

Лабораторные работы

Лабораторная работа #1

Введите вариант

Чтобы узнать сайт, введите свой номер варианта.

Описать бизнес-процесс в соответствии с нотацией BPMN 2.0, после чего реализовать его в виде приложения на базе Spring Boot.

Порядок выполнения работы:

1. Выбрать один из бизнес-процессов, реализуемых сайтом из варианта задания.
2. Утвердить выбранный бизнес-процесс у преподавателя.
3. Специфицировать модель реализуемого бизнес-процесса в соответствии с требованиями BPMN 2.0.
4. Разработать приложение на базе Spring Boot, реализующее описанный на предыдущем шаге бизнес-процесс. Приложение должно использовать СУБД PostgreSQL для хранения данных, для всех публичных интерфейсов должны быть разработаны REST API.
5. Разработать набор curl-скриптов, либо набор запросов для REST клиента Insomnia для тестирования публичных интерфейсов разработанного программного модуля. Запросы Insomnia оформить в виде файла экспорта.
6. Развернуть разработанное приложение на сервере [helios](#).

Содержание отчёта:

1. Текст задания.
2. Модель потока управления для автоматизируемого бизнес-процесса.
3. UML-диаграммы классов и пакетов разработанного приложения.
4. Спецификация REST API для всех публичных интерфейсов разработанного приложения.
5. Исходный код системы или ссылка на репозиторий с исходным кодом.
6. Выводы по работе.

Вопросы к защите лабораторной работы:

1. Понятие бизнес-логики в программных системах. Уровень бизнес-логики в многоуровневой архитектуре программных систем.
2. Основные концепции, используемые при разработке бизнес-логики. CDI, IoC, управление транзакциями, безопасность, распределённая обработка данных.
3. Моделирование бизнес-процессов. BPM и BPMN.
4. Спецификация BPMN 2.0. Принципы составления и основные элементы моделей бизнес-процессов.
5. Объекты потока управления, роли и артефакты в BPMN.
6. Использование Spring Framework для реализации бизнес-логики. Реализация CDI и IoC. Связь уровня бизнес-логики с другими уровнями архитектуры программных систем в Spring.

Чтобы узнать задание, введите свой номер варианта.

Правила выполнения работы:

1. Все изменения, внесённые в реализуемый бизнес-процесс, должны быть учтены в описывающей его модели, REST API и наборе скриптов для тестирования публичных интерфейсов модуля.
2. Доработанное приложение необходимо развернуть на сервере **helios**.

Содержание отчёта:

1. Текст задания.
2. Модель потока управления для автоматизируемого бизнес-процесса.
3. Спецификация пользовательских привилегий и ролей, реализованных в приложении.
4. UML-диаграммы классов и пакетов разработанного приложения.
5. Спецификация REST API для всех публичных интерфейсов разработанного приложения.
6. Исходный код системы или ссылка на репозиторий с исходным кодом.
7. Выводы по работе.

Вопросы к защите лабораторной работы:

1. Понятие транзакции. Особенности реализации транзакций на уровне бизнес-логики, отличия от транзакций на уровне БД.
2. Распределённые транзакции, спецификация XA. Реализация в приложениях на базе Jakarta EE и Spring.
3. Реализация управления транзакциями в Spring. Аннотация **@Transactional**. Декларативное и программное управления транзакциями.
4. Jakarta Transactions / Java Transaction API. Основные принципы и программные интерфейсы. Работа с JTA в приложениях на базе Spring / Spring Boot.
5. Менеджеры транзакций: Bitronix, Atomikos, Narayana. Использование менеджера транзакций Jakarta EE в приложениях на базе Spring / Spring Boot.
6. Разграничение доступа и политики безопасности в корпоративных приложениях. Пользователи, роли и привилегии. Реализация политик безопасности на уровне бизнес-логики.
7. Технология Spring Security. Основные понятия, аннотации, конфигурационные файлы и API. Использование на уровне бизнес-логики.
8. Технология JAAS. Основные понятия, конфигурационные файлы и API. Использование на уровне бизнес-логики, в т.ч. совместно с Spring Security.
9. Способы хранения информации об учётных записях пользователей в приложениях на Java.
10. Подходы к реализации аутентификации пользователей в приложениях на Java.

Лабораторная работа #3

Введите вариант:

...распределением элементов между несколькими узлами и выполнением периодических операций с использованием планировщика задач, а также интеграцию с внешней информационной системой.

Требования к реализации асинхронной обработки:

1. Перед выполнением работы необходимо согласовать с преподавателем набор прецедентов, в реализации которых целесообразно использование асинхронного распределённого выполнения задач. Если таких прецедентов использования в имеющейся бизнес-процесса нет, нужно согласовать реализацию новых прецедентов, доработав таким образом модель бизнес-процесса из лабораторной работы #1.
2. Асинхронное выполнение задач должно использовать модель доставки "очередь сообщений".
3. В качестве провайдера сервиса асинхронного обмена сообщениями необходимо использовать очередь сообщений на базе Apache ActiveMQ.
4. Для отправки сообщений необходимо использовать протокол OpenWire с использованием библиотеки Java OpenWire Transport.
5. Для получения сообщений необходимо использовать слушателя сообщений JMS на базе Spring Boot (@JmsListener).

Требования к реализации распределённой обработки:

1. Обработка сообщений должна осуществляться на двух независимых друг от друга узлах сервера приложений.
2. Если логика сценария распределённой обработки предполагает транзакционность выполняемых операций, они должны быть включены в состав распределённой транзакции.

Требования к реализации запуска периодических задач по расписанию:

1. Согласовать с преподавателем прецедент или прецеденты, в рамках которых выглядит целесообразным использовать планировщик задач. Если такие прецеденты отсутствуют - согласовать с преподавателем новые и добавить их в модель автоматизируемого бизнес-процесса.
2. Реализовать утверждённые прецеденты с использованием планировщика задач Spring (@Scheduled).

Требования к интеграции с внешней Корпоративной Информационной Системой (EIS):

1. Корпоративная Информационная Система, с которой производится интеграция, а также её функциональные возможности выбираются на усмотрение преподавателя и согласуются с ним.
2. Взаимодействие с внешней Корпоративной Информационной Системой должно быть реализовано с помощью технологии JCA (Jakarta Connectors).

Правила выполнения работы:

1. Все изменения, внесённые в реализуемый бизнес-процесс, должны быть учтены в описывающей его модели, REST API и наборе скриптов для тестирования публичных интерфейсов модуля.

1. Текст задания.
2. Модель потока управления для автоматизируемого бизнес-процесса со всеми внесёнными изменениями.
3. UML-диаграммы классов и пакетов разработанного приложения.
4. Спецификация REST API для всех публичных интерфейсов разработанного приложения.
5. Диаграмма развёртывания (Deployment Diagram), визуализирующая интеграцию с EIS.
6. Исходный код системы или ссылка на репозиторий с исходным кодом.
7. Выводы по работе.

Вопросы к защите лабораторной работы:

1. Асинхронное выполнение задач. Преимущества и недостатки, подходы к реализации.
2. Спецификация Java Message Service.
3. Ресурсы и сообщения JMS. Модели взаимодействия "очередь" и "подписка".
Распределённая обработка сообщений.
4. Протоколы взаимодействия с очередями сообщений: MQTT, AMQP, STOMP, XMPP.
Отправка сообщений с использованием HTTP + WebSockets.
5. Apache ActiveMQ. Архитектура, способы взаимодействия, поддерживаемые протоколы, особенности реализации JMS. Протокол OpenWire и его реализации для различных платформ.
6. RabbitMQ. Архитектура, способы взаимодействия, поддерживаемые протоколы, особенности реализации JMS.
7. Apache Kafka. Особенности обработки сообщений, сходства и отличия с очередями сообщений. Архитектура, особенности построения масштабируемых решений, интеграция с Service Discovery.
8. Периодические задачи, планировщики выполнения задач.
9. Cron. Архитектура, интеграция в ОС, способы конфигурации, синтаксис Cron Expression.
10. Quartz. Архитектура, интеграция с приложением, способы конфигурации.
11. Выполнение периодических задач в Java / Jakarta EE и Spring. Java / Jakarta EE Timer Services и Spring `@Scheduled`.
12. Подходы к интеграции информационных систем -- в общем, и на примере JCA / Jakarta Connectors.

Лабораторная работа #4

Введите вариант:

Чтобы узнать задание, введите свой номер варианта.

Вопросы к защите лабораторной работы:

1. BPM-фреймворки. Особенности реализации бизнес-логики, преимущества и недостатки по сравнению с реализацией логики "вручную".
2. Платформа Camunda. Архитектура, состав, поддерживаемые языки, особенности разработки программ.
3. Механизмы редактирования бизнес-процессов в Camunda. Camunda Modeler.
Использование "внешних" редакторов.
4. Роли и права доступа в Camunda.

8. Реализация GUI в Camunda. Управление формами.

