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

ИНСТИТУТ ЭКОНОМИКИ, МАТЕМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ
ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И АНАЛИЗА ДАННЫХ
КАФЕДРА СИСТЕМНОГО АНАЛИЗА И ИНФОРМАТИКИ

УТВЕРЖДЕНА
решением кафедры Системного анализа и информатики
Протокол от «03» сентября 2020 г. № 01

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Б1.В.ДВ.04.02 Основы Java+ Фреймворк Spring
(индекс и наименование дисциплины (модуля), в соответствии с учебным планом)

09.03.03 Прикладная информатика _
(код и наименование направления подготовки (специальности))

"Веб-разработка"
направленность (профиль/специализация)

бакалавр
квалификация

очная форма обучения
форма(ы) обучения

Год набора - 2021

Москва, 2020 г.

Автор составитель:

к.э.н.. доцент кафедры Политико-правовых дисциплин
и социальных коммуникаций Новиков А.М.

Заведующий кафедрой
Системного анализа и информатики
К.техн. наук, доцент Маруев С.А.

СОДЕРЖАНИЕ

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

2. Объем и место дисциплины в структуре образовательной программы…….....

3. Содержание и структура дисциплины …….

4. Материалы текущего контроля успеваемости обучающихся и фонд оценочных средств промежуточной аттестации по дисциплине …….

5. Методические указания для обучающихся по освоению дисциплины…….....

6. Учебная литература и ресурсы информационно-телекоммуникационной сети "Интернет", учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине……..…….

6.1. Основная литература……..……..……..……

6.2. Дополнительная литература……...……..……..

6.3. Учебно-методическое обеспечение самостоятельной работы……

6.4. Нормативные правовые документы…….……..……

6.5. Интернет-ресурсы…….……..……

6.6. Иные источники…….…….…….…

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

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

1.1. Дисциплина Б1.В.ДВ.04.02 «Основы Java+ Фреймворк Spring» обеспечивает овладение следующими компетенциями с учетом этапа:

1.1.

Код компетенции	Наименование компетенции	Код этапа освоения компетенции	Наименование этапа освоения компетенции
ПКр – 1	Способен проводить обследование организаций, выявлять информационные потребности пользователей, формировать требования к информационной системе	ПКр-1.1 ПКр-1.2 ПКр-1.3	Знает принципы разработки планов коммуникаций с заказчиком в проектах создания (модификации) ИС Умеет организовывать заключение договоров на выполняемые работы, связанные с ИС Владеет навыками организации заключения договоров на выполняемые работы, связанные с ИС
ПКр-4	способность документировать процессы создания информационных систем на стадиях жизненного цикла	ПКр-4.1 ПКр-4.2. ПКр-4.3.	Знает принципы документирования существующих бизнес-процессов организации заказчика (реверс-инжиниринг бизнес-процессов организации) Умеет выполнять документирование существующих бизнес-процессов организации заказчика Владеет навыками создания пользовательской документации к ИС

УК ОС–2	Способность разработать проект на основе оценки ресурсов и ограничений	УК ОС–2.1 УК ОС–2.2 УК ОС–2.3	УК-2.1. Способен применять необходимые для осуществления профессиональной деятельности правовые нормы и методологические основы принятия управленческого решения. УК-2.2. Способен анализировать альтернативные варианты решений для достижения намеченных результатов; разрабатывать план, определять целевые этапы и основные направления работ. УК-2.3. Владеет методиками разработки цели и задач проекта; методами оценки продолжительности и стоимости проекта, а также потребности в ресурсах.

1.2. В результате освоения дисциплины у обучаемых должны быть сформированы:

ОТФ/ТФ *(при наличии профстандарта)*/ трудовые или профессиональные действия	Код этапа освоения компетенции	Результаты обучения
06.015 СПЕЦИАЛИСТ ПО ИНФОРМАЦИОННЫМ СИСТЕМАМ С Выполнение работ и управление работами по созданию (модификации) и сопровождению ИС, автоматизирующих задачи	ПКр-1.1 ПКр-1.2 ПКр-1.3	на уровне знаний: Знает принципы разработки планов коммуникаций с заказчиком в проектах создания (модификации) ИС Знает порядок подготовки и формирования документации для заключения договоров на выполняемые работы, связанные с ИС на уровне умений:

организационного управления и бизнес-процессы С/03.6 Планирование коммуникаций с заказчиком в проектах создания (модификации) и ввода ИС в эксплуатацию С/42.6 Организация заключения договоров на выполняемые работы, связанных с ИС		Умеет организовывать заключение договоров на выполняемые работы, связанные с ИС на уровне навыков: Владеет навыками выстраивания коммуникаций с заказчиком в проектах создания (модификации) и ввода ИС в эксплуатацию Владеет навыками организации заключения договоров на выполняемые работы, связанные с ИС
06.015 СПЕЦИАЛИСТ ПО ИНФОРМАЦИОННЫМ СИСТЕМАМ С Выполнение работ и управление работами по созданию (модификации) и сопровождению ИС, автоматизирующих задачи организационного управления и бизнес-процессы С/07.6 Документирование существующих бизнес-процессов организации заказчика (реверс-инжиниринг бизнес-процессов организации) С/22.6 Создание пользовательской документации к ИС	ПКр-4.1 ПКр-4.2. ПКр-4.3.	ПКр-4.1.1 Знает правила и стандарты разработки пользовательской документации к ИС ПКр-4.1.2 Знает основные принципы документирования и реинжиниринга бизнес-процессов организации ПКр-4.2.1 Умеет выполнять документирование существующих бизнес-процессов организации заказчика ПКр-4.2.2 Умеет применять правила и стандарты разработки пользовательской документации к ИС ПКр-4.3.1 Владеет навыками создания пользовательской документации к ИС ПКр-4.3.2 Владеет навыками реинжиниринга бизнес-процессов организации

	УК ОС–2.1 УК ОС–2.2 УК ОС–2.3	на уровне знаний*:* необходимые для осуществления профессиональной деятельности правовые нормы и методологические основы принятия управленческого решения. Умеет анализировать альтернативные варианты решений для достижения намеченных результатов; разрабатывать план, определять целевые этапы и основные направления работ. на уровне навыков *(типовые действия выполняются автоматически, без воспроизведения алгоритма):* Владеет методиками разработки цели и задач проекта; методами оценки продолжительности и стоимости проекта, а также потребности в ресурсах.

2. Объем и место дисциплины в структуре ОП ВО

Объем дисциплины

Объем дисциплины Б1.В.ДВ.04.02 «Основы Java+ Фреймворк Spring» в ЗЕ и академических/астрономических часах составляет 4 ЗЕ (144/108 ч). Дисциплина реализуется с применением дистанционных образовательных технологий.

Количество академических/астрономических часов по очной дистанционной форме обучения, выделенных на контактную работу 48/36 часов (в т.ч. лекц. – 16/12 ч., практ. – 32/24 ч.); на самостоятельную работу обучающихся - 60/45 часов.

Место дисциплины в структуре ОП ВО

В соответствии с учебным планом дисциплина Б1.В.ДВ.04.02 «Основы Java+ Фреймворк Spring» относится к вариативной части дисциплин по выбору учебного плана по направлению подготовки 09.03.03 Прикладная информатика. и изучается на 3 курсе в 6 семестре.
Дисциплины опирается на объём знаний информатики, вычислительных машин, сетей и систем телекоммуникаций, программирования;
Форма промежуточной аттестации для очной дистанционной формы обучения – экзамен.

3. Содержание и структура дисциплины

№ п/п	Наименование тем и/или разделов	Объем дисциплины, час.						Форма текущего контроля успеваемости**, промежуточной аттестации ***
		Всего	Контактная работа обучающихся с преподавателем по видам учебных занятий				СР	
			Л/ЭО, ДОТ*	ЛР/ ЭО, ДОТ*	ПЗ/ ЭО, ДОТ*	КСР		
Тема 1	Базовые понятия Java-технологий	26/19,5	4/3	-	10/7,5	-	12/9	О, К, Т, Р
Тема 2	Преимущества Java Spring	26/19,5	4/3	-	10/7,5	-	12/9	Т, О
Тема 3	Аспектно-ориентированное программирование	32/24	8/6	-	12/9	-	12/9	Т, О, КР
Промежуточная аттестация		36/27						Э
Всего:		**144/108**	16/12	-	32/24		60/45	

Примечание:

** – при применении электронного обучения, дистанционных образовательных технологий в соответствии с учебным планом;*

*** – формы текущего контроля успеваемости: опрос (О), тестирование (Т), контрольная работа (КР), коллоквиум (К), реферат (Р).*

**** - разработчик указывает необходимые формы промежуточной аттестации экзамен (Э).*

Содержание дисциплины

Тема 1. Базовые понятия Java-технологий

Организация сессий в Java. RMI (Remote Method Invocation) — это API, который предоставляет механизм для создания распределенного приложения на Java. Создание объектов Stub и Skeleton из класса реализации с использованием rmic. Создание и выполнение программы клиентского приложения. Особенности исполнения Java приложений в сети. Hibernate как объектно-реляционное картографическое решение с открытым исходным кодом на Java. Инструмент ORM для отображения объектов, хранящихся в базе данных. Многопоточность в Java.

Тема 2. Преимущества Java Spring

Spring как облегченный фреймворк для поддержки различных фреймворков, таких как Hibernate, Struts, Tapestry, JSF и т.д. Особенности Spring Framework, архитектура, основной модуль. Доступ к данным и интеграция. Веб-сервлеты, сокеты, портлеты, инструментарий разработки серверных приложений. Spring Web MVC Framework для деления бизнес-логики, логики представления и логики навигации веб-приложений.

Тема 3. Аспектно-ориентированное программирование

Аспектно-ориентированное программирование позволяет глобальным свойствам программы решать, как она компилируется в исполняемую программу. Аспект:

Аспект — это класс, который позволяет реализовать приложение JEE. Аспекты могут быть типичным классом, настроенным с использованием конфигурации Spring XML, или обычным классом, аннотированным с помощью аннотации @Aspect.
Точка соединения: это точка-кандидат в выполнении программы, где аспект может быть подключен. Это может быть метод, вызываемый как выбрасываемое исключение, или даже поле, которое модифицируется.
Совет: Совет — это фактические действия, предпринятые для конкретной точки соединения. Этот метод будет выполняться, когда конкретная точка соединения встречает точку сопоставления в приложении.
Pointcut: Это выражение, которое можно сопоставить с точками соединения, чтобы проверить, следует ли выполнить этот совет или нет.
Целевой объект: на основе целевого объекта применяются устройства. В AOP подкласс создается во время выполнения, где целевой метод должен быть переопределен, и рекомендации включаются в зависимости от их конфигурации.
Прокси: этот объект должен быть создан после применения рекомендации к целевому объекту.
Плетение: это метод связывания аспекта с другими объектами или типами приложения для создания рекомендованного объекта. Особенности реализации.

4. Материалы текущего контроля успеваемости обучающихся и фонд оценочных средств промежуточной аттестации по дисциплине

4.1. Формы и методы текущего контроля успеваемости обучающихся и промежуточной аттестации

4.1.1. В ходе реализации дисциплины Б1.В.ДВ.04.02 «Основы Java+ Фреймворк Spring» используются следующие методы текущего контроля успеваемости обучающихся[1]:

Тема и/или раздел		Методы текущего контроля успеваемости в аудитории	Методы текущего контроля успеваемости с применением ДОТ
Тема 1	Базовые понятия Java-технологий	Опрос, коллоквиум	Тестирование, Реферат
Тема 2	Преимущества Java Spring	Опрос	Тестирование
Тема 3	Аспектно-ориентированное программирование	Опрос	Тестирование, Контрольная работа

4.1.2. Экзамен проводится с применением следующих методов (средств):

Экзамен проводится с применением следующих методов (средств): без использования дистанционных образовательных технологий в форме выполнения практических заданий на компьютере с использованием инструментария веб-технологий, а именно LAMP/LEMP/FAMP/FEMP. HTML, CSS, JavaScript, TypeScript.

[1] *Разработчик программы самостоятельно выбирает форму заполнения пункта 1.1.*

PHP 7, MySQL. Symfony, Laravel, Zend Framework, Yii, Kohana. LESS, SASS, PostCSS. Gulp, Webpack, Babel. БЭМ. React, Vue, Angular. Git, Mercurial. Jenkins, Gitlab.

4.2. Материалы текущего контроля успеваемости обучающихся.

Преподаватель оценивает уровень подготовленности обучающихся к занятию по следующим показателям:

- опрос - устные ответы на вопросы преподавателя по теме занятия. Критерии оценивания устных ответов на вопросы преподавателя по теме занятия - правильность и полнота ответа;

- выступление с презентациями по реферату темы №1 отчетные доклады с презентациями по предлагаемым темам. Критериями оценивания реферата:

степень усвоения понятий и категорий по теме;
умение работать с документальными и литературными источниками;
грамотность изложения материала;
самостоятельность работы, наличие собственной обоснованной позиции;
Критерии оценивания презентации
оформлен титульный лист;
количество слайдов соответствует требованиям (мин.10 слайдов);
использование дополнительных эффектов в Power point, инфографики;
слайды представлены в логической последовательности.

- выполнение практических заданий на компьютере. Критерии оценивания выполнения заданий:

правильность выполнения задания;
оригинальность решений;
полнота выполнения задания.

- количество правильных ответов при тестировании. Критерии оценки выполненных студентами тестов определяются преподавателем самостоятельно. Рекомендуются следующие критерии оценки:

1. 85% – 100% правильных ответов – «отлично»;
2. 66% – 84% правильных ответов – «хорошо»;
3. 50% – 65% правильных ответов – «удовлетворительно»;
4. менее 50% правильных ответов – «неудовлетворительно».

- выполнение контрольной работы на компьютере. Критерии оценивания обучаемого в ходе выполнения:

3 балла – задание выполнено полностью, студент показал отличные знания, проявил активную позицию при ответах на вопросы, отлично владеет инструментарием

2 балла – задание выполнено с недочетами, студент показал хорошие знания, хорошо ответил на вопросы, владеет инструментарием;

1 балл – задание выполнено не полностью, студент показал слабые знания, не полностью ответил на вопросы, слабо владеет инструментарием;

0 баллов – задание не выполнено, студент не знает, не может ответить на вопросы, не владеет инструментарием.

Оценка знаний, умений, навыков проводится на основе балльно-рейтинговой системы 70% из 100% (70 баллов из 100) - вклад по результатам посещаемости

занятий, активности на занятиях, выступления с докладами, выполнение всех практических заданий и контрольных работ, правильность ответов в ходе тестирования и контрольных практических заданий.

Детализация баллов и критерии оценки текущего контроля успеваемости утверждается на заседании кафедры.

Примеры типовых оценочных материалов исходя из методов текущего контроля успеваемости

Типовые вопросы для проведения опроса

1. Возможности
2. Быстрый старт
3. Собеседование по Java EE — Spring Framework.
4. Введение в веб-модель MVC Spring
5. Spring Framework 3.0.0 RC2
6. Знакомимся со Spring Framework
7. Spring Framework
8. Java Spring Framework Masterclass

Вварианты тестовых заданий

Интерфейс - это шаблон, который содержит только сигнатуру методов. Сигнатура метода состоит из номеров параметров, типа параметра (значения, ссылки или выхода) и порядка параметров. Интерфейс не имеет самостоятельной реализации, поскольку содержит только определение методов без какого-либо тела метода. Интерфейс определяется с помощью ключевого слова *interface* . Кроме того, вы не можете создать экземпляр интерфейса. Различные функции интерфейса следующие:

- Интерфейс используется для реализации множественного наследования в коде. Эта особенность интерфейса весьма отличается от возможности абстрактных классов, потому что класс не может получить функции более чем одного класса, но может легко реализовать несколько интерфейсов.
- Он определяет конкретный набор методов и их аргументы.
- Переменные в интерфейсе должны быть объявлены как *public* , *static* и *final,* а методы должны быть *public* и *abstract* .
- Класс, реализующий интерфейс, должен реализовывать все его методы.
- Интерфейс может происходить из более чем одного интерфейса.

SOLID является аббревиатурой от первых пяти принципов объектно-ориентированного проектирования (OOD) Роберта С. Мартина.

- S - принцип единоличной ответственности . У класса должна быть одна-единственная причина для изменения, означающая, что у класса должна быть только одна работа.
- O - Open-замкнутый принцип . Объекты или объекты должны быть открыты для расширения, но закрыты для модификации.
- L - принцип подстановки Лискова . Пусть q (x) свойство, доказуемое для объектов x типа T. Тогда q (y) должно быть доказуемо для объектов y типа S, где S является подтипом T.
- I - Принцип разделения интерфейса . Никогда не следует заставлять клиента реализовывать интерфейс, который он не использует, или нельзя заставлять клиентов зависеть от методов, которые он не использует.

- D - Принцип обращения зависимостей . Сущности должны зависеть от абстракций, а не от конкрементов. В нем говорится, что модуль высокого уровня не должен зависеть от модуля низкого уровня, но они должны зависеть от абстракций.
- Возвращение nullобычно является лучшей идеей, если вы намереваетесь указать, что данные недоступны.
- Пустой объект подразумевает, что данные были возвращены, тогда как возвращение nullясно указывает, что ничего не было возвращено.
- Кроме того, возврат a nullприведет к нулевому исключению, если вы попытаетесь получить доступ к членам в объекте, что может быть полезно для выделения ошибочного кода - попытка получить доступ к элементу ничего не имеет смысла. Доступ к членам пустого объекта не потерпит неудачу, а значит ошибки могут остаться незамеченными.

Ниже приведена последовательность жизненного цикла бина в Spring:
Instantiate - сначала контейнер весны находит определение bean-компонента из файла XML и создает экземпляр bean-компонента.
- Заполнение свойств. Используя внедрение зависимостей, Spring заполняет все свойства, указанные в определении компонента.
- Set Bean Name - если bean-компонент реализует интерфейс BeanNameAware, Spring передает идентификатор bean-компонента методу setBeanName ().
- Set Bean factory - если Bean реализует интерфейс BeanFactoryAware, Spring передает beanfactory методу setBeanFactory ().
- Предварительная инициализация - также называется постобработкой бина. Если с bean-компонентом связаны какие-либо BeanPostProcessors, Spring вызывает метод postProcesserBeforeInitialization ().
- Инициализация bean-компонентов - если bean-компонент реализует IntializingBean, вызывается его метод afterPropertySet (). Если бин имеет объявление метода init, вызывается указанный метод инициализации.
- Постинициализация - если есть какие-либо BeanPostProcessors, связанные с бином, будут вызваны их методы postProcessAfterInitialization ().
- Готов к использованию. Теперь компонент готов к использованию приложением.
- Destroy - если компонент реализует DisposableBean, он вызовет метод destroy ().

Жизненный цикл Bean в Spring Bean Factory Container выглядит следующим образом:
- Контейнер Spring находит определение bean-компонента из XML-файла и создает экземпляр bean-компонента.
- Используя внедрение зависимостей, Spring заполняет все свойства, указанные в определении bean-компонента.
- Если компонент реализует интерфейс BeanNameAware, фабрика вызывает setBeanName()передачу идентификатора компонента.
- Если bean-компонент реализует интерфейс BeanFactoryAware, фабрика вызывает его setBeanFactory(), передавая свой экземпляр.
- Если есть какие-либо BeanPostProcessors, связанные с бином, ProcessBeforeInitialization()будут вызваны их пост- методы.
- Если для компонента указан метод init, он будет вызван.
- Наконец, если есть какие-либо BeanPostProcessors, связанные с bean-компонентом, postProcessAfterInitialization()будут вызваны их методы.

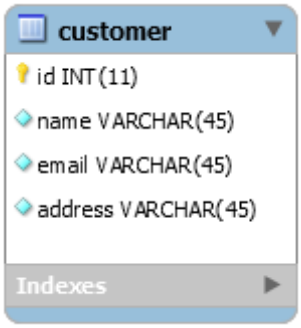
Spring MVC Framework предоставляет следующие способы, помогающие нам добиться надежной обработки исключений.

- На основе контроллера - мы можем определить методы обработчика исключений в наших классах контроллера. Все, что нам нужно, это аннотировать эти методы аннотацией @ExceptionHandler.
- Глобальный обработчик исключений - Обработка исключений является сквозной задачей, и Spring предоставляет аннотацию @ControllerAdvice, которую мы можем использовать с любым классом для определения нашего глобального обработчика исключений.
- Реализация HandlerExceptionResolver - для общих исключений, в большинстве случаев мы обслуживаем статические страницы. Spring Framework предоставляет HandlerExceptionResolverинтерфейс, который мы можем реализовать для создания глобального обработчика исключений. Причиной этого дополнительного способа определения глобального обработчика исключений является то, что Spring Framework также предоставляет классы реализации по умолчанию, которые мы можем определить в нашем файле конфигурации Spring Bean для получения преимуществ обработки исключений Spring Framework.

Варианты контрольных работ

1. Создание базы данных

Мы будем использовать MySQL. В нашем пример мы будем работать с данными в таблице **customer**, которая находится в схеме с именем **sales**. Таблица **customer** имеет 4 поля: **id**, **name**, **email** и **address**:



Вы можете запустить следующий MySQL скрипт для создания схемы и таблицы:

```
CREATE DATABASE `sales`;
CREATE TABLE `customer` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(45) NOT NULL,
  `email` varchar(45) NOT NULL,
  `address` varchar(45) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

2. Создание проекта в Eclipse

Создайте Dynamic Web Project в Eclipse, и конвертируйте его в проект Maven: для этого нажмите ПКМ по проекту, выберете **Configure** > **Convert to Maven Project**. В открывшемся диалоговом окне *Create new POM* введите следующую информацию:

— Group Id: net.codejava

— Artifact Id: CustomerManager

Также убедитесь, что версия JRE для проекта Java 8 или новее.

Затем, откройте pom.xml (файл Maven), чтобы настроить зависимости для этого проекта. Объявите свойства версий для Spring и Hibernate Frameworks:

```
<properties>
   <spring.version>5.1.5.RELEASE</spring.version>
   <hibernate.version>5.4.1.Final</hibernate.version>
</properties>
```

Укажите зависимость для Spring Framework:

```
<dependency>
   <groupId>org.springframework</groupId>
   <artifactId>spring-context</artifactId>
   <version>${spring.version}</version>
</dependency>
```

Для создания веб-приложений Spring MVC:

```
<dependency>
   <groupId>org.springframework</groupId>
   <artifactId>spring-webmvc</artifactId>
   <version>${spring.version}</version>
</dependency>
```

Для использования Spring Data JPA:

```
<dependency>
   <groupId>org.springframework</groupId>
   <artifactId>spring-orm</artifactId>
   <version>${spring.version}</version>
</dependency>
<dependency>
   <groupId>org.springframework.data</groupId>
   <artifactId>spring-data-jpa</artifactId>
   <version>2.1.5.RELEASE</version>
</dependency>
```

Мы используем Hibernate в качестве реализации JPA, поэтому добавим следующую зависимость:

```
<dependency>
   <groupId>org.hibernate</groupId>
   <artifactId>hibernate-core</artifactId>
   <version>${hibernate.version}</version>
</dependency>
```

Для того, чтобы приложение работало с MySQL, нам нужна зависимость для драйвера MySQL JDBC:

```
<dependency>
   <groupId>mysql</groupId>
   <artifactId>mysql-connector-java</artifactId>
   <version>8.0.14</version>
   <scope>runtime</scope>
```

```
</dependency>
```

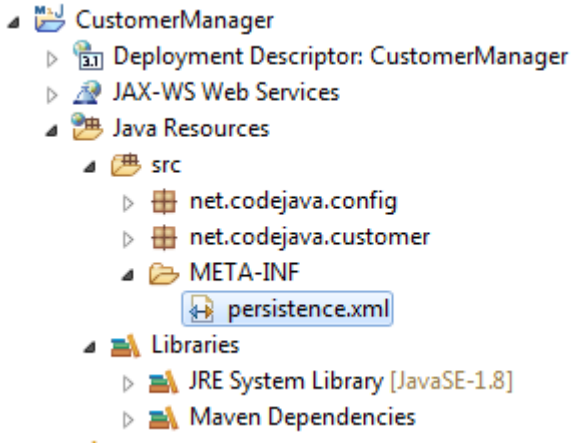
И теперь зависимости для Java Servlet, JSP и JSTL:

```
<dependency>
   <groupId>javax.servlet</groupId>
   <artifactId>javax.servlet-api</artifactId>
   <version>3.1.0</version>
   <scope>provided</scope>
</dependency>
<dependency>
   <groupId>javax.servlet.jsp</groupId>
   <artifactId>javax.servlet.jsp-api</artifactId>
   <version>2.3.1</version>
   <scope>provided</scope>
</dependency>
<dependency>
   <groupId>jstl</groupId>
   <artifactId>jstl</artifactId>
   <version>1.2</version>
</dependency>
```

Создайте два Java пакета в корне проекта:
— net.codejava.config: для классов конфигурации.
— net.codejava.customer: для классов приложения.

3. Создание файла настройки JPA

Поскольку мы используем JPA, нам нужно определить свойства для подключения к базе данных в файле persistence.xml, а не в hibernate.cfg.xml. Создайте новый каталог с именем META-INF в исходной папке проекта, чтобы поместить в него файл persistence.xml:



И впишите в этот файл код представленный ниже:

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence xmlns="http://xmlns.jcp.org/xml/ns/persistence"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence
        http://xmlns.jcp.org/xml/ns/persistence/persistence_2_1.xsd"
   version="2.1">

   <persistence-unit name="SalesDB">
```

```
        <properties>
            <property name="javax.persistence.jdbc.url"
value="jdbc:mysql://localhost:3306/sales" />
            <property name="javax.persistence.jdbc.user" value="root" />
            <property name="javax.persistence.jdbc.password" value="(password)" />
            <property name="javax.persistence.jdbc.driver" value="com.mysql.jdbc.Driver"
/>
            <property name="hibernate.show_sql" value="true" />
            <property name="hibernate.format_sql" value="true" />
        </properties>
    </persistence-unit>

</persistence>
```

Как вы можете видеть, мы указываем свойства для соединения с базой данных, такие как URL, user, password и класс JDBC драйвера. Так же обратите внимание, что имя SalesDB будет использоваться нами в коде конфигурации.

4. Создание Model Class

Создайте класс Customer, который сопоставляется с таблицей **customer** в базе данных следующим образом:

```
package net.codejava.customer;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;

@Entity
public class Customer {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String name;
    private String email;
    private String address;

    protected Customer() {
    }

    protected Customer(String name, String email, String address) {
        this.name = name;
        this.email = email;
        this.address = address;
    }

    // геттеры и сеттеры не показаны для краткости. Создайте их с помощью средств
своей IDE, или вручную.

}
```

Как вы могли увидеть, мы используем аннотацию @Entity для сопоставления этого класса с таблицей **customer** (имя класса совпадает с именем таблицы). Все имена полей класса идентичны именам полей в таблице. Поле id имеет аннотации @Id и @GeneratedValue, чтобы указать, что это поле является первичным ключом и его значение генерируется автоматически.

5. Конфигурация Spring MVC и Spring Data JPA

Далее, напишем Java код для настройки Spring MVC и Spring Data JPA. Мы будем использовать конфигурацию на основе Java, так как она проще, чем XML.

Настройка Spring Dispatcher Servlet

Для использования Spring MVC в нашем приложении, нам надо зарегистрировать Spring Dispatcher Servlet при запуске приложения, написав следующий класс:

```
package net.codejava.config;

import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.ServletRegistration;

import org.springframework.web.WebApplicationInitializer;
import
org.springframework.web.context.support.AnnotationConfigWebApplicationContext;
import org.springframework.web.servlet.DispatcherServlet;

public class WebAppInitializer implements WebApplicationInitializer {
   @Override
   public void onStartup(ServletContext servletContext) throws ServletException {
      AnnotationConfigWebApplicationContext appContext = new
AnnotationConfigWebApplicationContext();
      appContext.register(WebMvcConfig.class);

      ServletRegistration.Dynamic dispatcher = servletContext.addServlet(
            "SpringDispatcher", new DispatcherServlet(appContext));
      dispatcher.setLoadOnStartup(1);
      dispatcher.addMapping("/");

   }
}
```

Метод onStartup() этого класса будет автоматически вызываться сервлетом при загрузке приложения. Spring Dispatcher Servlet обарабатывает все запросы сопоставляя URL "/" и ищет конфигурацию в классе WebMvcConfig, которй описан ниже.

Настройка Spring MVC

Создайте класс WebMvcConfig в пакете net.codejava.config, содержащий следующий код:

```
package net.codejava.config;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.ComponentScan;
```

```
import org.springframework.context.annotation.Configuration;
import org.springframework.web.servlet.view.InternalResourceViewResolver;

@Configuration
@ComponentScan("net.codejava ")
public class WebMvcConfig {
   @Bean(name = "viewResolver")
   public InternalResourceViewResolver getViewResolver() {
      InternalResourceViewResolver viewResolver = new
InternalResourceViewResolver();
      viewResolver.setPrefix("/WEB-INF/views/");
      viewResolver.setSuffix(".jsp");
      return viewResolver;
   }
}
```

Этот класс помечен аннотацией @Configuration, сообщающей Spring, что это файл конфигурации. Аннотация @ComponentScan говорит Spring искать классы конфигурации в пакете net.codejava.

В этом классе мы создаём bean-компонент, который распознаёт представления (View), с помощью указания префикса и суффикса для этих представлений. Поэтому создайте каталог views внутри каталога WebContent/WEB-INF для хранения JSP файлов.

Здесь вы можете добавить и другие конфигурации Spring MVC.

Настройка Spring Data JPA

Для работы с Spring Data JPA нам надо создать два beans-компонента: EntityManagerFactory и JpaTransactionManager. Поэтому создадим другой конфигурационный класс JpaConfig:

```
package net.codejava.config;

import javax.persistence.EntityManagerFactory;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.data.jpa.repository.config.EnableJpaRepositories;
import org.springframework.orm.jpa.JpaTransactionManager;
import org.springframework.orm.jpa.LocalEntityManagerFactoryBean;
import org.springframework.transaction.annotation.EnableTransactionManagement;

@Configuration
@EnableJpaRepositories(basePackages = {"net.codejava.customer"})
@EnableTransactionManagement
public class JpaConfig {
   @Bean
   public LocalEntityManagerFactoryBean entityManagerFactory() {
      LocalEntityManagerFactoryBean factoryBean = new
LocalEntityManagerFactoryBean();
      factoryBean.setPersistenceUnitName("SalesDB");
```

```
        return factoryBean;
    }

    @Bean
    public JpaTransactionManager transactionManager(EntityManagerFactory
entityManagerFactory) {
        JpaTransactionManager transactionManager = new JpaTransactionManager();
        transactionManager.setEntityManagerFactory(entityManagerFactory);

        return transactionManager;
    }
}
```

Здесь мы используем две важные аннотации:

- @EnableJpaRepositories: сообщает Spring Data JPA, что нужно искать классы репозитория в указанном пакете (net.codejava) для внедрения соответсвующего кода во время выполнения.
- @EnableTransactionManagement: сообщает Spring Data JPA, чтобы тот генерировал код для управления транзакциями во время выполнения.

В этом классе первый метод создаёт экземпляр EntityManagerFactory для управления Persistence Unit нашей SalesDB (это имя указано выше в persistence.xml).

Последний метод создаёт экземпляр JpaTransactionManager для EntityManagerFactory, созданный методом ранее.

Это минимальная необходимая конфигурация для использования Spring Data JPA.

Вопросы и задания для самостоятельной подготовки

1. Создать Maven project
2. Объявить основные библиотеки Spring
3. Code Project
4. Spring @Configuration & IoC
5. Spring ApplicationContext
6. Правило работы Spring
7. Программирование веб приложения используя Spring Boot
8. Аппендикс: Download библиотека Spring

Примерные темы для написания рефератов

1. Технология MVC
2. Создание проекта
3. Инверсия управления
4. Внедрение зависимости
5. Конфигурация Maven
6. Класс *ApplicationContext*
7. Конфигурация бинов с помощью аннотаций

Шкала оценивания

Уровень освоения компетенций по дисциплине «Основы Java+ Фреймворк Spring» определяется:

- знанием содержания процессов самоорганизации и самообразования, их особенностей и технологий реализации, исходя из целей совершенствования профессиональной деятельности;
- умением найти необходимую информацию, самостоятельно решать стандартные задачи профессиональной деятельности, выполнять действия в изученной последовательности, в том числе в новых условиях, на новом содержании;
- навыками использования современных информационных визуальных цифровых технологий и способами их реализации;
- способностью самостоятельно строить процесс овладения информацией, отобранной и структурированной для выполнения профессиональной деятельности;
- умением обеспечивать сохранность здоровья себе и сотрудникам при выполнении профессиональных задач.

Низкий «неудовлетворительно/неэкзамен» - компетенция не освоена или освоена в недостаточной мере. Студент не знает, либо знает на слабом уровне теоретический материал по дисциплине. Не владеет терминологией и основными понятиями из профессиональной сферы или называет неуверенно, с ошибками.

Пороговый (базовый) «удовлетворительно/экзамен» - компетенция освоена удовлетворительно, но достаточно. Студент освоил основную базу теоретических знаний. Владеет терминологией и основными понятиями из профессиональной сферы.

Продвинутый «хорошо/экзамен» - компетенция освоена достаточно хорошо. Студент знает теоретический материал по дисциплине, умеет применить эти знания на практике. Чётко и ясно формулирует свои мысли. Знает специальную и публицистическую литературу по профессиональным вопросам.

Высокий «отлично/экзамен» - компетенция освоена в полной мере или на продвинутом уровне. Студент знает теоретический материал, умеет применить эти знания на практике и имеет опыт в профессионально-практической деятельности. Приводит актуальные примеры из сферы профессиональной деятельности; демонстрирует способности к нестандартной интерпретации поставленного вопроса.

Наименование темы (раздела)	**Показатели оценивания**	**Критерии оценивания**	**Оценка (баллы)**
Базовые понятия Java-технологий	Вопрос на экзамене, Р	Экзамен	Удовлетворительно -41 -60, жорошо 61-80, отлично 81-100, Не зачтено – 40-0
Преимущества Java Spring	Вопрос на экзамене	Экзамен	Удовлетворительно -41 -60, жорошо 61-80, отлично 81-100, Не зачтено – 40-0
Аспектно-ориентированное программирование	Вопрос на экзамене, КР	Экзамен	Удовлетворительно -41 -60, жорошо 61-80, отлично 81-100, Не зачтено – 40-0

В начале каждого семестра до студентов доводится информация о структуре набора баллов за семестр:

Максимальный балл за посещение и активную работу в семестре – 60 баллов.

Максимальный балл за каждую контрольную работу – 7 баллов.

Максимальный балл за каждую решенную задачу – 7 баллов.

Максимальный балл за реферат по выбранной теме – 12 баллов.

Вес текущих контрольных работ различен и зависит от этапов формирования компетенций. Решение и постановка прикладных задач, формирующих компетенции, выше, чем простое владение информационными технологиями.

В результате каждый студент четко представляет свое число набранных баллов.

Текущая аттестация обучаемых. Текущая аттестация обучаемых по дисциплине «Основы Java+ Фреймворк Spring» проводится в соответствии с Уставом Академии, Положением о текущей аттестации студентов по программам ВО и является обязательной.

Текущая аттестация по дисциплине «Основы Java+ Фреймворк Spring» проводится в форме опроса и контрольных мероприятий по оцениванию фактических результатов обучения студентов и осуществляется ведущим преподавателем.

Объектами оценивания выступают:

учебная дисциплина (активность на занятиях, своевременность выполнения различных видов заданий, посещаемость практических занятий по аттестуемой дисциплине);

уровень овладения практическими умениями и навыками по всем видам учебной работы, уровень освоения компетенций;

результаты самостоятельной работы.

Активность обучаемого на занятиях оценивается по скорости и правильности выполнения практических заданий на компьютерах.

Кроме того, оценивание обучаемого проводится на контрольной неделе в соответствии с распоряжением проректора по учебной работе. Оценивание обучаемого на контрольной неделе проводится преподавателем независимо от наличия или отсутствия обучаемого (по уважительной или неуважительной причине) на занятии. Оценка носит комплексный характер и учитывает достижения обучаемого по основным компонентам учебного процесса за текущий период.

Оценивание обучаемого на занятиях осуществляется с использованием балльно-рейтинговой системы в соответствии с технологической картой дисциплины. Оценивание обучаемого на контрольной неделе также осуществляется по балльно-рейтинговой системе с выставлением оценок в ведомости и указанием количества пропущенных занятий.

4.3. Оценочные средства для промежуточной аттестации

4.3.1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы. Показатели и критерии оценивания компетенций с учетом этапа их формирования

1.2.

Код компетенции	Наименование компетенции	Код этапа освоения компетенции	Наименование этапа освоения компетенции

ПКр – 1	Способен проводить обследование организаций, выявлять информационные потребности пользователей, формировать требования к информационной системе	ПКр-1.1 ПКр-1.2 ПКр-1.3	Знает принципы разработки планов коммуникаций с заказчиком в проектах создания (модификации) ИС Умеет организовывать заключение договоров на выполняемые работы, связанные с ИС Владеет навыками организации заключения договоров на выполняемые работы, связанные с ИС
ПКр-4	способность документировать процессы создания информационных систем на стадиях жизненного цикла	ПКр-4.1 ПКр-4.2. ПКр-4.3.	Знает принципы документирования существующих бизнес-процессов организации заказчика (реверс-инжиниринг бизнес-процессов организации) Умеет выполнять документирование существующих бизнес-процессов организации заказчика Владеет навыками создания пользовательской документации к ИС
УК ОС–2	Способность разработать проект на основе оценки ресурсов и ограничений	УК ОС–2.1 УК ОС–2.2	УК-2.1. Способен применять необходимые для осуществления профессиональной деятельности правовые нормы и методологические основы принятия управленческого решения. УК-2.2. Способен анализировать

		УК ОС–2.3	альтернативные варианты решений для достижения намеченных результатов; разрабатывать план, определять целевые этапы и основные направления работ. УК-2.3. Владеет методиками разработки цели и задач проекта; методами оценки продолжительности и стоимости проекта, а также потребности в ресурсах.

ОТФ/ТФ *(при наличии профстандарта)*/ трудовые или профессиональные действия	Код этапа освоения компетенции	Результаты обучения
06.015 СПЕЦИАЛИСТ ПО ИНФОРМАЦИОННЫМ СИСТЕМАМ С Выполнение работ и управление работами по созданию (модификации) и сопровождению ИС, автоматизирующих задачи организационного управления и бизнес-процессы С/03.6 Планирование коммуникаций с заказчиком в проектах создания (модификации) и ввода ИС в эксплуатацию С/42.6 Организация заключения договоров на выполняемые	ПКр-1.1 ПКр-1.2 ПКр-1.3	на уровне знаний: Знает принципы разработки планов коммуникаций с заказчиком в проектах создания (модификации) ИС Знает порядок подготовки и формирования документации для заключения договоров на выполняемые работы, связанные с ИС на уровне умений: Умеет организовывать заключение договоров на выполняемые работы, связанные с ИС на уровне навыков: Владеет навыками выстраивания коммуникаций с заказчиком в проектах создания (модификации) и ввода ИС в эксплуатацию Владеет навыками организации заключения договоров на выполняемые работы, связанные с ИС

работы, связанных с ИС		
06.015 СПЕЦИАЛИСТ ПО ИНФОРМАЦИОННЫМ СИСТЕМАМ С Выполнение работ и управление работами по созданию (модификации) и сопровождению ИС, автоматизирующих задачи организационного управления и бизнес-процессы С/07.6 Документирование существующих бизнес-процессов организации заказчика (реверс-инжиниринг бизнес-процессов организации) С/22.6 Создание пользовательской документации к ИС	ПКр-4.1 ПКр-4.2. ПКр-4.3.	ПКр-4.1.1 Знает правила и стандарты разработки пользовательской документации к ИС ПКр-4.1.2 Знает основные принципы документирования и реинжиниринга бизнес-процессов организации ПКр-4.2.1 Умеет выполнять документирование существующих бизнес-процессов организации заказчика ПКр-4.2.2 Умеет применять правила и стандарты разработки пользовательской документации к ИС ПКр-4.3.1 Владеет навыками создания пользовательской документации к ИС ПКр-4.3.2 Владеет навыками реинжиниринга бизнес-процессов организации

	УК ОС–2.1 УК ОС–2.2 УК ОС–2.3	на уровне знаний*:* необходимые для осуществления профессиональной деятельности правовые нормы и методологические основы принятия управленческого решения. Умеет анализировать альтернативные варианты решений для достижения намеченных результатов; разрабатывать план, определять целевые этапы и основные направления работ. на уровне навыков *(типовые действия выполняются автоматически, без воспроизведения алгоритма):* Владеет методиками разработки цели и задач проекта; методами оценки продолжительности и стоимости проекта, а также потребности в ресурсах.

Промежуточная аттестация обучаемых по дисциплине «Основы Java+ Фреймворк Spring» проводится в соответствии с Уставом Академии, Положением о промежуточной аттестации студентов по программам ВО. Промежуточная аттестация по дисциплине «Основы Java+ Фреймворк Spring» проводится в соответствии с Учебным планом в 6 семестре в форме экзамена. Обучаемые допускаются к экзамену по дисциплине в случае выполнения ими учебного плана по дисциплине: выполнения всех заданий и мероприятий, предусмотренных программой дисциплины по формам текущего контроля и выполненных рефератах. В случае наличия учебной задолженности обучаемый отрабатывает пропущенные занятия в соответствии с требованиями преподавателя. Экзамен и зачет принимает ведущий преподаватель.

Оценка знаний обучаемого на экзамене складывается из:

ответа на один вопрос экзамена;

выполнения всех текущих контрольных работ;

защиты рефератов по третьей теме;

Оценка знаний обучаемого на экзамене носит комплексный характер, является балльной и определяется его:

- ответом на экзамене;
- учебными достижениями в семестровый период.

Знания, умения, навыки обучаемого на зачете оцениваются оценками: «отлично» – 5, «хорошо» – 4, «удовлетворительно» – 3, «неудовлетворительно» – 2. Кроме того, обучаемому выставляется оценка в соответствии с балльно-рейтинговой системой. Основой для определения оценки служит уровень усвоения обучаемыми материала, предусмотренного данной рабочей программой.

4.3.2 Типовые оценочные средства

Темы коллоквиумов по дисциплине «Основы Java+ Фреймворк Spring»

Коллоквиум организуется для проверки самостоятельной работы по теме, предложенную преподавателем (обучаемым). Коллоквиум позволяет автору научиться четко и грамотно формулировать мысли, структурировать информацию, использовать основные категории анализа, выделять причинно-следственные связи, иллюстрировать понятия соответствующими примерами, аргументировать свои выводы; овладеть научным стилем речи. Коллоквиум содержит: четкое изложение сути поставленной проблемы, включать самостоятельно проведенный анализ этой проблемы с использованием концепций и аналитического инструментария, рассматриваемого в рамках дисциплины, выводы, обобщающие авторскую позицию по поставленной проблеме.

1. Установка BeanDefinition и парсирование конфигурации контекста (Xml, JavaConfig, Groovy и AnnotationConfigApplicationContext). На первоначальном этапе создаются все интерфейсы – BeanDefinition. В зависимости от выбора конфигурации необходимо использовать соответствующий механизм - парсирование компонентов приложения.
2. Настройка созданных интерфейсов (видов) – BeanDefinition.
3. Установка кастомных объектов с помощью FactoryBean.
4. Создание экземплярных (черновых) бинов.
5. Настройка созданных бинов. Проверка функциональности написанных бинов.
6. Контроллер – inversion of control. Обеспечивают корректную работоспособность, отвечают за конфигурацию элементов приложения и регулируют функциональный цикл объектов Java.
7. Доступ к данным – функционал направлен на управление реляционными базами данных (БД) в Java. Благодаря данному фреймворку обеспечивается решение огромного количества сложных задач на базе Java environments.
8. Фреймворк аспектно-ориентировочной настройки. Работает с компонентами, которые не могут реализоваться без каких-либо потерь на Java.
9. Управления транзакциями. Фреймворк координирует АР и инструментарий настроившими объектами управлениями транзакций для элементов Java.
10. Фреймворк удаленного доступа и управления. Конфигурация происходит локально.
11. Аутентификация и авторизация. Поддерживают конфигурацию на многих популярных протоколах через Spring Security.

Критерии оценки коллоквиумов:

Оценка «удовлетворительно» предполагает, что полученные результаты в значительной степени соответствуют поставленной цели (цель работы достигнута в основном). Обоснована актуальность работы. В процессе анализа литературы отобраны наиболее важные источники, продемонстрировано понимание решаемой проблемы. Выбраны адекватные цели научный подход, методы, процедуры. Они в значительной степени реализованы в работе. Выводы имеют наглядный и проверяемый характер. Требования по оформлению работы в основном выполнены.

Оценка «хорошо» ставится, когда полученные результаты преимущественносоответствуют поставленной цели и задачам. Обоснована практическая и теоретическая актуальность работы. В процессе анализа литературы отобран и проанализирован широкий круг теоретических и эмпирических источников. Выбраны и обоснованы применяемые научные подходы, методы и процедуры. Полученные результаты в целом логичны, доказательны и

систематизированы. Оформление работы в целом соответствует существующим требованиям.

Оценка «отлично» предполагает: полученные результаты полностью соответствуют поставленной цели. Обоснована практическая и теоретическая значимость работы. Проведен детальный анализ теоретических и эмпирических источников, выводы автора самостоятельны и аргументированы. Выбраны и подробно описаны применяемые в работе научные подходы, методы и процедуры. Содержание работы полностью отражает узловые проблемы темы, исследовательская часть (в курсовой работе) выполнена самостоятельно, методологически корректно и содержит достоверные и интересные выводы и положения. Оформление работы полностью отвечает всем требованиям

Вопросы к экзамену по дисциплине «Основы Java+ Фреймворк Spring»

1. Расскажите о Spring Framework.
2. Каковы некоторые из важных особенностей и преимуществ Spring Framework?
3. Что вы понимаете под Dependency Injection (DI)?
4. Как реализуется DI в Spring Framework?
5. Каковы преимущества использования Spring Tool Suite?
6. Приведите названия некоторых важных Spring модулей.
7. Что вы понимаете под аспектно-ориентированным программированием (Aspect Oriented Programming — AOP)?
8. Что такое Aspect, Advice, Pointcut, JointPoint и Advice Arguments в АОП?
9. В чем разница между Spring AOP и AspectJ АОП?
10. Что такое IoC контейнер Spring?
11. Что такое Spring бин?
12. Какое значение имеет конфигурационный файл Spring Bean?
13. Каковы различные способы настроить класс как Spring Bean?
14. Какие вы знаете различные scope у Spring Bean?
15. Что такое жизненный цикл Spring Bean?
16. Как получить объекты ServletContext и ServletConfig внутри Spring Bean?
17. Что такое связывание в Spring и расскажите об аннотации @Autowired?
18. Каковы различные типы автоматического связывания в Spring?
19. Является ли Spring бин потокобезопасным?
20. Что такое контроллер в Spring MVC?
21. Какая разница между аннотациями @Component, @Repository и @Service в Spring?
22. Расскажите, что вы знаете о DispatcherServlet и ContextLoaderListener.
23. Что такое ViewResolver в Spring?
24. Что такое MultipartResolver и когда его использовать?
25. Как обрабатывать исключения в Spring MVC Framework?
26. Как создать ApplicationContext в программе Java?
27. Можем ли мы иметь несколько файлов конфигурации Spring?
28. Каковы минимальные настройки, чтобы создать приложение Spring MVC?
29. Как бы вы связали Spring MVC Framework и архитектуру MVC?
30. Как добиться локализации в приложениях Spring MVC?
31. Как мы можем использовать Spring для создания веб-службы RESTful, возвращающей JSON?
32. Приведите пример часто используемых аннотаций Spring.
33. Можем ли мы послать объект как ответ метода обработчика контроллера?
34. Как загрузить файл в Spring MVC?
35. Как проверить (валидировать) данные формы в Spring Web MVC Framework?

36. Что вы знаете Spring MVC Interceptor и как он используется?
37. Spring JdbcTemplate класс и его применение.
38. Как использовать Tomcat JNDI DataSource в веб-приложении Spring?
39. Каким образом можно управлять транзакциями в Spring?
40. Расскажите о Spring DAO.
41. Как интегрировать Spring и Hibernate?
42. Расскажите о Spring Security.
43. Как внедрить java.util.Properties в Spring Bean?
44. Назовите некоторые из шаблонов проектирования, используемых в Spring Framework?

Критерии оценки компетенций

Промежуточная аттестация обучаемых по дисциплине «Основы Java+ Фреймворк Spring» проводится в соответствии с Уставом Академии, Положением о промежуточной аттестации студентов по программам ВПО. Промежуточная аттестация по дисциплине «Основы Java+ Фреймворк Spring» проводится в соответствии с Учебным планом в 6 семестре в форме экзамена. Обучаемые допускаются к экзамену по дисциплине в случае выполнения ими учебного плана по дисциплине: выполнения всех заданий и мероприятий, предусмотренных программой дисциплины по формам текущего контроля и выполненных заданий. В случае наличия учебной задолженности обучаемый отрабатывает пропущенные занятия в соответствии с требованиями преподавателя. Оценка знаний обучаемого на зачете носит комплексный характер, является балльной и определяется его:

- ответом на экзамене;
- учебными достижениями в семестровый период.

Знания, умения, навыки студента на экзамене оцениваются оценками: «отлично» – 5, «хорошо» – 4 , «удовлетворительно» – 3, «неудовлетворительно» – 2. Кроме того, обучаемому выставляется оценка в соответствии с балльно-рейтинговой системой. Основой для определения оценки служит уровень усвоения обучаемыми материала, предусмотренного данной рабочей программой.

Темы рефератов по дисциплине «Основы Java+ Фреймворк Spring»

1. Автоматическое управление ресурсами.
2. Обработка исключенных компонентов при доступе к данным исключения самого "Спринга".
3. Прозрачное управление транзакциями во время каких-либо операций с данными.
4. Разархивирование ресурсов.
5. Обработка абстракции во многих форматах
6. Поддержка конфигурации на базе HBase.
7. Профессиональная поддержка Batch Spring.
8. Поддержка пользования с интеграцией Java Spring.
9. Параметризация конфигурации Hadoop.
10. Устройства шаблонной обработки соединительных компонентов с Hadoop. Декларативная поддержка для инструментов Hadoop.
11. Программируемая поддержка для компонентов Hadoop.

Критерии оценки рефератов:

Оценка «удовлетворительно» предполагает, что полученные результаты в значительной степени соответствуют поставленной цели (цель работы достигнута в основном). Обоснована актуальность работы. В процессе анализа литературы отобраны наиболее важные источники, продемонстрировано понимание решаемой

проблемы. Выбраны адекватные цели научный подход, методы, процедуры. Они в значительной степени реализованы в работе. Выводы имеют наглядный и проверяемый характер. Требования по оформлению работы в основном выполнены.

Оценка «хорошо» ставится, когда полученные результаты преимущественносоответствуют поставленной цели и задачам. Обоснована практическая и теоретическая актуальность работы. В процессе анализа литературы отобран и проанализирован широкий круг теоретических и эмпирических источников. Выбраны и обоснованы применяемые научные подходы, методы и процедуры. Полученные результаты в целом логичны, доказательны и систематизированы. Оформление работы в целом соответствует существующим требованиям.

Оценка «отлично» предполагает: полученные результаты полностью соответствуют поставленной цели. Обоснована практическая и теоретическая значимость работы. Проведен детальный анализ теоретических и эмпирических источников, выводы автора самостоятельны и аргументированы. Выбраны и подробно описаны применяемые в работе научные подходы, методы и процедуры. Содержание работы полностью отражает узловые проблемы темы, исследовательская часть (в курсовой работе) выполнена самостоятельно, методологически корректно и содержит достоверные и интересные выводы и положения. Оформление работы полностью отвечает всем требованиям

Шкала оценивания по результатам зачета с оценкой:

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	Критерии оценивания результатов обучения			
	неудовлетворительно	удовлетворительно	хорошо	отлично
ПКр – 1. Способен выполнять работы и управлять работами по созданию (модификации) и сопровождению информационных систем, автоматизирующих задачи организационного управления и бизнес-процессы				
на уровне знаний: принципов разработки планов коммуникаций с заказчиком в проектах создания (модификации) ИС; Знает принципы, методы и стандарты разработки	Не знает как проводить описание прикладных процессов и информационного обеспечения решения разработки планов коммуникаций с заказчиком в проектах создания (модификации) ИС;	Демонстрирует частичные знания в описании прикладных процессов и информационного обеспечения решения разработки планов коммуникаций с заказчиком в проектах создания (модификации)	Демонстрирует системные знания в описании прикладных процессов и информационного обеспечения решения разработки планов коммуникаций с заказчиком в проектах создания (модификации)	Проявляет высокий уровень знаний всего учебного материала

архитектуры ИС; Знает основные методологии и средства проектирования и дизайна ИС; Знает основные методологии и средства разработки баз данных ИС; Знает правила и стандарты разработки пользовательской документации к ИС; Знает принципы и приемы командообразования.	методологии и средства проектирования и дизайна ИС; правила и стандарты разработки пользовательской документации к ИС; принципы и приемы командообразования.	ИС; методологии и средства проектирования и дизайна ИС; правила и стандарты разработки пользовательской документации к ИС; принципы и приемы командообразования.	ИС; методологии и средства проектирования и дизайна ИС; правила и стандарты разработки пользовательской документации к ИС; принципы и приемы командообразования.	
на уровне умений: Умеет выполнять документирование существующих бизнес-процессов организации заказчика; Умеет применять принципы, методы и стандарты разработки архитектуры ИС; Умеет профессионально выполнять разработку баз данных ИС;	Не умеет применять знания на практике	Демонстрирует частичный уровень умений	Демонстрирует достаточный уровень умений	Демонстрирует высокий уровень умений для разработки алгоритмов решения задач

Умеет организовывать программы обучения и развития персонала.				
на уровне навыков: Владеет навыками выстраивания коммуникаций с заказчиком в проектах создания (модификации) и ввода ИС в эксплуатацию; Владеет навыками документирования и реинжиниринга бизнес-процессов организации; Владеет навыками проектирования и разработки дизайна ИС; Владеет навыками создания пользовательской документации к ИС; Владеет навыками организации заключения договоров на выполняемые работы, связанные с ИС	Не владеет практическими навыками работы с технологиями и программным инструментарие м	Владеет практическими навыками частично работы с технологиями и программным инструментарием	Владеет практическим и навыками формирования требований к информационной системе достаточно свободно	Владеет практическим и навыками работы на компьютере и использования стандартных информационно-коммуникационных технологий на высоком уровне
УК ОС-2 Способность разработать проект на основе оценки ресурсов и ограничений				

на уровне знаний: Знает необходимые для осуществления профессиональной деятельности правовые нормы и методологические основы принятия управленческого решения. Знает методы анализа альтернативных решений для достижения намеченных результатов Знает методики разработки цели и задач проекта	Не знает как проводить описание прикладных процессов и информационного обеспечения решения разработки планов коммуникаций с заказчиком в проектах создания (модификации) ИС; методологии и средства проектирования и дизайна ИС; правила и стандарты разработки пользовательской документации к ИС; принципы и приемы командообразования.	Демонстрирует частичные знания в описании прикладных процессов и информационного обеспечения решения разработки планов коммуникаций с заказчиком в проектах создания (модификации) ИС; методологии и средства проектирования и дизайна ИС; правила и стандарты разработки пользовательской документации к ИС; принципы и приемы командообразования.	Демонстрирует системные знания в описании прикладных процессов и информационного обеспечения решения разработки планов коммуникаций с заказчиком в проектах создания (модификации) ИС; методологии и средства проектирования и дизайна ИС; правила и стандарты разработки пользовательской документации к ИС; принципы и приемы командообразования.	Проявляет высокий уровень знаний всего учебного материала
на уровне умений: Умеет использовать методологические основы принятия управленческого решения в профессиональной области Умеет разрабатывать план, определять целевые этапы и	Не умеет применять знания на практике	Демонстрирует частичный уровень умений	Демонстрирует достаточный уровень умений	Демонстрирует высокий уровень умений для разработки алгоритмов решения задач

основные направления работ				
на уровне навыков: Владеет методами оценки продолжительности и стоимости проекта, а также потребности в ресурсах.	Не владеет практическими навыками работы с технологиями и программным инструментарием	Владеет практическими навыками частично работы с технологиями и программным инструментарием	Владеет практическим и навыками формирования требований к информационной системе достаточно свободно	Владеет практическим и навыками работы на компьютере и использования стандартных информационно-коммуникационных технологий на высоком уровне

4.4. Методические материалы

4.4.1. Методические материалы, определяющие процедуру оценивания ответов обучающихся на вопросы на понимание лекционного материала

Критериями оценки ответа обучающихся на лекционном занятии выступают:

- правильность ответов на вопросы преподавателя по изученному
- материалу;
- полнота и лаконичность ответа;
- степень понимания тематики предмета;
- логика и аргументированность изложения материала;
- приведение примеров, демонстрирующих умение и владение
- полученными знаниями по темам предмета в раскрытии поставленных вопросов.

4.4.2. Методические материалы, определяющие процедуру оценивания при проведении опроса на практическом занятии

Оценки "отлично" заслуживает обучающийся, обнаруживший всестороннее, систематическое и глубокое знание вопроса, умение свободно ориентироваться в теме, усвоивший основную, и знакомый с дополнительной литературой, рекомендованной программой. Как правило, оценка "отлично" выставляется обучающимся, усвоившим взаимосвязь основных понятий в их значении для приобретаемой профессии, проявившим творческие способности в понимании, изложении и использовании учебно-программного материала;

Оценки "хорошо" заслуживает обучающийся, обнаруживший полное знание темы, успешно выполняющий предусмотренные программой задания, усвоивший основную литературу, рекомендованную в программе. Как правило, оценка

"хорошо" выставляется обучающимся, показавшим систематический характер знаний по пройденному материалу и способным к их самостоятельному пополнению и обновлению в ходе дальнейшей учебы и профессиональной деятельности;
Оценки "удовлетворительно" заслуживает обучающийся, обнаруживший знание основного материала в объеме, необходимом для дальнейшего усвоения материала и предстоящей работы по профессии, знакомый с основной литературой, рекомендованной программой.
Оценка "неудовлетворительно" выставляется обучающемуся, обнаружившему пробелы в знаниях основного материала темы, допустившему принципиальные ошибки в понимании и изложении учебного материала.

4.4.3. Методические материалы, определяющие процедуру оценивания промежуточной аттестации по дисциплине

Экзамен проводится путем распределения по времени в течение всего срока обучения проведения мониторинга уровня освоения компетенции по результатам выполнения заданий по подготовке к опросу. При условии положительных ответов на вопросы экзамена, компетенции считаются сформированными, и по данной дисциплине выставляется оценка. В ином случае, компетенции считаются несформированными, и по данной дисциплине выставляется оценка «неудовлетворительно».
Экзамен принимается в устной форме, по вопросам. Задание включает теоретический и практический вопрос. Оценка знаний обучающегося на экзамене носит комплексный характер и определяется его:

- ответом на экзамене;
- учебными достижениями в семестровый период.

Знания, умения, навыки обучающегося на экзамене оцениваются оценками: «отлично», «хорошо», «удовлетворительно», «неудовлетворительно». Основой для определения оценки служит уровень усвоения обучающимися материала, предусмотренного данной рабочей программой.

Оценивание студента на зачете с оценкой по дисциплине «Основы Java+ Фреймворк Spring»

Оценка	Требования к знаниям
Отлично	Оценка «отлично» выставляется обучающемуся, если он глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически стройно его излагает на экзамене, умеет тесно увязывать теорию с практикой, свободно справляется с задачами, вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, правильно обосновывает принятое решение. Учебные достижения в семестровый период демонстрируют высокую степень овладения программным материалом.
Хорошо	Оценка «хорошо» выставляется обучающемуся, если он твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении практических вопросов и задач, владеет необходимыми навыками и приемами их выполнения. Учебные достижения в семестровый период демонстрируют хорошую степень овладения программным материалом.

Удовлетворительно	Оценка «удовлетворительно» выставляется обучающемуся, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала, испытывает затруднения при выполнении практических работ. Учебные достижения в семестровый период демонстрируют достаточную (удовлетворительную) степень овладения программным материалом.
Неудовлетворительно	Оценка «неудовлетворительно» выставляется обучающемуся, который не знает значительной части программного материала, допускает существенные ошибки, неуверенно, с большими затруднениями выполняет практические работы. Как правило, оценка «неудовлетворительно» ставится обучающимся, которые не могут продолжить обучение без дополнительных занятий по соответствующей дисциплине. Учебные достижения в семестровый период не демонстрировали достаточную степень овладения программным материалом на пороговом уровне.

Экзаменпроводится во время экзаменационной сессии по традиционной шкале оценивания.
Студентам, не выполнившим требования по промежуточной аттестации, по уважительным причинам, предоставляется возможность их выполнения и сдачи.

5. Методические указания для обучающихся по освоению дисциплины

Домашние задания соответствуют приведенным ранее темам для коллоквиумов и опросов при самостоятельной подготовке, и могут быть опубликованы:
1. На общих ресурсах академии (в сетевой папке);
2. Высланы на общую почту группы
Конкретный способ коммуникации со студентами определяется преподавателем. Для подготовки к практическим занятиям необходимо ознакомиться с лекциями, рекомендованной литературой, иными источниками, интернет ресурсами, повторить материал предыдущих практических занятий. Самостоятельно разобрать вопросы к практическим занятиям по темам. При возникновении вопросов - обратиться к преподавателю по электронной почте с указанием конкретной проблемы и (или) прийти к преподавателю на консультацию в установленное время.

5.1. Методические указания по вопросам на понимание лекционного материала

На лекциях рекомендуется слушать предлагаемый лектором материал, при этом параллельно конспектировать основные положения, поскольку это дает наибольший результат в усвоении материала. Предоставляется возможность задавать вопросы на уточнение понимания темы и принимать участие в ее обсуждении. Кроме этого, для лучшего освоения материала и систематизации знаний по дисциплине, необходимо постоянно разбирать материалы лекций по конспектам и учебным пособиям. Во время самостоятельной проработки лекционного материала особое внимание следует уделять возникшим вопросам, непонятным терминам, спорным точкам зрения. Все такие моменты следует выделить или выписать отдельно для дальнейшего обсуждения на семинарском занятии. В случае необходимости обращаться к преподавателю за консультацией. Полный список литературы по дисциплине приведен в разделе 6 программы.

5.2. Методические указания по подготовке вопросов для самостоятельного изучения к занятиям практического (семинарского) типов

Подготовка обучающегося к практическому занятию осуществляется на основании плана раскрытия темы практического занятия, которое разрабатывается преподавателем на основе рабочей программы и своевременно доводится до сведения обучающегося.

При подготовке к практическому занятию, обучающемуся необходимо внимательно изучить основные вопросы темы семинара. Важным условием успешной подготовки к практическому занятию является четкая организация самостоятельной работы студентов по изучению учебной и дополнительной литературы. Умение анализировать и применять для ответов на вопросы и решения задач и заданий полученные знания при самостоятельной подготовке в значительной степени определяет успешность освоения материала по дисциплине и формирование у обучающихся соответствующих компетенций.

Подготовка вопросов для самостоятельного изучения включает: изучение необходимой литературы (обязательной, дополнительной литературы, специальных периодических изданий, Интернет-ресурсов), подготовку конспекта ответа, ответы на вопросы.

При подготовке к практическим занятиям важно:

- использовать достаточно широкий диапазон массива информации, провести обзор литературы и специальных изданий, составить каталог Интернет-ресурсов;
- представить различные подходы, четко и полно определить рассматриваемые понятия, выявить взаимосвязи понятий и явлений, взаимозависимости и связи с другими вопросами;
- грамотно структурировать материал, ясно, четко и логично его излагать, приводить соответствующие примеры из практики, для иллюстрации положений, тезисов и выводов использовать таблицы, схемы, графики, диаграммы.

Вопросы для самостоятельной подготовки к занятиям практического (семинарского) типа указаны в разделе 4.2.

5.3. Методические рекомендации по подготовке к экзамену по дисциплине

Ответ на экзамене предусматривает устный ответ на теоретические вопросы. При подготовке к экзамену обучающийся обращается к пройденному материалу, сосредоточенному в конспектах лекций, учебниках и других источниках информации.

Повторяя, обобщая, закрепляя и дополняя полученные знания, поднимает их на качественно-новый уровень — уровень системы совокупных данных, что позволяет ему понять логику всего предмета в целом. Новые знания обучающийся получает в ходе самостоятельного изучения того, что не было изложено в лекциях и на семинарских занятиях.

Экзаменкак особая форма учебного процесса имеет свои особенности, специфические черты и некоторые аспекты, которые необходимо обучающемуся знать и учитывать в своей работе. Это, прежде всего:

• что и как запоминать при подготовке к зачету;
• по каким источникам и как готовиться;
• на чем сосредоточить основное внимание;
• каким образом в максимальной степени использовать программу курса;
• что и как записать, а что выучить дословно и т. п.

На экзамене, как правило, проверяется не столько уровень запоминания обучающимся учебного материала, сколько то, насколько успешно он оперирует

теми или иными научными понятиями и категориями, систематизирует факты, как умеет мыслить, аргументировано отстаивать определенную позицию, объясняет и пересказывает заученную информацию.
Программу курса необходимо максимально использовать как в ходе подготовки, так и на самом экзамене. Ведь она включает в себя разделы, темы и основные проблемы, в рамках которых и формируются вопросы для экзамена.
Оптимальным для подготовки к экзамену является вариант, когда обучающийся начинает подготовку к нему с первых занятий по данному курсу.
При подготовке к экзамену по наиболее сложным вопросам, ключевым проблемам и важнейшим понятиям необходимо сделать краткие письменные записи в виде тезисов, планов, определений. Особое внимание в ходе подготовки к экзамену следует уделять конспектам лекций, ибо они обладают рядом преимуществ по сравнению с печатной продукцией. Как правило, они более детальные, иллюстрированные, что позволяет оценивать современную ситуацию, отражать самую свежую научную и оперативную информацию, отвечать на вопросы, интересующие аудиторию, в данный момент, тогда как при написании и опубликовании печатной продукции проходит определенное время, и материал быстро устаревает.
В то же время, подготовка по одним конспектам лекций недостаточна, необходимо использовать и иную учебную литературу. Не следует бояться дополнительных и уточняющих вопросов на экзамене. Они, как правило, задаются или помимо экзаменационного вопроса для выявления общей подготовленности, или в рамках билета для уточнения высказанной мысли.

6. Учебная литература и ресурсы информационно-телекоммуникационной сети "Интернет", включая перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

6.1. Основная литература

1. Юлиана Козмина, Роб Харроп, Крис Шефер, Кларенс Хо. Spring 5 для профессионалов- М.: СПб: Диалектика-Вильямс, 2019. - 608 с.
2. Герберт Java 2 v5.0 (Tiger). Новые возможности / Герберт, Шилдт. - М.: СПб: БХВ-Петербург, 20018. - 208 с.
3. Даконта, М. XML и Java 2 / М. Даконта, А. Саганич. - М.: СПб: Питер, 2019. - 384 с.
4. Мугал, Халид А. Java. Руководство по подготовке к сдаче сертификационного экзамена CX-310-035 (+ CD-ROM) / Мугал, Халид А., Расмуссен, Рольф В.. - М.: КУДИЦ-Образ, 2018. - 688 с.

6.2. Дополнительная литература

5. Нотон Java. Справочное руководство. Все, что необходимо для программирования на Java / Нотон, Патрик. - М.: Бином, 1996. - 448 с.
6. Ноултон, П. Java 2 в подлиннике / П. Ноултон, Г. Шилдт. - М.: СПб: BHV, 2001. - **526** с.
7. Программирование на языке Java. - М.: Мультимедиа Технологии и Дистанционное Обучение, 2004. - 416 с.

6.3. Учебно-методическое обеспечение самостоятельной работы

8. Положение об организации самостоятельной работы студентов федерального государственного бюджетного образовательного учреждения

высшего образования «Российская академия народного хозяйства и государственной службы при Президенте Российской Федерации» (в ред. приказа РАНХиГС от 11.05.2016 г. № 01-2211). http://www.ranepa.ru/images/docs/prikazy-ranhigs/Pologenie_o_samostoyatelnoi_rabote.pdf

Самостоятельная работа студентом осуществляется для закрепления изученного материала после практических занятий или лабораторных работ, для выполнения домашних заданий, для подготовки к контрольным работам, для изучения дополнительных материалов.

№ пп	Тип занятия	Указания
	Тема 1	Базовые понятия Java-технологий
1	СРС	Посмотреть в интернете понятие и особенности информации, данных, знаний; клиент-серверных технологий, определить, в чем различие этих понятий. Обратить внимание на протокол HTTP.
2	СРС	Выяснить роль технологий веб в системах организационно-экономического управления. Рассмотреть понятие Основы JavaScripta. Изучить уровни представления информационных объектов веб при описании предметной области.
3	СРС	Для подготовки к занятиям использовать лекции, интернет ресурсы, основную и дополнительную литературу.
	Тема 2	Преимущества Java Spring
1	СРС	Изучить аспекты рассмотрения информационных процессов веб в системах управления: преобразование и движение. Процесс принятия решения как основной элемент преобразования информации. Знать основные фазы процесса принятия решений как информационного процесса. Подробно разобрать процессы передачи информации. Знать определение и свойства потока информации веб.
2	СРС	Знать уровни представления информационных процессов. Подробно изучить концептуальную модель информационного процесса. Разобрать структуры логического и физического уровней представления информационного процесса веб.
3	СРС	Для подготовки к занятиям использовать лекции, интернет ресурсы, основную и дополнительную литературу.
	Тема 3	Аспектно-ориентированное программирование
1	СРС	Разобрать понятие инструментария веб. Изучить место и роль интернет-приложений в общем контуре организационно-экономического управления. Подробно разобрать этапы развития веб. Выделить основные процессы, обеспечивающие работу веб приложений. Изучить основные свойства сайта. Знать преимущества внедрения веб в сферу управления и бизнеса.
2	СРС	Разобрать понятие Основы JavaScripta. Выделить и изучить подходы к разработке веб сайтов.

3	СРС	Для подготовки к занятиям использовать лекции, интернет ресурсы, основную и дополнительную литературу

Рекомендации по организации самостоятельной работы студентов:

Вид учебных занятий	Организация деятельности 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. Обозначить вопросы, термины, материал, который вызывает трудности, пометить и попытаться найти ответ в рекомендуемой литературе. Если самостоятельно не удается разобраться в материале, необходимо сформулировать вопрос и задать преподавателю на консультации, на практическом занятии.
Практические занятия	Проработка рабочей программы, уделяя особое внимание целям и задачам, структуре и содержанию дисциплины. Анализ литературы и/или Интернет – ресурсов, решение типовых задач, проведение консультаций с преподавателем по проблемным вопросам лекционного материала, подготовка к опросу и/или коллоквиуму, постановка задачи для практического задания, оформление отчета по заданию, работа в малой группе.
Контрольная работа / индивидуальное задание	Ознакомление с постановкой задачи, разработка алгоритма решения, выбор программно-аппаратных средств реализации. Реализация в выбранной инструментальной среде. Подготовка выводов и/или короткого отчета в виде презентации.
Подготовка к дискуссиям, докладам, написанию реферата	Поиск литературы и составление библиографии, использование от 3 до 5 научных работ, изложение мнения авторов и своего суждения по выбранному вопросу; изложение основных аспектов проблемы. Изучение научной, учебной, нормативной и другой литературы. Отбор необходимого материала; формирование выводов и разработка конкретных алгоритмов и/или рекомендаций по решению поставленной цели и задачи; проведение практических исследований по данной теме. Разработка презентации для доклада основных результатов проведенной работы. Желательно оформление результатов в виде статьи в сборник трудов конференции или тематический научный журнал.

6.4. Нормативные правовые документы

9. IEEE P1003.0 «Руководство по окружению открытых систем POSIX».
10. ГОСТ Р ИСО/МЭК 12207. Процессы жизненного цикла программных средств.

6.5. Интернет-ресурсы, справочные системы.

11. http://smarty.php.net – веб-технологии
12. http://yandex.ru/cgi/vesna - примеры скриптов веб-приложений
13. http://belitsoft.ru/ - инструментарий разработки интернет-приложений
14. http://www.virtech.ru – инструментарий разработки интернет-приложений
15. http://www.oracle.sun.com – инструментарий разработки интернет-приложений
16. 1. http://www.consultant.ru/ - Официальный сайт компании «Консультант Плюс».
17. http://www.cfin.ru/software/index.shtml - Сайт «Корпоративный менеджмент». Обзор информационных систем для основных функций управления и бизнеса.
18. http://citforum.ru/ - «Сервер информационных технологий» - on-line библиотека информационных материалов по компьютерным технологиям.
19. http://www.intuit.ru/ - Образовательный портал дистанционного обучения.
20. www.coursera.org - Платформа для бесплатных онлайн - лекций (проект по публикации образовательных материалов в интернете, в виде набора бесплатных онлайн - курсов).

6.6. Иные источники

21. Рихтер CLR via C#. Программирование на платформе Microsoft .NET Framework 2.0 на языке C# / Рихтер, Джефри. - М.: Питер, 2007. - 656 с.
22. Рихтер Программирование на платформе Microsoft. NET Framework / Рихтер, Джеффри. - М.: Русская Редакция, 2002. - 512 с.
23. Рихтер, Дж. Программирование на платформе .NET Framework / Дж. Рихтер. - М.: Microsoft Press. Русская Редакция; Издание 2-е, испр., 2003. - 512 с.

7. Материально-техническое и программное обеспечение дисциплины (модуля)

Название лаборатории/класса, оснащенного необходимым, в соответствии с требованиями ФГОС/ СУОС, оборудованием	**Наименование оборудования**	**Перечень лицензионного программного обеспечения**
учебная аудитория для проведения занятий лекционного типа, текущего контроля и промежуточной аттестации	Рабочие места студентов: столы и стулья – соответственно количеству студентов. Рабочее место для инвалида и лиц с ОВЗ: парта с телескопической столешницей на электромеханическом приводе - 1 шт., кресло-коляска для инвалидов 18" - 1 шт., индукционная петля - 1 шт., компьютер с версией для слабовидящих - 1 шт., кнопка вызова сотрудников – 1 шт. Рабочее место преподавателя: стол – 1	Мультимидийный проектор Консультант Плюс

	шт., стул – 1 шт, кафедра - 1 шт. Доска меловая и маркерная. Экран, ноутбук Lenovo ideapad 100/15, проектор	
информационно – аналитическая лаборатория – учебная аудитория для проведения занятий семинарского типа, курсового проектирования (выполнения курсовых работ), групповых и индивидуальны х консультаций	Рабочие места: столы компьютерные – в соотвествии с количеством студентов, кресло Престиж Profi -B-20 Самба бордо в рубчик - 15 шт., подставка для ног Fellowes FS-48121 Standard черный - 15 шт. Рабочее место преподавателя: стол компьютерный - 1 шт., стул - 1 шт. Доска меловая или маркерная Персональные компьютеры	Мультимидийный проектор Консультант Плюс
библиотека - помещение для самостоятельной работы	Рабочие места: столы и стулья. Рабочее место преподавателя: стол – 1 шт., стул - 1 шт., кафедра библиотечная - 1 шт. Телефон – 1 шт., цифровой многофункциональный копир - 1 шт., копировальный аппарат МФУ – 1 шт., принтер - 1 шт., сканер – 1 шт. Шкаф – 7 шт, стеллаж-33 шт, библиотечная стойка – 2 шт., стенд – 2 шт. Меловая или маркерная доска. Персональные компьютеры	Мультимидийный проектор Консультант Плюс

Программное обеспечение:

В процессе лекционных и семинарских занятий используется следующее программное обеспечение:

- программы, обеспечивающие навигацию в сети Интернет: «Google chrome»;
- программы, демонстрации видео материалов: проигрыватель «Windows Media»;
- программы для демонстрации и создания презентаций: «Microsoft Power Point».

Информационные справочные системы:

1. www.nnir.ru / - Российская национальная библиотека
2. www.nns.ru / -Национальная электронная библиотека
3. www.rsi.ru / - Российская государственная библиотека
4. www.biznes-karta.ru / -Агентство деловой информации «Бизнес-карта»
5. www. rbs.ru / - Информационное агентство «РосБизнесКонсалтинг»
6. www.aport.ru / - Поисковая система
7. www.rambler.ru / - Поисковая система
8. www.yandex.ru / - Поисковая система
9. www.busineslearning.ru / - Система дистанционного бизнес образования
10. www.test.specialist.ru / - Центр компьютерного обучения МГТУ им. Н. Э. Баумана
11. http://www.consultant.ru/ - Консультант плюс
12. http://www.garant.ru/ - Гарант
13. Научная библиотека РАНХиГС. URL: http://lib.ranepa.ru/;

14. Научная электронная библиотека eLibrary.ru. URL: http://elibrary.ru/defaultx.asp;
15. Национальная электронная библиотека. URL: www.nns.ru; Российская государственная библиотека. URL: www.rsl.ru;
16. Российская национальная библиотека. URL: www.nnir.ru; Электронная библиотека Grebennikon. URL: http://grebennikon.ru/;
17. Электронно-библиотечная система Издательства «Лань». URL: http://e.lanbook.com;
18. Электронно-библиотечная система ЮРАЙТ. URL: http://www.biblio-online.ru/.