**Министерство науки и высшего образования Российской Федерации**

Федеральное государственное автономное образовательное учреждение

высшего образования «Уральский федеральный университет

имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий - РТФ

Центр ускоренного обучения

Отчет по лабораторной работе № 3

по дисциплине «Технологии Программирования»

Старший преподаватель: Н.А. Архипов

Студент гр. РИВ-220938у М.С. Ермаков

Екатеринбург 2023

**1. Цель работы:**

**Цель:** Доработать простой REST сервис

**Учебные вопросы:**

1) Доработать логику изменения исходящего response

2) Добавить и настроить логирование в приложении

3) Ответить письменно на вопросы в соответствии с номером в списке

4) Оформить отчет и прикрепить его на сайте https://edu.itlearn.ru/ в соответствующем курсе, в соответствующем разделе

**2. Описание задачи:**

**Описание входящего (request) и исходящего (response) сообщений, схема работы сервиса**

Реализация дополнительного функционала

Необходимо самостоятельно доработать приложение:

1. Добавьте вывод логов уровня INFO везде, где изменяется response и request таким образом чтобы по логам можно было отследить всю логику работы приложения.

2. Добавьте вывод логов уровня ERROR в те места программы, где выбрасываются исключения, а также добавить сообщения которые возникают при ошибке в bindingResult.

3. В классе Request измените тип поля systemName с типа String на тип созданного самостоятельно перечисления Systems. Элементами перечисления сделайте следующие значения ERP («Enterprise Resource Planning»), CRM («Customer Relationship Management»), WMS (Warehouse Management System).

Вопросы

Письменно ответьте на вопросы, если ваш номер в списке группы четный, то нужно ответить на четные вопросы, если нечетный, то нужно ответить на нечетные вопросы.

1. Что такое ERP система? Приведите примеры.

2. Что такое CRM система? Приведите примеры.

3. Что такое WMS система? Приведите примеры.

4. Какие библиотеки логгирования бывают в Spring?

5. Какие уровник логгирования в @Slf4j, в чем их отличие?

6. Что означает аннотация @Qualifier?

**3. Ход выполнения:**

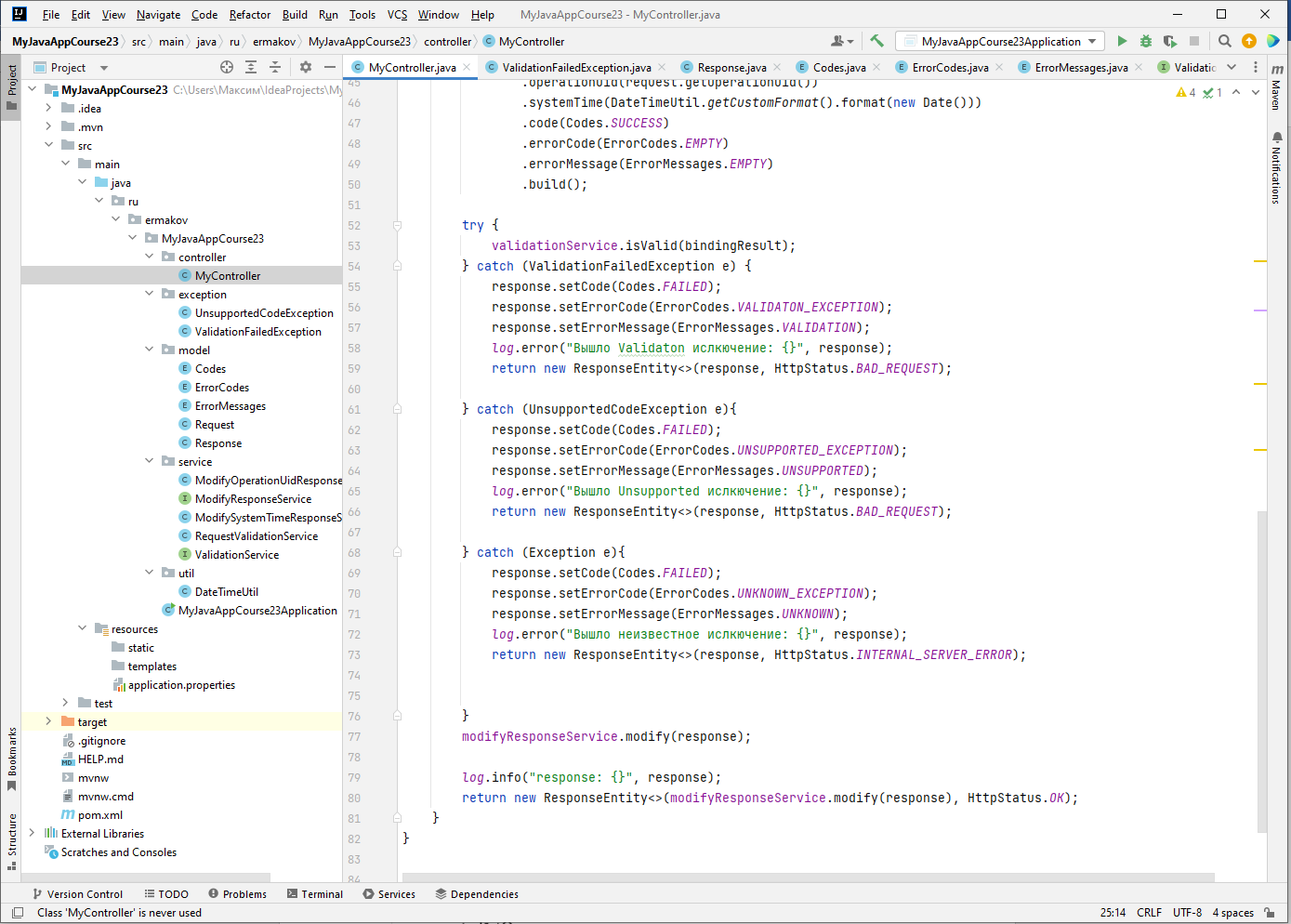


Рисунок 1 – Структура Проекта

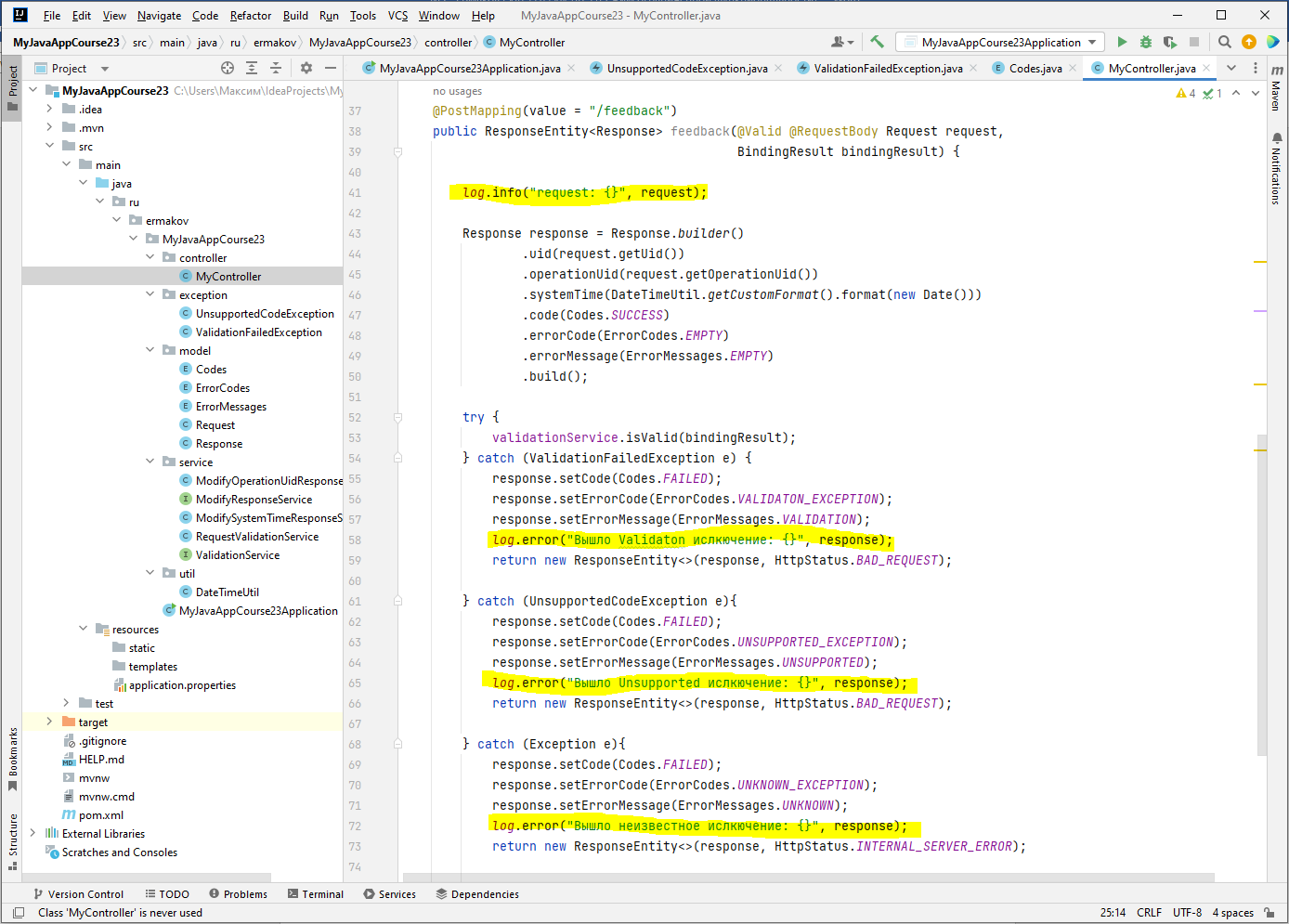


Рисунок 2 – Добавление Логирования уровня info и error

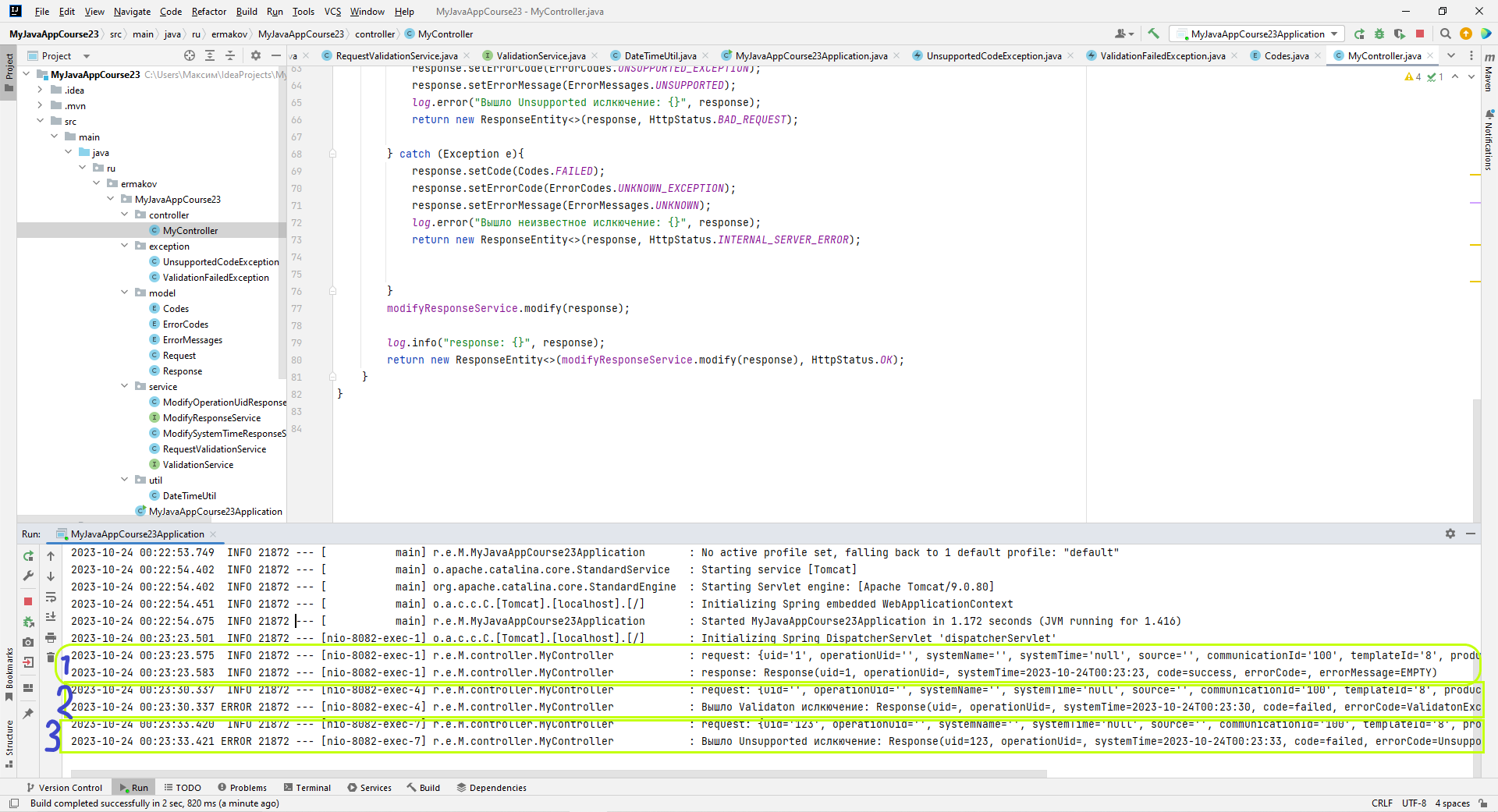


Рисунок 3 – Тестирования логирования через постман (3 кейса, рабочий, ошибка валидации, и собственное исключение unsupported)

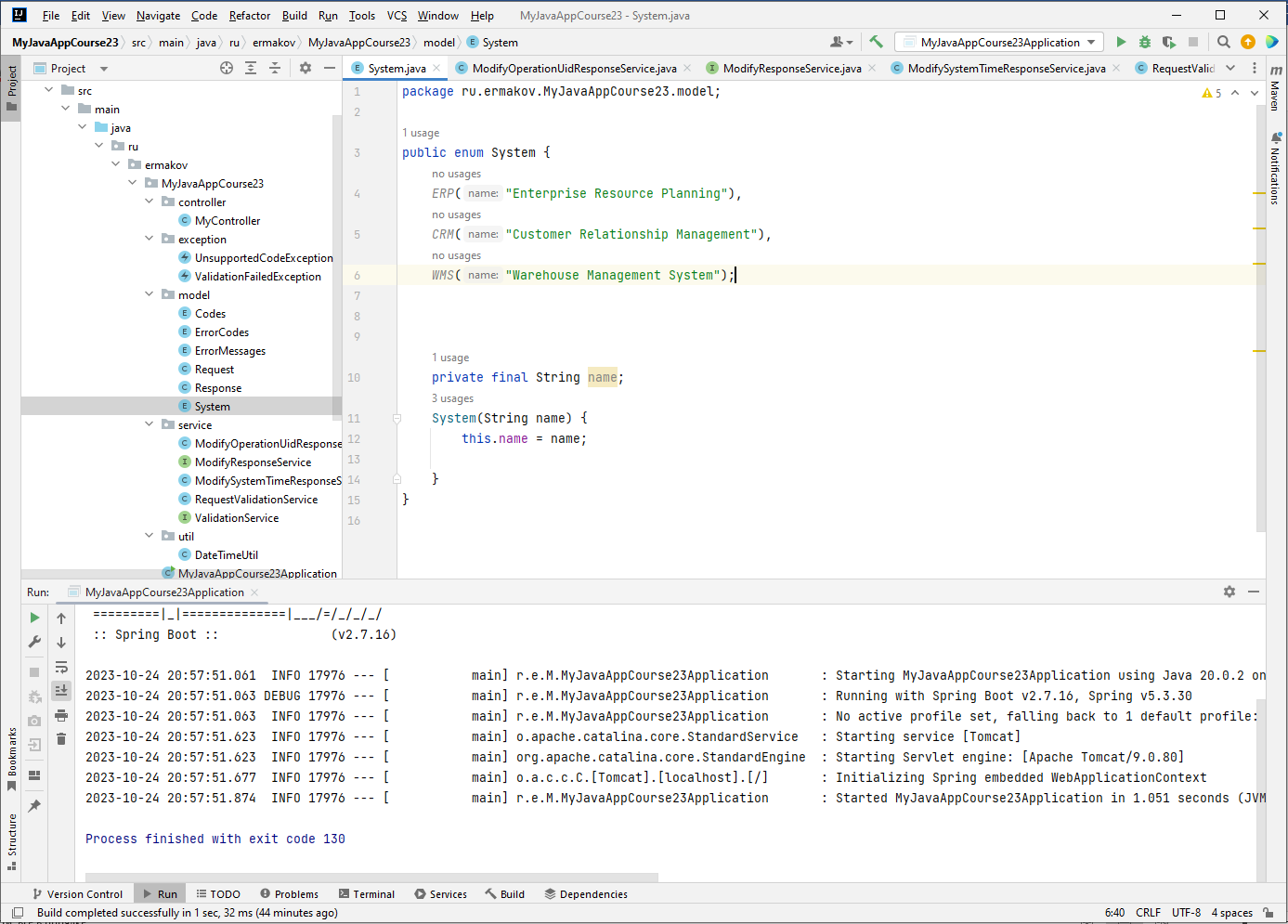


Рисунок 4 – Добавление собственного перечисления enum System

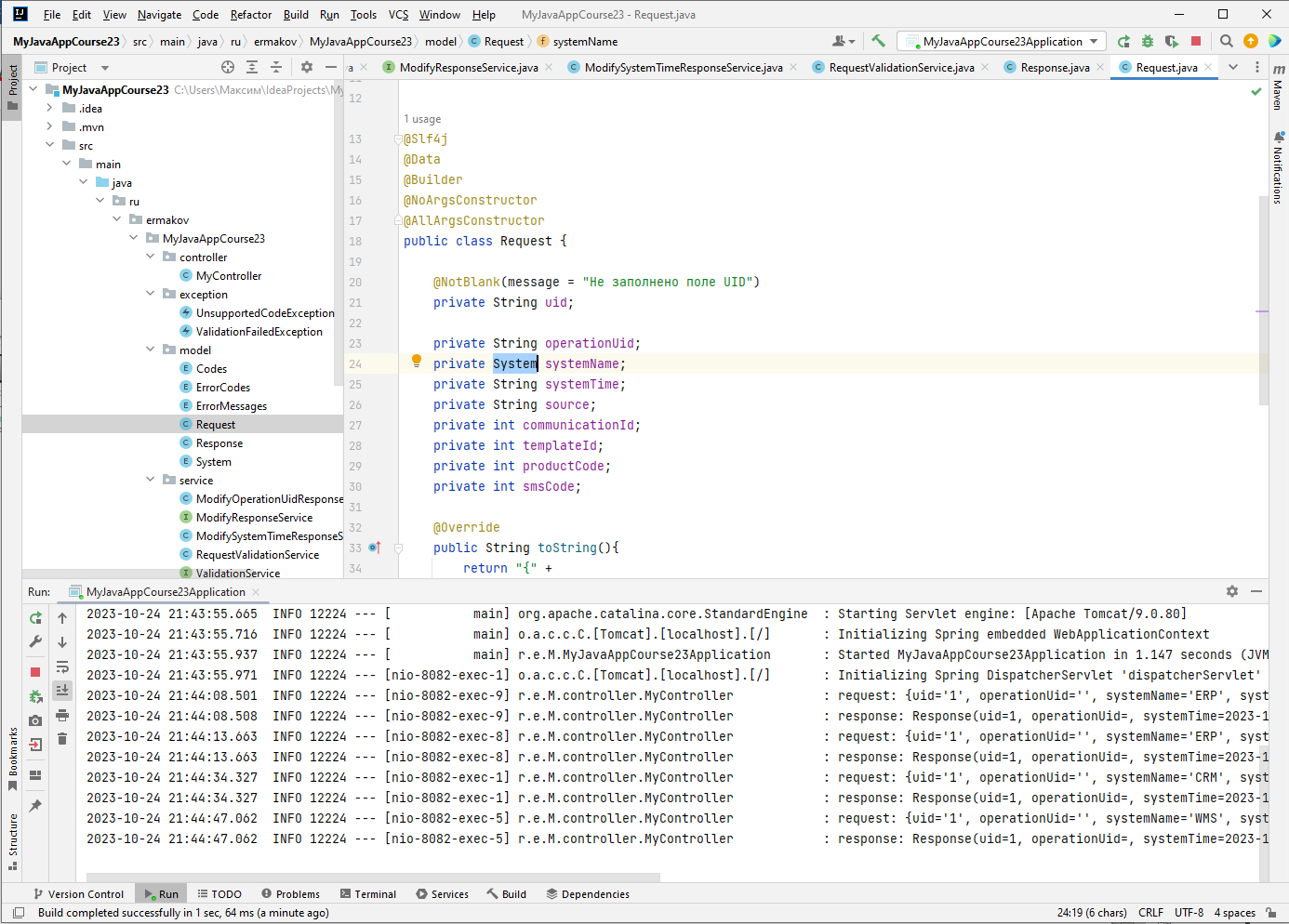


Рисунок 5 – Изменение поля с String на System для

**Ответы на вопросы:**

2) Что такое CRM система? Приведите примеры

**Customer Relationship Management (CRM)** — это система управления взаимоотношениями с клиентом. CRM помогает компаниям работать с клиентами.

Разные виды CRM-систем помогают:

* координировать работу предприятия;
* обеспечивать коммуникацию внутри компании;
* хранить информацию о клиентах;
* планировать встречи, звонки, рассылки сообщений;
* автоматизировать продажи;
* оценивать эффективность рекламных кампаний;
* автоматически формировать отчеты.

В общем виде CRM-система работает так:

Интегрирует разные каналы коммуникации в единый интерфейс;

автоматически обрабатывает различные данные;

объединяет задачи в непрерывный рабочий поток в рамках одной платформы

обеспечивает постоянный контакт с клиентом.

Сотрудники предприятия регистрируются в системе и получают доступ к базе данных — сведениям о клиентах, их контактам, истории покупок. Затем, в зависимости от профиля, каждый сотрудник выполняет свою работу. Например, маркетологи создают рекламную рассылку и отправляют ее через CRM нужному сегменту аудитории, операторы контакт-центра обзванивают клиентов, которые оставили заявки на обслуживание. Система фиксирует каждое действие: это помогает составлять отчеты, отслеживать эффективность сотрудников и стратегий, продумывать пути развития компании.

Однако для каждой системы набор функций будет своим, как и принципы работы. Перечень инструментов определяет вид СРМ.

По данным исследования рынка CRM России самая используемая система — CRM Битрикс24.

4) Какие библиотеки логгирования бывают в Spring?

Spring Boot использует Commons Logging для всеохватывающего логирования (журналирования, logging) на внутреннем уровне, но оставляет открытой базовую реализацию лога. Конфигурации по умолчанию предусмотрены для Java Util Logging, Log4J2 и Logback. В каждом случае диспетчеры логирования предварительно сконфигурированы на использование консольного вывода с возможностью вывода в файл.

По умолчанию, если используются "Стартеры", для ведения лога используется Logback. Соответствующая маршрутизация Logback также предусмотрена для обеспечения корректной работы зависимых библиотек, использующих Java Util Logging, Commons Logging, Log4J или SLF4J.

Для Java доступно множество фреймворков логирования. Как правило, изменять зависимости логирования не нужно, и настройки Spring Boot по умолчанию работают просто отлично.

Если вы развертываете свое приложение в контейнере сервлетов или на сервере приложений, логирование, выполняемое с помощью API- интерфейса Java Util Logging, не маршрутизируется к логам приложения. Это предотвращает появление в логах приложения записей, выполняемых контейнером или другими приложениями, которые были развернуты в нем.

Выводятся следующие элементы:

* Дата и время: точность до миллисекунды и легкая сортировка.
* Уровень ведения лога: ERROR, WARN, INFO, DEBUG или TRACE.
* Идентификатор процесса.
* Разделитель --- для разделения начала фактических сообщений лога.
* Название потока: заключено в квадратные скобки (может быть усечено для вывода в консоль).
* Имя диспетчера логирования: обычно это имя исходного класса (часто сокращенное).
* Сообщение лога.

6) Что означает аннотация @Qualifier?

**Аннотация @Qualifier** позволяет уточнить имя бина, который надо внедрить. Используется прямо перед аргументом.

Имя бина такое же, как имя класса, но с маленькой буквы. При условии, что имя класса начинается только с одной большой буквы (наш случай).

**Ссылка** на репозиторий **github.com** =

<https://github.com/themakson/MyJavaAppCourse2/tree/main/MyJavaAppCourse23>

**Вывод:**

В ходе лабораторной работы 3 мы доработали созданный ранее простой REST сервис с помощью фреймворка Spring Boot в языке программирования Java, модифицировали и упростили, а также добавили логирование уровня info и error, для контролирования работы программы, а также научились создавать собсвтенные константы – enums.

|  |  |
| --- | --- |
| Сдал: | М.С. Ермаков |
| Проверил: | Н.А. Архипов |