

**Федеральное государственное автономное образовательное
учреждение высшего образования
«Московский физико-технический институт
(национальный исследовательский университет)»**

УТВЕРЖДЕНО

**Директор физтех-школы
радиотехники и компьютерных
технологий**

А.В. Дворкович

	Рабочая программа дисциплины (модуля)
по дисциплине:	Корпоративные информационные системы
по направлению:	Прикладные математика и физика
профиль подготовки:	Радиотехника и компьютерные технологии Физтех-школа Радиотехники и Компьютерных Технологий кафедра инфокоммуникационных систем и сетей
курс:	3
квалификация:	бакалавр

Семестр, формы промежуточной аттестации: 6 (весенний) - Дифференцированный зачет

Аудиторных часов: 30 всего, в том числе:

лекции: 0 час.

семинары: 30 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 15 час.

Всего часов: 45, всего зач. ед.: 1

Программу составил: А.Б. Корчак, канд. физ.-мат. наук

Программа обсуждена на заседании кафедры инфокоммуникационных систем и сетей 09.04.2020

Аннотация

Курс дает представление о современных технологиях разработки корпоративных информационных систем и навыки их проектирования. Курс включает в себя разработку Enterprise-приложения, работающего поверх Oracle/PostgreSQL баз данных с применением языков SQL/PL-SQL, а также построенного на микросервистной архитектуре и современных технологиях как BackEnd, так и FrontEnd. Отдельное внимание уделяется процессам разработки и принципам командной работы.

Курс разделен на два направления – системная/бизнес аналитика и Java разработка.

1. Цели и задачи

Цель дисциплины

Получение технологических знаний и навыков, необходимых для разработки, внедрения и/или поддержки корпоративных информационных систем.

Задачи дисциплины

- Получение представлений о проектировании баз данных (далее БД) и навыков написания SQL запросов (на примере Oracle SQL), необходимых для разработки, внедрения и поддержки корпоративных информационных систем.
- Получение навыков разработки информационных систем (на Java) и изучение связанных с этим принципов командной работы и технологий Java Enterprise Edition

2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.1 Анализирует задачу, выделяя этапы ее решения, действия по решению задачи
	УК-1.2 Находит, критически анализирует и выбирает информацию, необходимую для решения поставленной задачи
	УК-1.3 Рассматривает различные варианты решения задачи, оценивает их преимущества и недостатки
	УК-1.4 Грамотно, логично, аргументированно формирует собственные суждения и оценки
	УК-1.5 Определяет и оценивает практические последствия возможных вариантов решения задачи
ОПК-1 Способен применять фундаментальные знания, полученные в области физико-математических и (или) естественных наук, и использовать их в профессиональной деятельности	ОПК-1.1 Способен анализировать поставленную задачу, намечать пути ее решения
	ОПК-1.2 Способен строить математические модели, производить количественные расчеты и оценки
	ОПК-1.3 Способен определять границы применимости полученных результатов
ОПК-4 Способен осуществлять сбор и обработку научно-технической и (или) технологической информации для решения фундаментальных и прикладных задач	ОПК-4.1 Владеет методами научного поиска и интеллектуального анализа информации при решении задач профессиональной деятельности
	ОПК-4.4 Владеет навыками работы с компьютером и компьютерными сетями с целью получения, хранения и обработки научной (технической, технологической) информации
ПК-4 Способен критически оценивать применимость используемых методик и методов	ПК-4.1 Знает численные порядки величин, характерных для соответствующей профессиональной области
	ПК-4.3 Способен обосновать причинно-следственные отношения используемых понятий и моделей

3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны знать:

- синтаксис языка Java, принципы объектно-ориентированного программирования и особенности их реализации в языке Java;
- некоторые шаблоны объектно-ориентированного проектирования, применимые в веб-приложениях и корпоративных системах;
- многоуровневую архитектуру КИС, модель Java Enterprise Edition;
- основные сведения о серверах приложений, IoC, CDI;
- структуру библиотеки Java EE, назначение ее ключевых компонентов;
- основные понятия реляционных баз данных и базовые принципы их проектирования.

уметь:

- читать и рисовать ER-диаграммы для структуры БД;
- проектировать несложную БД из нескольких сущностей;
- использовать рассматриваемые библиотеки Java EE (пакеты javax.*) для написания реальных программ, библиотеку log4j – для эффективного логирования отладочной информации, библиотеку JUnit – для написания модульных тестов программ;
- писать несложный код на других языках кроме Java, необходимый для хранения данных в реальных Java-программах (XML, SQL, JPAQL) и для построения пользовательских веб-интерфейсов к серверным Java-приложениям (HTML, JSF);
- эффективно и по назначению использовать интегрированные среды разработки (Java IDE), репозиторий коллективной работы (Subversion), средства сборки (Maven).

владеть:

- навыками освоения большого объема информации в области программирования;
- навыками самостоятельного и быстрого поиска недостающей информации в Интернете;
- навыками отладки приложений с использованием дебаггера в Java IDE, логирования, а также правильной обработки исключительных ситуаций в коде;
- культурой программирования: умением писать код, читабельный с точки зрения стиля, имен переменных и методов, комментариев и др.; навыком проектирования модульных тестов к коду;
- навыками командной разработки программ (в т.ч. навыком версионирования кода), навыками краткого документирования программ (в т.ч. с применением языка UML).
- навыками отладки SQL запросов.

4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

4.1. Разделы дисциплины (модуля) и трудоемкости по видам учебных занятий

№	Тема (раздел) дисциплины	Трудоемкость по видам учебных занятий, включая самостоятельную работу, час.			
		Лекции	Семинары	Лаборат. работы	Самост. работа
1	Web приложения в Java EE		8		3
2	Проектирование баз данных		7		3
3	Enterprise Java Beans		5		3
4	Практические аспекты разработки ИС		5		3
5	Продвинутые подходы к разработке КИС		5		3
Итого часов			30		15
Подготовка к экзамену		0 час.			
Общая трудоёмкость		45 час., 1 зач.ед.			

4.2. Содержание дисциплины (модуля), структурированное по темам (разделам)

Семестр: 6 (Весенний)

1. Web приложения в Java EE

Введение в Java EE. Многоуровневая архитектура
CDI
Web приложения. Web сервер, JSP
JSF
Современные Web-технологии

2. Проектирование баз данных

Введение в реляционные БД
Проектирование реляционных БД
Адаптивные объектные модели

3. Enterprise Java Beans

Введение в Enterprise Java Beans
JPA в Java EE
Дополнительные возможности EJB

4. Практические аспекты разработки ИС

Инструментарий коллективной разработки
Шаблоны проектирования
Разбор полнофункционального Java EE приложения

5. Продвинутые подходы к разработке КИС

PL/SQL
Интеграция приложений. Web сервисы. JMS. MDB.
Транзакции. Безопасность

5. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Компьютерный класс, оборудованный мультимедийным проектором.

6. Перечень рекомендуемой литературы

Основная литература

1. СУБД : Язык SQL в примерах и задачах [Текст] : учеб. пособие для вузов / И. Ф. Астахова [и др.] .— М. : Физматлит, 2009 .— 168 с.
2. Гупта А. Java EE 7. Основы. – Диалектика / Вильямс, 2014 – 336 с.
3. Туманов В.Е. Основы проектирования реляционных баз данных: учебное пособие. – ИНТУИТ; БИНОМ. Лаборатория знаний, 2007 – 424 с.

Дополнительная литература

1. Библиотека профессионала. Java 2 [Текст] : Т. 2. Тонкости программирования/К. С. Хорстманн, Г. Корнелл , [пер. с англ. Я. П. Волковой, Д. Я. Ивченко, под ред. Ю. Н. Артеменко], Java 2. Vol. 2 : Advanced Features, -М., Вильямс, 2009
2. Библиотека профессионала. Java 2 [Текст] : Т. 1. Основы/К. С. Хорстманн, Г. Корнелл , Java 2. Vol. 1 : Fundamentals, [пер. с англ. Н. А. Мухина], -М., И. Д. Вильямс, 2013
3. Хеффельфингер Д. Java EE 6 и сервер приложений GlassFish 3. – ДМК Пресс, 2013 – 416 с.

7. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)

1. <http://learn.ncedu.ru/> – система дистанционного обучения (LMS) с материалами для лекционных и практических занятий данного курса: презентации, примеры, задания к лабораторным работам.
2. <http://edu-netcracker.com:8080> (SQL Skill Bench) – система автоматической проверки задач на написание SQL запросов.
3. <http://docs.oracle.com/javaee/7/tutorial/doc/> – пособие по Java EE от Oracle (на английском).
4. <http://java-course.appspot.com/students> – пособие по Java EE от А. Сабурова
5. http://docs.oracle.com/cd/E17781_01 – документация по Oracle XE (11.2)

8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень необходимого программного обеспечения и информационных справочных систем (при необходимости)

На практических занятиях и при самостоятельной работе используется следующее программное обеспечение:

- веб-браузер с доступом в интернет;
- среда разработки (IDE for Java EE: IDEA или NetBeans или Eclipse), включающая средства модульного тестирования (JUnit), сборки (Maven) и доступа к коллективному репозиторию (Subversion);
- сервер приложений (GlassFish);
- СУБД (Oracle XE; возможно размещение на сервере вместо локального компьютера);
- средство работы с базой данных (Oracle SQL Developer);
- желательно также CASE-средство с поддержкой UML (StarUML или иное).

9. Методические указания для обучающихся по освоению дисциплины (модуля)

Успешное освоение данной дисциплины требует большой самостоятельной работы студента. В программе курса приведено время, минимально необходимое для работы студента над темой.

Самостоятельная работа включает в себя:

- изучение материалов лекционных занятий (как до, так и после занятий),
- чтение рекомендованной литературы,
- решение коротких задач по некоторым темам (SQL, PL/SQL и др.),
- (главное) командная разработка учебной информационной системы, разбитая на несколько этапов (лабораторных работ, посвященных конкретным технологиям Java EE)

Контроль за самостоятельной работой студента осуществляется по итогам решения коротких задач (в случае SQL – проверяемых автоматически), в процессе приемки лабораторных работ, а также индивидуальных консультаций.

Показателем владения материалом служит умение решать задачи, близкие к реальным (тесты с контрольными вопросами по теории имеют на порядок меньшую значимость). Не все задачи курса являются обязательными, но для достижения цели курса студенту необходимо решить как можно больше задач.

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

по направлению:	Прикладные математика и физика
профиль подготовки:	Радиотехника и компьютерные технологии Физтех-школа Радиотехники и Компьютерных Технологий кафедра инфокоммуникационных систем и сетей
курс:	3
квалификация:	бакалавр
Семестр, формы промежуточной аттестации: 6 (весенний) - Дифференцированный зачет	
Разработчик:	А.Б. Корчак, канд. физ.-мат. наук

1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.1 Анализирует задачу, выделяя этапы ее решения, действия по решению задачи
	УК-1.2 Находит, критически анализирует и выбирает информацию, необходимую для решения поставленной задачи
	УК-1.3 Рассматривает различные варианты решения задачи, оценивает их преимущества и недостатки
	УК-1.4 Грамотно, логично, аргументированно формирует собственные суждения и оценки
	УК-1.5 Определяет и оценивает практические последствия возможных вариантов решения задачи
ОПК-1 Способен применять фундаментальные знания, полученные в области физико-математических и (или) естественных наук, и использовать их в профессиональной деятельности	ОПК-1.1 Способен анализировать поставленную задачу, намечать пути ее решения
	ОПК-1.2 Способен строить математические модели, производить количественные расчеты и оценки
	ОПК-1.3 Способен определять границы применимости полученных результатов
ОПК-4 Способен осуществлять сбор и обработку научно-технической и (или) технологической информации для решения фундаментальных и прикладных задач	ОПК-4.1 Владеет методами научного поиска и интеллектуального анализа информации при решении задач профессиональной деятельности
	ОПК-4.4 Владеет навыками работы с компьютером и компьютерными сетями с целью получения, хранения и обработки научной (технической, технологической) информации
ПК-4 Способен критически оценивать применимость используемых методик и методов	ПК-4.1 Знает численные порядки величин, характерных для соответствующей профессиональной области
	ПК-4.3 Способен обосновать причинно-следственные отношения используемых понятий и моделей

2. Показатели оценивания компетенций

В результате изучения дисциплины «Корпоративные информационные системы» обучающийся должен:

знать:

- синтаксис языка Java, принципы объектно-ориентированного программирования и особенности их реализации в языке Java;
- некоторые шаблоны объектно-ориентированного проектирования, применимые в веб-приложениях и корпоративных системах;
- многоуровневую архитектуру КИС, модель Java Enterprise Edition;
- основные сведения о серверах приложений, IoC, CDI;
- структуру библиотеки Java EE, назначение ее ключевых компонентов;
- основные понятия реляционных баз данных и базовые принципы их проектирования.

уметь:

- читать и рисовать ER-диаграммы для структуры БД;
- проектировать несложную БД из нескольких сущностей;
- использовать рассматриваемые библиотеки Java EE (пакеты javax.*) для написания реальных программ, библиотеку log4j – для эффективного логирования отладочной информации, библиотеку JUnit – для написания модульных тестов программ;
- писать несложный код на других языках кроме Java, необходимый для хранения данных в реальных Java-программах (XML, SQL, JPAQL) и для построения пользовательских веб-интерфейсов к серверным Java-приложениям (HTML, JSF);
- эффективно и по назначению использовать интегрированные среды разработки (Java IDE), репозиторий коллективной работы (Subversion), средства сборки (Maven).

владеть:

- навыками освоения большого объема информации в области программирования;
- навыками самостоятельного и быстрого поиска недостающей информации в Интернете;
- навыками отладки приложений с использованием дебаггера в Java IDE, логирования, а также правильной обработки исключительных ситуаций в коде;
- культурой программирования: умением писать код, читабельный с точки зрения стиля, имен переменных и методов, комментариев и др.; навыком проектирования модульных тестов к коду;
- навыками командной разработки программ (в т.ч. навыком версионирования кода), навыками краткого документирования программ (в т.ч. с применением языка UML).
- навыками отладки SQL запросов.

3. Перечень типовых (примерных) вопросов, заданий, тем для подготовки к текущему контролю

С целью контроля освоения обучающимися учебного материала проводится устный опрос в начале занятия по теме прошлому семинару или в конце занятия по пройденной теме.

4. Перечень типовых (примерных) вопросов и тем для проведения промежуточной аттестации обучающихся

Пример контрольных вопросов для теста по последнему разделу дисциплины (в [квадратных скобках указаны сложности вопросов – от 1 до 5]):

1.[3] Какой из перечисленных пунктов НЕ является типом EJB бина?

- Управляемый сообщением
- Объектный
- Контекстный
- Сессионный

2.[5] В чем отличие JMS от RPC?

- При использовании RPC клиент и сервер, как правило, должны быть написаны на одном языке. JMS, напротив, может без дополнительных расширений организовать взаимодействие гетерогенных систем.
- RPC-вызов, как правило, синхронный. JMS использует асинхронные вызовы.
- RPC, в отличие от JMS, использует сессии для сохранения состояний в процессе взаимодействия.
- При RPC-вызове параметры передаются по значению, при отправке JMS-сообщения – по ссылке.

3.[3] Выберите модели messaging'a, поддерживаемые JMS

- 50% point-to-point
- -40% wide broadcast
- 50% publish/subscribe
- -40% digital subscriber line

4.[4] JMS Connection представляет собой:

- -50% Соединение между источником сообщений и их получателем
- 50% Соединение источника сообщений и JMS Provider
- 50% Соединение получателя событий и JMS Provider
- -50% Абстракцию над JDBC Connection для сохранения сообщений в случае, если их невозможно доставить сразу

5.[4] Выберите те части JMS-сообщения, которые обязательно должны присутствовать в валидном сообщении

- 100% Header
- -25% Properties
- -25% Body
- -25% Footer
- -25% Attachment

6.[4] Что из перечисленного не является валидным body format для JMS-сообщения?

- BytesMessage

- TextMessage
- StreamMessage
- MapMessage
- ThreadMessage

7.[3] Основным исключением, декларируемым в JMS API, является:

- MessagingException
- JMSEException
- DeliveryException
- RemoteException

8.[2] Какой из этих прикладных протоколов SOAP использует в качестве транспорта:

- FTP
- NTP
- XMPP
- HTTP
- HTML

9.[2] Какая аннотация JAX-WS помечает класс как Service Endpoint?

- @WebAccess
- @WebMethod
- @WebService
- @Service
- @ServiceEndpoint

10.[3] Чтобы передать объект Java-класса как параметр JAX-WS веб-сервиса, необходимо:

- Чтобы класс реализовывал интерфейс Serializable
- Чтобы класс был описан в соответствующем WSDL
- Чтобы существовал JAXB-маппинг этого класса в XML
- Все вышеперечисленное

11.[3] Выберите допустимые режимы (modes) передачи параметров в процедуры

- 33% IN
- 33% OUT
- 33% IN OUT

12.[5] Для перехвата исключений всех типов используется конструкция:

- 30% when Others then
- -50% when Exceptions then
- 70% when Exception then
- -50% catch Exception ex
- -50% catch Exception then

13.[1] В PL/SQL можно объявлять / использовать свой тип исключения

- Нет
- Да, с помощью синтаксиса имя_типа_искл exception / throw имя_типа_искл
- Да, с помощью синтаксиса имя_типа_искл exception / raise имя_типа_искл

14.[2] Если предполагается, что функция при одном и том же наборе входных данных возвращает одно и то же значение, то такая функция может быть помечена как

- Константная
- Детерминистическая
- Недетерминистическая
- Оптимизированная по аргументам

15.[5] Модули на уровне блока:

- 33% описываются внутри секции declare PL/SQL блока или модуля;
- 33% не содержат ключевых слов create or replace;
- -67% не содержат секции объявления локальных параметров
- 33% видны только внутри своего блока или модуля

16.[5] Перегруженные программные блоки могут отличаться:

- 33% только списком входных аргументов
- 33% только типом блока (процедура/функция)

- 33% списком входных аргументов и одновременно типом блока
- -67% только типом выходного значения

17.[1] Триггеры DML:

- вызываются операциями DML
- содержат в себе ответные операции DML
- могут явно запускаться программистом

18.[3] Триггеры DML могут вызываться:

- 50% отдельно для каждой записи, затрагиваемой вызывающей инструкцией
- 50% для инструкции модификации данных в целом
- -50% для события – запуска экземпляра Oracle
- -50% для события – создания новой таблицы

19.[1] В случае возникновения ошибки в ходе выполнения триггера (в котором нет автономных транзакций) будет откатан только триггер, но не вызвавшая его операция. Верно ли это утверждение?

20.[2] Для того чтобы получить доступ к данным, на модификацию которых сработал триггер типа «for each row», необходимо воспользоваться:

- псевдотаблицами inserted, deleted, updated
- псевдотаблицами inserted, deleted
- псевдозаписями :new, :old

21.[2] Для того чтобы узнать, на какую операцию сработал триггер, необходимо воспользоваться:

- логическими предикатами inserting, updating, deleting
- агрегатными функциями count(new), count(old)
- агрегатными функциями count(inserted), count(deleted)
- агрегатными функциями count(inserted), count(deleted), count(updated)

22.[4] Как лучше отменить базовую операцию из тела триггера?

- Вызвать rollback или rollback tran
- Вызвать rollback to savepoint sp_name
- Вызвать raise_application_error
- Искусственно выполнить ошибочный код в теле триггера

Пример контрольной задачи по SQL:

Для заданной базы данных решить следующую задачу. Предположим, что в компании зарплата начисляется пятого числа каждого месяца, и зарплату сотрудник начинает получать, начиная со следующего месяца после месяца приема на работу. Вывести для всех сотрудников: дату приема на работу и дату, когда сотруднику была выплачена первая зарплата.

Пример заданий по Java EE на оценку «удовлетворительно» (в случае невыполнения полноценного учебного проекта по разработке информационной системы):

1. Написать сервлет и HTML-файл с формой, по submit'у которой сервлет генерирует приветствие на основе введенных данных. Открывать HTML-файл также при обращении по GET к сервлету. Выводить число вызовов сервлета, вызовов текущего экземпляра, число текущих экземпляров, число уничтоженных экземпляров
2. Переписать HTML-файл с формой в JSF-файл, так что сервлет после POST открывает его не пустым, а с введенными данными. Выводить error page в случае некорректности введенных данных. Одно из полей формы сделать списком, наполняемым по данным от сервлета.
3. Создать stateless session bean с методами getWhole* и getNamesBy*, где * – к.-л. атрибут к.-л. таблицы. Написать JSF-клиент, вызывающий методы этого bean и показывающий результаты.

Критерии оценивания

Оценка отлично (10) выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины, проявляющему интерес к данной предметной области, продемонстрировавшему умение уверенно и творчески применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

Оценка отлично (9) выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

Оценка отлично (8) выставляется студенту, показавшему систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, правильное обоснование принятых решений, с некоторыми недочетами.

Оценка хорошо (7) выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но недостаточно грамотно обосновывает полученные результаты.

Оценка хорошо (6) выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности.

Оценка хорошо (5) выставляется студенту, если он в основном знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач достаточно большое количество неточностей.

Оценка удовлетворительно (4) выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он освоил основные разделы учебной программы, необходимые для дальнейшего обучения, и может применять полученные знания по образцу в стандартной ситуации.

Оценка удовлетворительно (3) выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, допускающему ошибки в формулировках базовых понятий, нарушения логической последовательности в изложении программного материала, слабо владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и с трудом применяет полученные знания даже в стандартной ситуации.

Оценка неудовлетворительно (2) выставляется студенту, который не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных принципов и не умеет использовать полученные знания при решении типовых задач.

Оценка неудовлетворительно (1) выставляется студенту, который не знает основного содержания учебной программы дисциплины, допускает грубейшие ошибки в формулировках базовых понятий дисциплины и вообще не имеет навыков решения типовых практических задач.

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

Дифференцированный зачет выставляется по итогам работы в семестре: 3 практические работы и их сдача (сопровождающая демонстрацией в т.ч. теоретических знаний); короткие задачи; 5 тестов с контрольными вопросами по теории.

Вышеуказанные критерии оценивания (см. п. 2) относятся, в основном, к приемке практических работ (этапов учебного проекта), что в совокупности дает 75% вклада в итоговую оценку студента по дисциплине. Вместо учебного проекта в качестве практических работ студенты могут выполнять приведенные в пункте 1 задачи по Java EE, которые имеют в разы меньший объем, и ограничивают оценку по дисциплине оценкой «удовлетворительно (4)».

Еще 15% дают тесты с контрольными вопросами по теории, 10% – автоматически проверяемые задачи по SQL. Имеющиеся короткие задачи на отдельные неосновные темы дисциплины (кроме SQL) являются сугубо необязательными, но в случае их решения могут повысить оценку (добавка до 20%).