораторных работ, требования к курсовому проекту
10.	SOA архитектура		Тест, защита лабораторных работ, требования к курсовому проекту
11.	Современные фреймворки		Тест, защита лабораторных работ, требования к курсовому проекту

7.3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Тестирование осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных тест-заданий на бумажном носителе. Время тестирования 30 мин. Затем осуществляется проверка теста экзаменатором и выставляется оценка согласно методики выставления оценки при проведении промежуточной аттестации.

Решение стандартных задач осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных задач на бумажном носителе. Время решения задач 30 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

Решение прикладных задач осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных задач на бумажном носителе. Время решения задач 30 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

8 УЧЕБНО МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ)

8.1 Перечень учебной литературы, необходимой для освоения дисциплины

1. Королев, Евгений Николаевич. Технологии сетевого программирования : учебное пособие / Е. Н. Королев ; Воронеж. гос. техн. ун-т .— Воронеж : Издательство ВГТУ, 2015 .— 115 с. ; 21 см .— Библиогр.: с. 114 .— 60.00 р., 250.

2. Королев, Е.Н. Программирование под Internet на языке Java : учеб. пособие / Е.Н.Королев .— Воронеж : ВГТУ, 2003 .— 209 с. — 57.00.
3. Королев, Е.Н. Проектирование и разработка приложений на языке Java : Учеб. пособие .— Воронеж : ГОУВПО "Воронежский государственный технический университет", 2008 .— 137 с. — 50-00.
4. Мухамедзянов Р.Р. JAVA. Серверные приложения [Электронный ресурс]/ Мухамедзянов Р.Р.— Электрон. текстовые данные.— М.: СОЛОН-ПРЕСС, 2010.— 336 с.— Режим доступа: <http://www.iprbookshop.ru/65089.html>.— ЭБС «IPRbooks»
5. Дубаков А.А. Сетевое программирование [Электронный ресурс]: учебное пособие/ Дубаков А.А.— Электрон. текстовые данные.— СПб.: Университет ИТМО, 2013.— 249 с.— Режим доступа: <http://www.iprbookshop.ru/68118.html>.— ЭБС «IPRbooks»

8.2 Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень лицензионного программного обеспечения, ресурсов информационно-телекоммуникационной сети «Интернет», современных профессиональных баз данных и информационных справочных систем:

Программное обеспечение
MySQL
NetBeans 8.1
jdk 11

Ресурс информационно-телекоммуникационной сети «Интернет»
<http://www.edu.ru/>
Образовательный портал ВГТУ

Информационная справочная система
<http://window.edu.ru>
<https://wiki.cchgeu.ru/>
<https://habr.com/ru/>
<http://javastudy.ru/interview/list-of-questions-javaee-interview/>
https://netbeans.org/kb/docs/javaee/javaee-entapp-ejb_ru.html
<http://www.knigafund.ru/> (ЭБС Книгафонд)
<http://www.book.ru/> (ЭБС BOOK.ru)
<http://ibooks.ru/> (ЭБС Ibooks (Айбукс))
<http://e.laibrary.ru>
<http://habr.com>
<http://fstec.ru>
<http://wikipedia.org>
<http://intuit.ru>

9 МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА, НЕОБХОДИМАЯ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА

Для проведения обучения по дисциплине используется компьютерный класс. Аудитория:

Компьютерный класс
Учебная аудитория для проведения учебных занятий

Комплект учебной мебели:

- рабочее место преподавателя (стол, стул);
- рабочие места обучающихся (столы, стулья)
- персональные компьютеры с установленным ПО, подключенные к сети Интернет (12 шт.);
- принтер;
- доска магнитно-маркерная поворотная

Помещение для самостоятельной работы. Читальный зал с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду.

10. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

По дисциплине «Технологии разработки клиент-серверных приложений» читаются лекции, проводятся лабораторные работы.

Основой изучения дисциплины являются лекции, на которых излагаются наиболее существенные и трудные вопросы, а также вопросы, не нашедшие отражения в учебной литературе.

Лабораторные работы выполняются на лабораторном оборудовании в соответствии с методиками, приведенными в указаниях к выполнению работ.

Вид учебных занятий	Деятельность студента
Лекция	Написание конспекта лекций: кратко, схематично, последовательно фиксировать основные положения, выводы, формулировки, обобщения; помечать важные мысли, выделять ключевые слова, термины. Проверка терминов, понятий с помощью энциклопедий, словарей, справочников с выписыванием толкований в тетрадь. Обозначение вопросов, терминов, материала, которые вызывают трудности, поиск ответов в рекомендуемой литературе. Если самостоятельно не удастся разобраться в материале, необходимо сформулировать вопрос и задать преподавателю на лекции или на практическом занятии.
Лабораторная работа	Лабораторные работы позволяют научиться применять теоретические знания, полученные на лекции при решении конкретных задач. Чтобы наиболее рационально и полно использовать все возможности лабораторных для подготовки к ним необходимо: следует разобрать лекцию по соответствующей теме, ознакомиться с соответствующим разделом учебника, проработать дополнительную литературу и источники, решить задачи и выполнить другие письменные задания.
Самостоятельная работа	Самостоятельная работа студентов способствует глубокому усвоению учебного материала и развитию навыков самообразования. Самостоятельная работа предполагает следующие составляющие: <ul style="list-style-type: none">- работа с текстами: учебниками, справочниками, дополнительной литературой, а также проработка конспектов лекций;- выполнение домашних заданий и расчетов;- работа над темами для самостоятельного изучения;

	<ul style="list-style-type: none">- участие в работе студенческих научных конференций, олимпиад;- подготовка к промежуточной аттестации.
Подготовка к промежуточной аттестации	Готовиться к промежуточной аттестации следует систематически, в течение всего семестра. Интенсивная подготовка должна начаться не позднее, чем за месяц-полтора до промежуточной аттестации. Данные перед экзаменом три дня эффективнее всего использовать для повторения и систематизации материала.

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

№ п/п	Перечень вносимых изменений	Дата внесения изменений	Подпись заведующего кафедрой, ответственной за реализацию ОПОП
----------	-----------------------------	----------------------------	--