**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»**

Студент группы РИЗ-240938у Бабенко М.А.

Преподаватель: Архипов Н.А.

**Екатеринбург, 2025 г.**

**Лабораторная работа N3**

Цель: доработать простой REST сервис.

**Задания для самостоятельной работы**

**Задание 1**

Cоздается пакет util, в нем класс DateTimeUtil. Метод в этом классе будет возвращать SimpleDateFormat.

Код класса DateTimeUtill показан на рисунке 1.

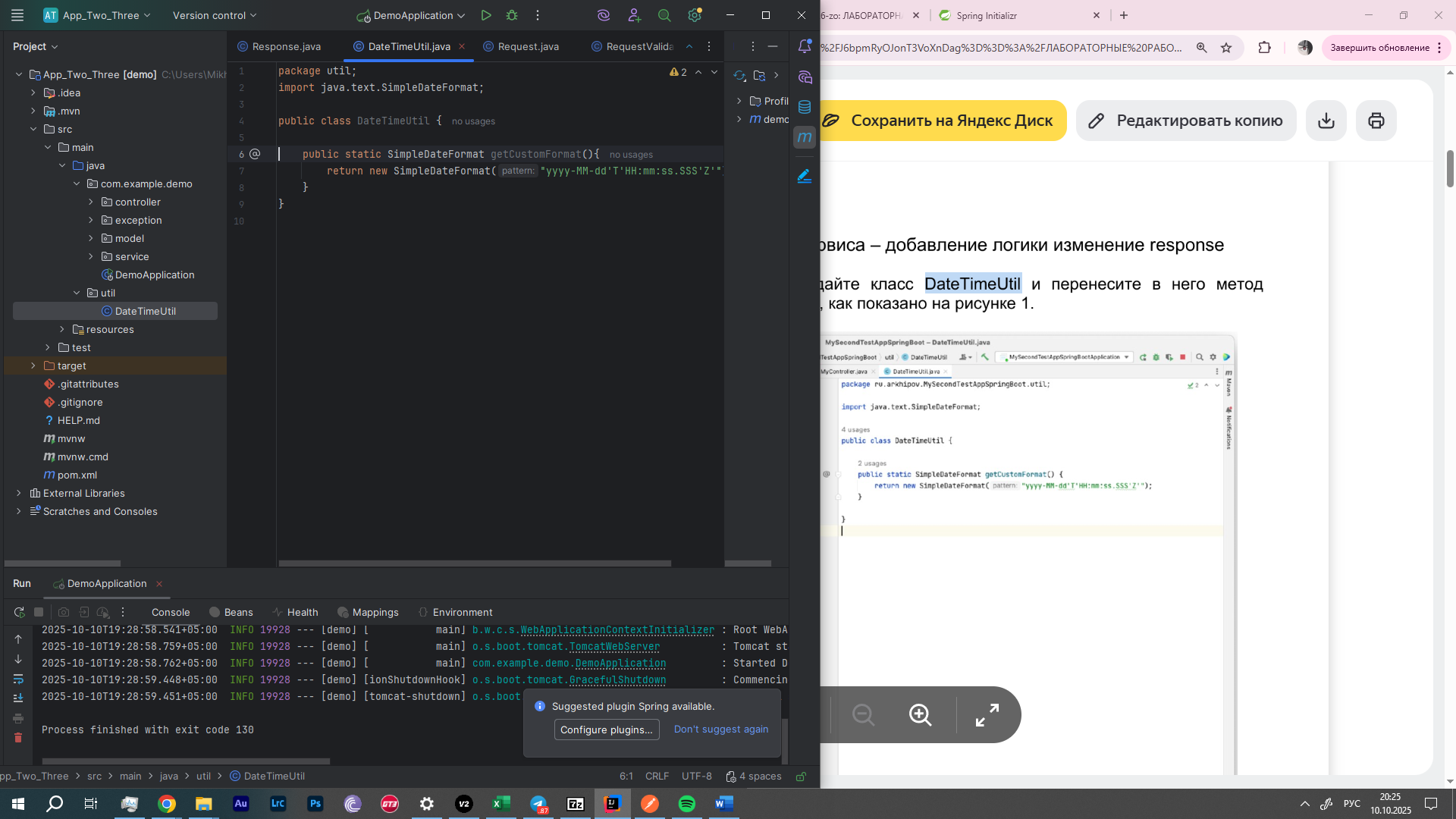


Рисунок 1 – Код класса DateTimeUtill

Затем, изменяем вызов объекта SimpleDateFormat вызовом только что созданного метода (через кастом формат).

Измененный блок кода MyConroller показан на рисунке 2.

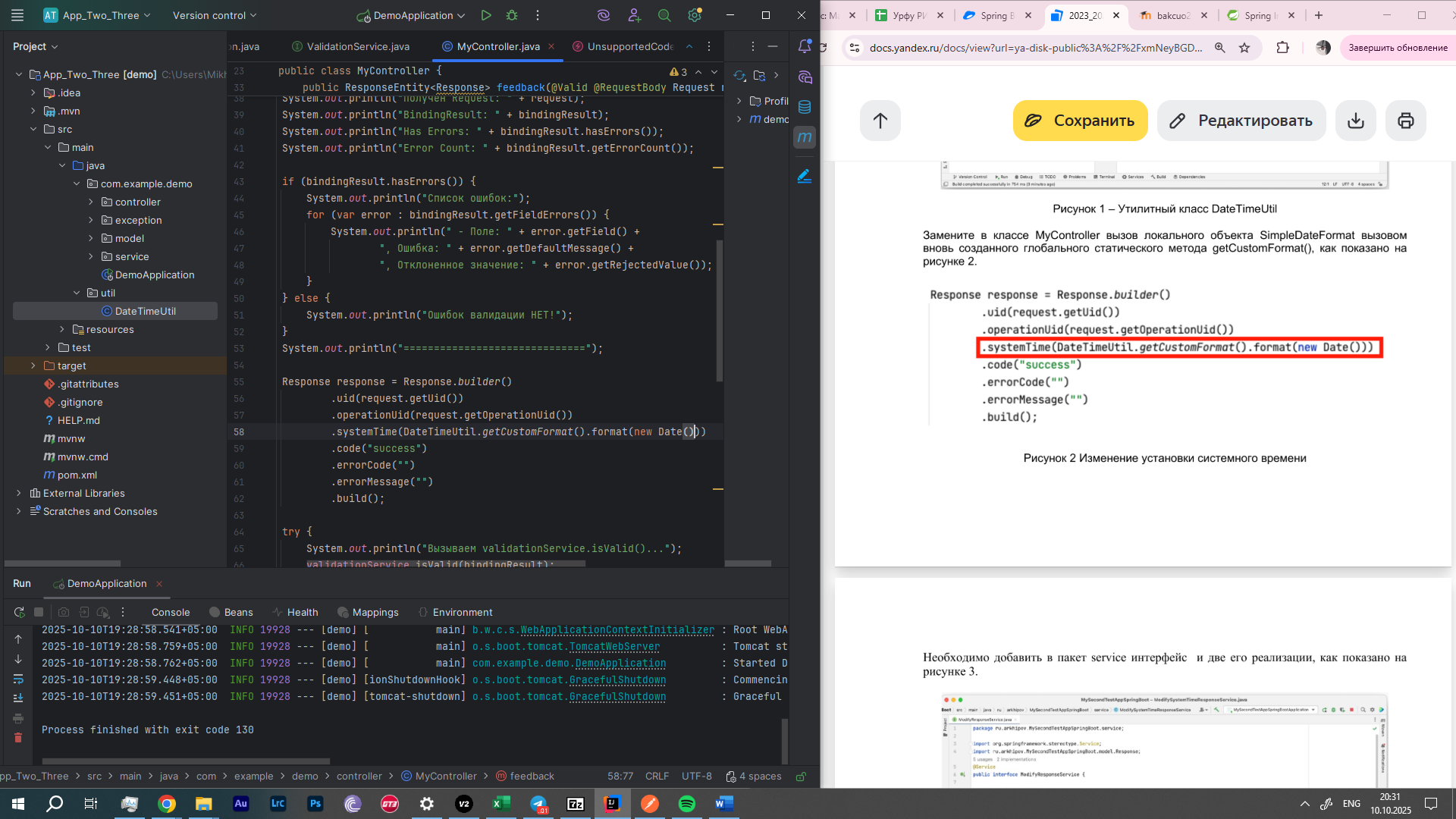


Рисунок 2 – Измененный блок кода MyConroller

Затем, создаем интерфейс ModifyResponseService внутри service и две его реализации – ModifyOperationUidResponseService и ModifySystemTimeResponseService, как указано в задании.

Результат создания показан на рисунке 3.

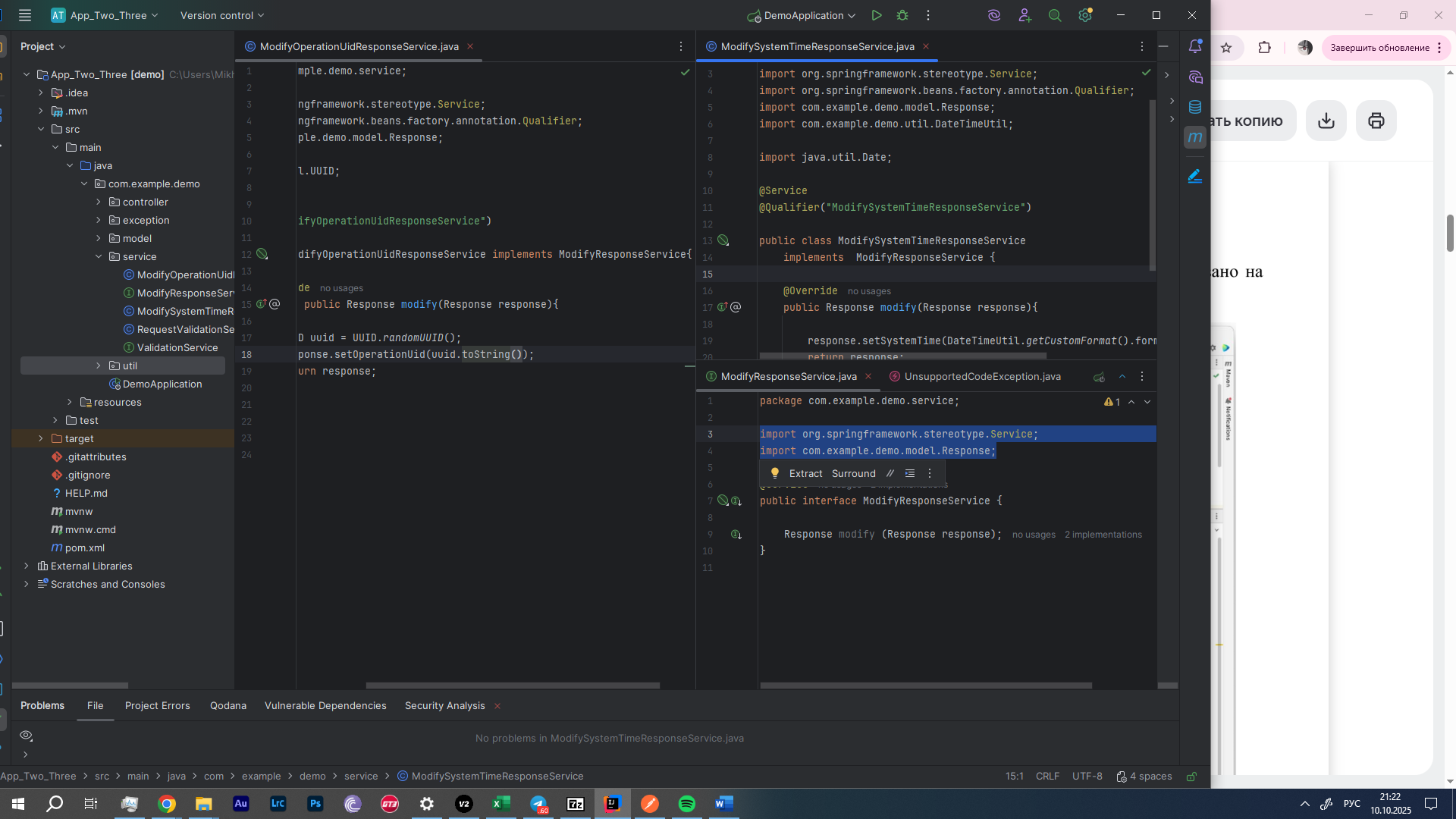


Рисунок 3 – Созданный интерфейс и реализации

Далее, добавляем три пересечения – Codes, ErrorMessages, ErrorCodes.

Результат создания показан на рисунке 4.

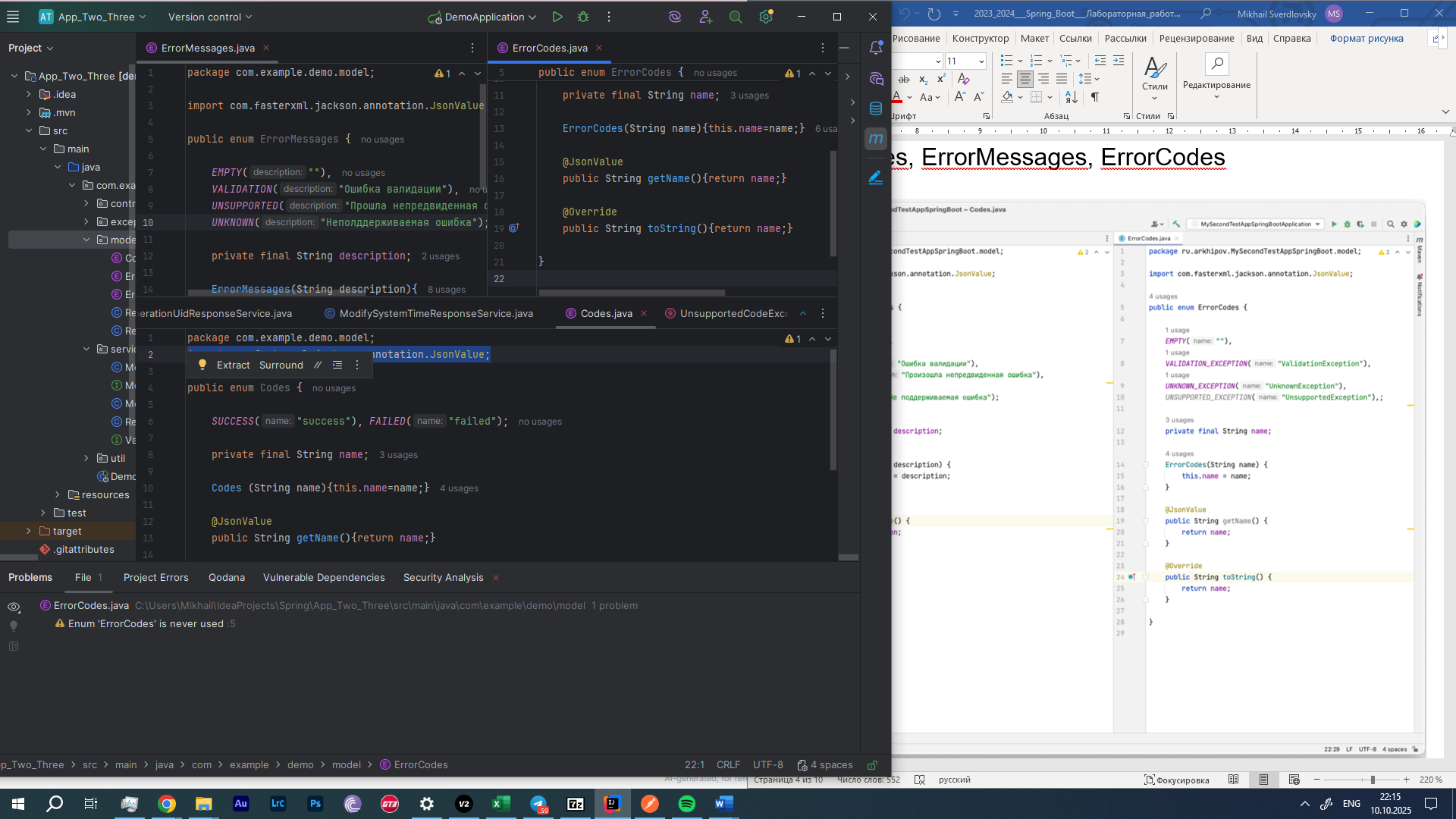


Рисунок 4 – Результат создания пересечений

Согласно заданию, модифицируем класс Response, изменив типы полей. Измененный блок кода класса Response показан на рисунке 5.

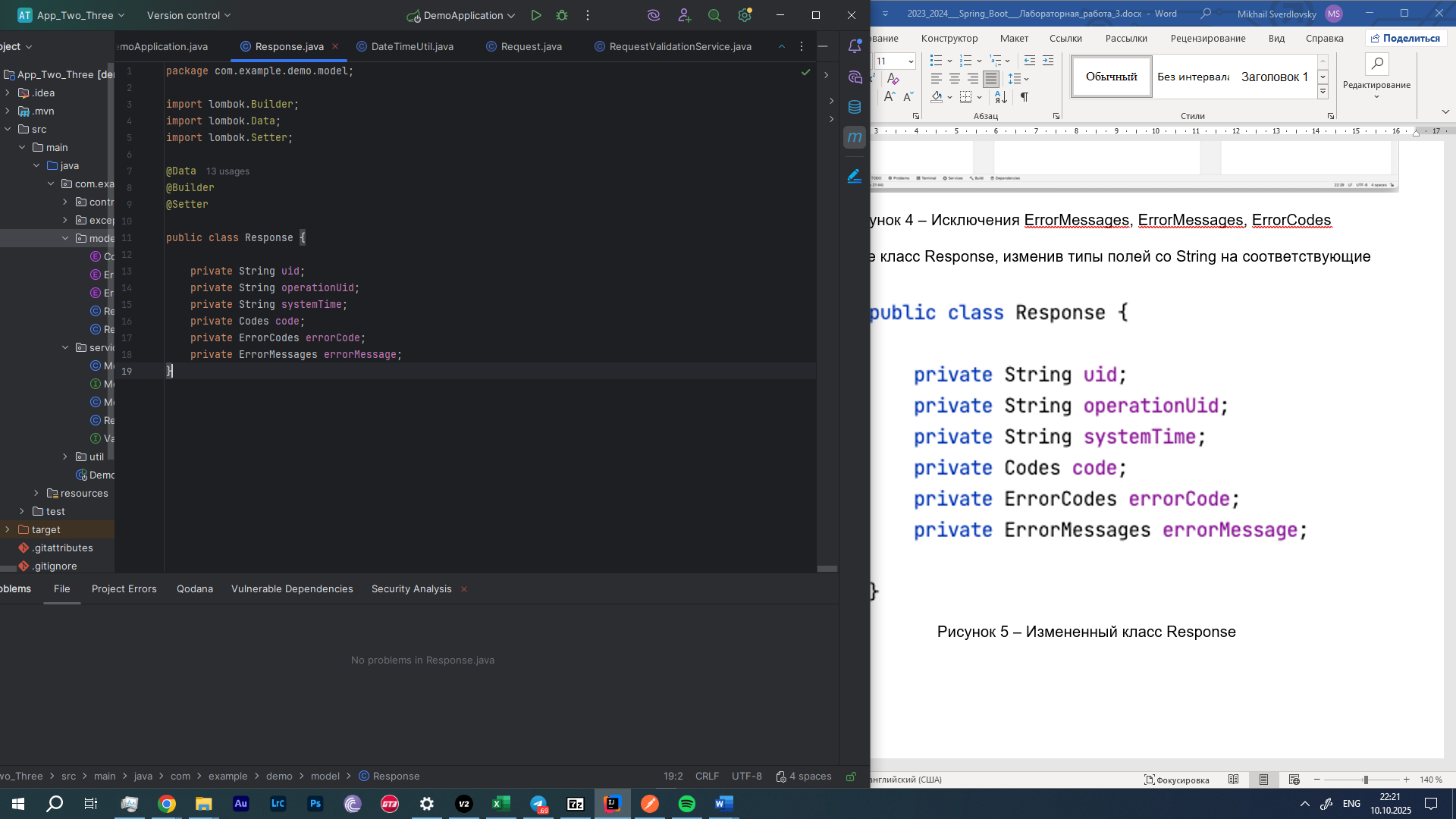


Рисунок 5 – Измененный блок кода класса Response

Также создается класс ValidationFailedException по образцу. Его код реализации показан на рисунке 6.

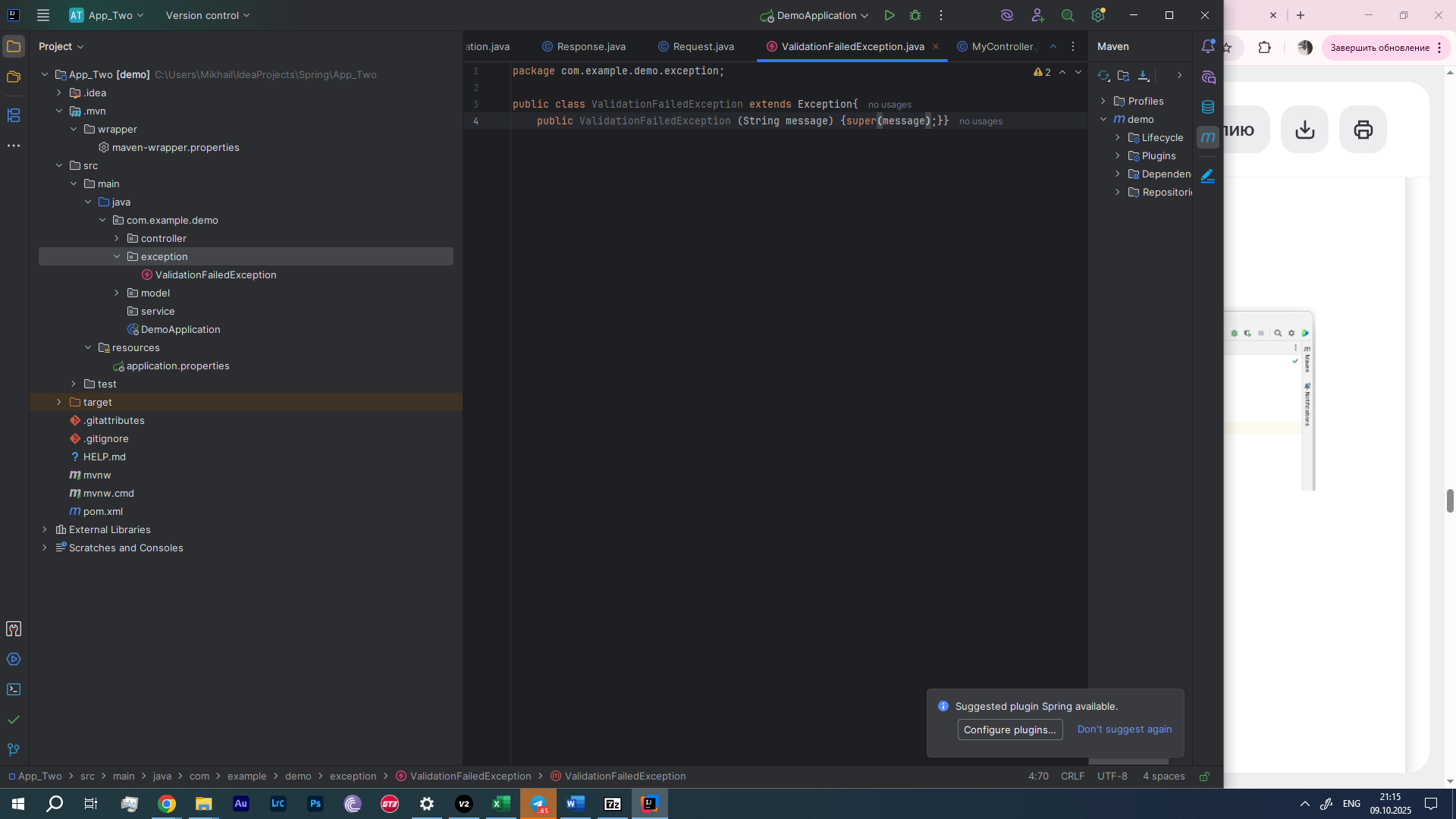


Рисунок 6 – Код реализации класса ValidationFailedException

Дополняем наш класс Request – добавим в него новый метод. Блок кода, дополняющий класс показан на рисунке 7.

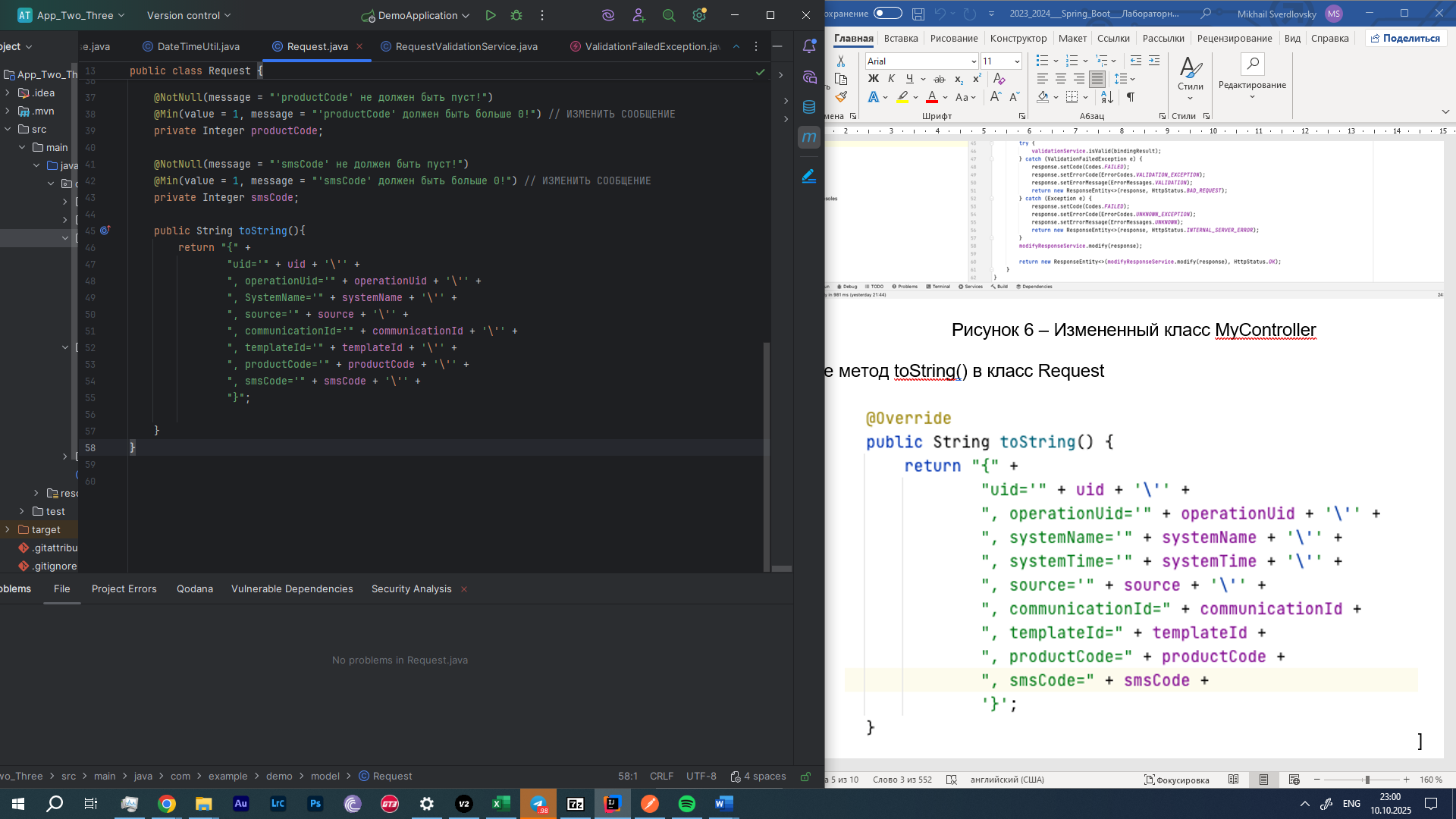


Рисунок 7 – Дополненный Request.class

Добавляем логгирование. Для этого вписываем в файл настройки Spring Boot – application.properties параметры, а также добавляем строку кода в MyController. Настройки показаны на рисунке 8.

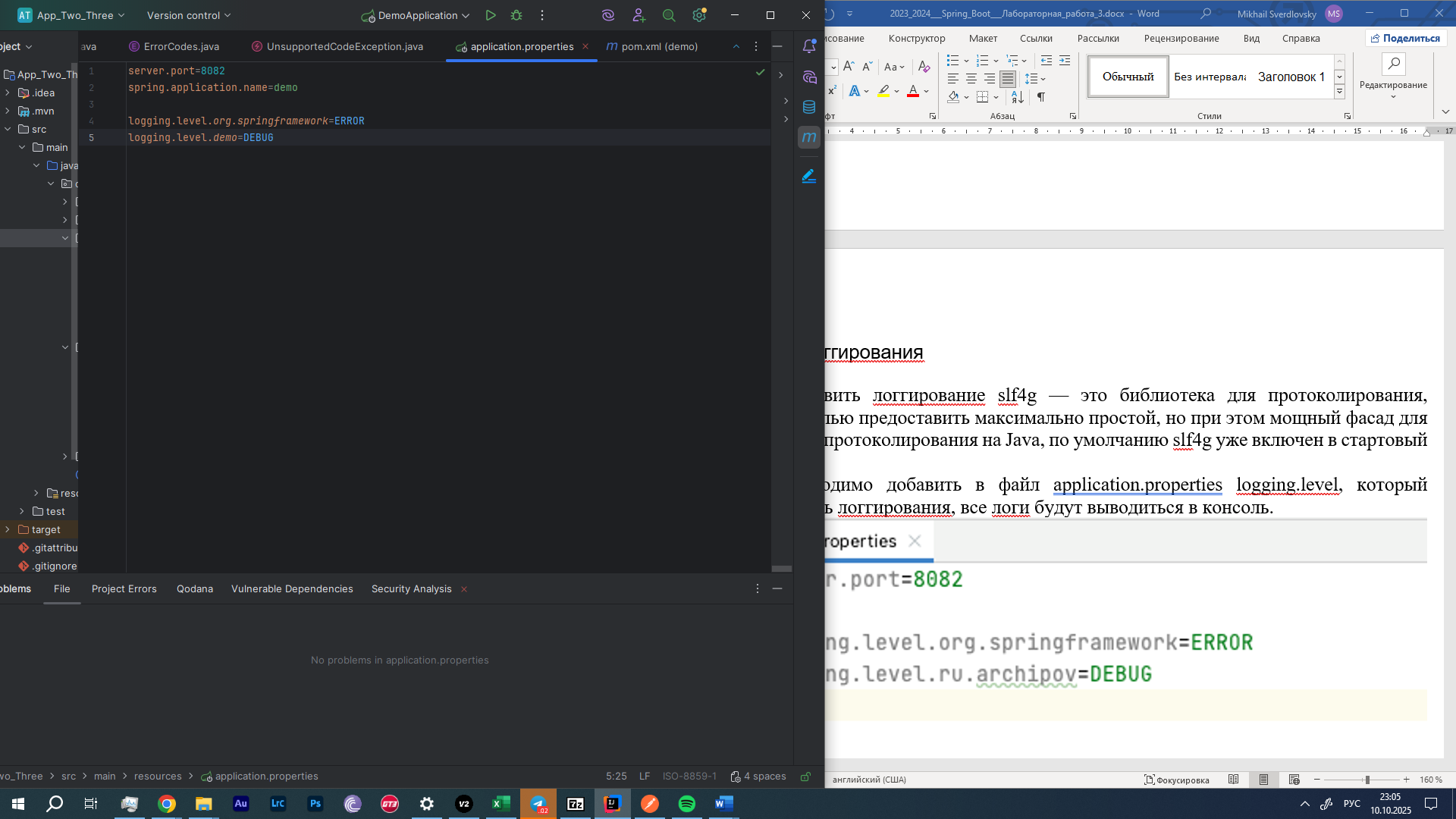


Рисунок 8 – Настройки application.properties

После реализации логирования, делаем один из бинов интерфейса приоритетным (аннотация @Qualifier), и запускаем программу. Логги успешного запуска программы показаны на рисунке 9.

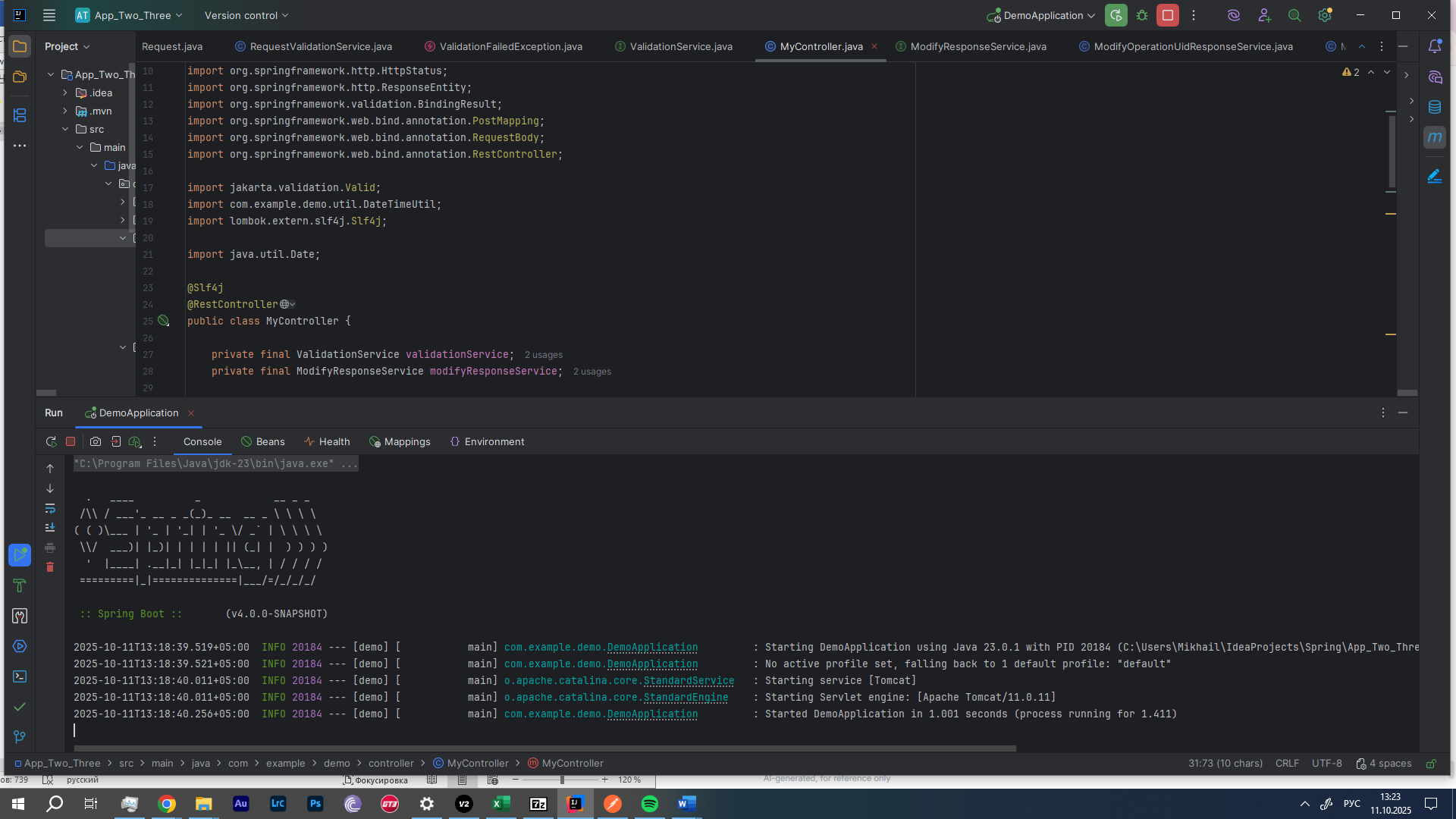


Рисунок 9 – Логгироваение при успешном запуске программы

**Задание 2. Реализация дополнительного функционала.**

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).

Для начала, вместо prntln, создадим реализацию вывода инфоромации о всех процессах через лог. Делается это через log.info. Также, для правильности отображения кириллицы, меняем кодировку выводимого текста в application.properties на UTF-8.

Часть измененного кода класса MyController.java показана на рисунке 10

.

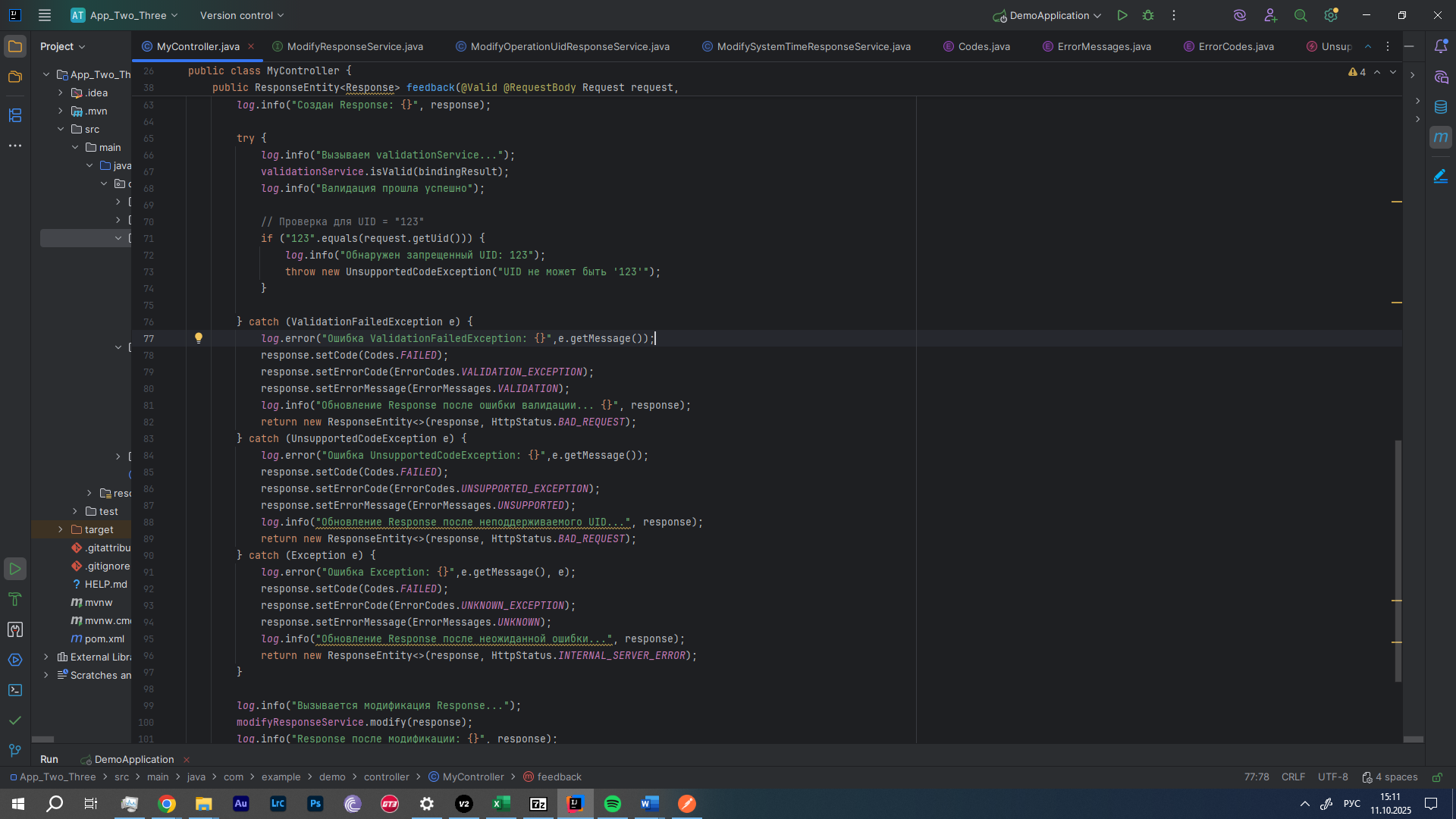


Рисунок 10 – Часть измененного кода класса MyController.java

Процесс логирования при отправке запроса на сервер показан на рисунке 11.

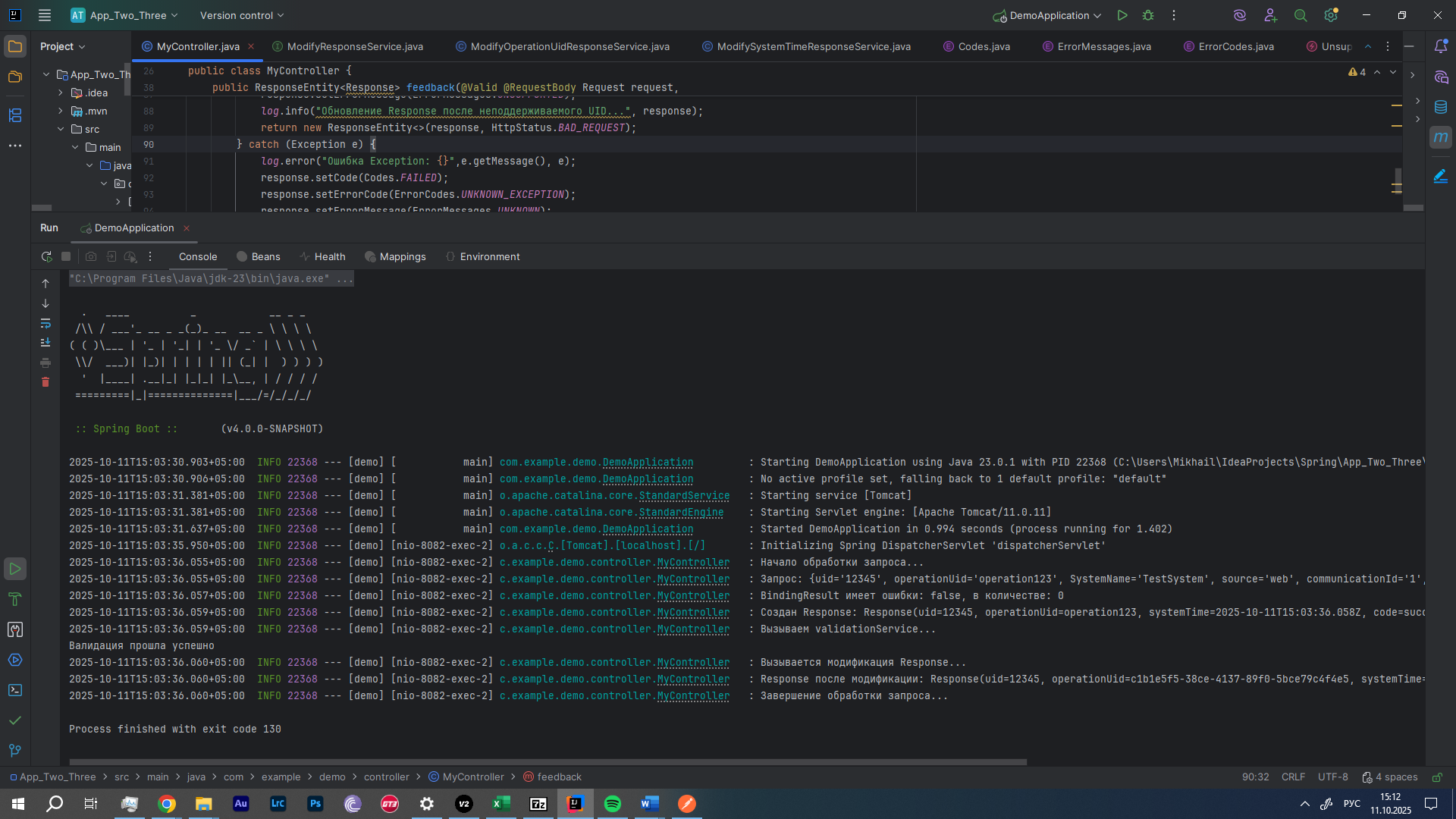


Рисунок 11 – Процесс логирования

Также для тестировки отправляем заведомо неудачные запросы, смотрим на вывод ошибок в лог. Пример лога при отправке таких запросов показан на рисунке 12.

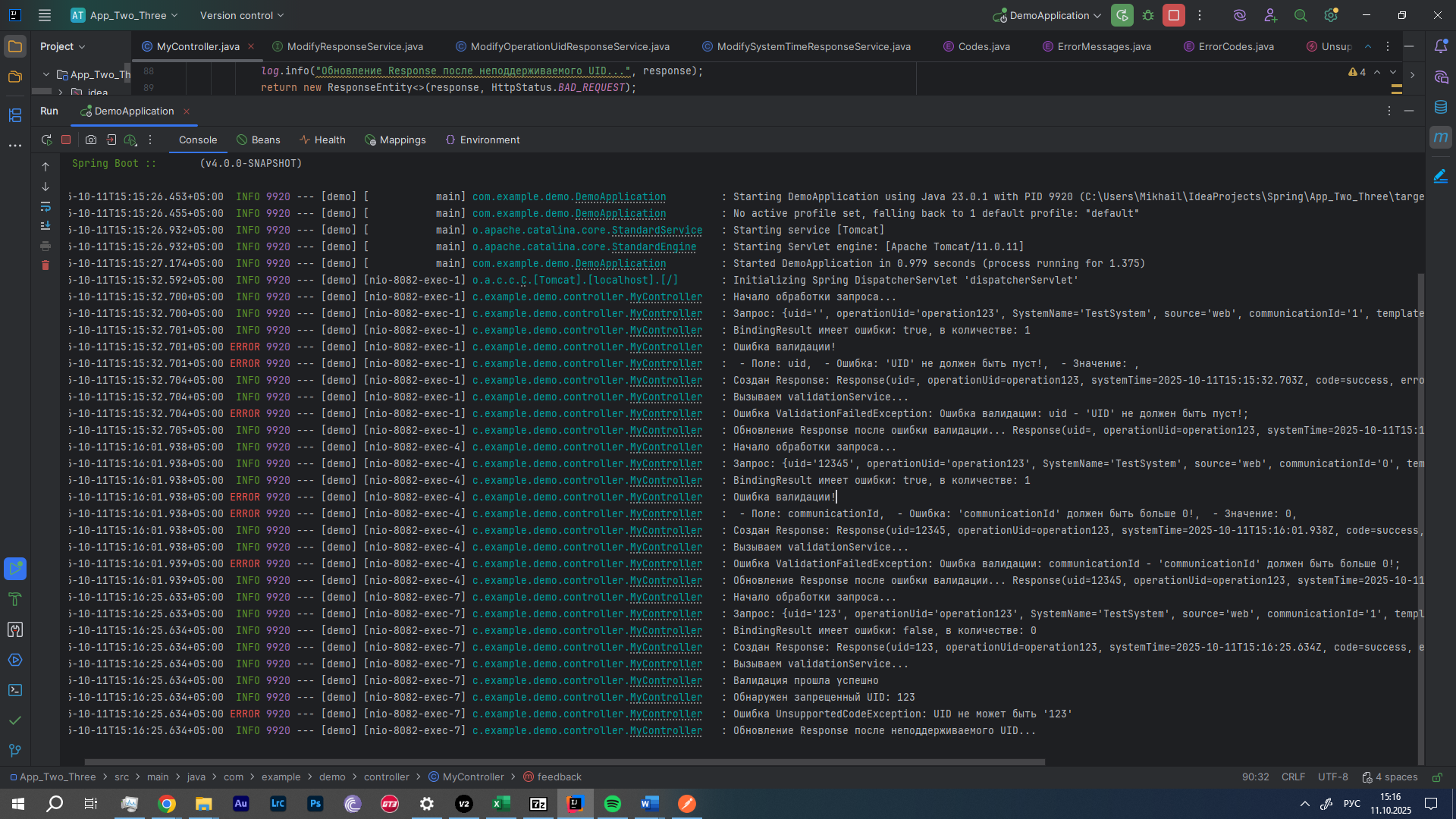


Рисунок 12 – Пример логгов при заведомо неудачных запросах

Согласно заданию, требуется в классе Request изменить тип поля на созданный самим Systems. Для начала, изменим тип поля со String на Systems. В пакете model создаем новый класс – Systems.java. В нем создаем enum – набор констант с именем Systems, и далее перечисляем их. Элементами в наборе будут ERP, CRM, WMS. Код класса Systems представлен на рисунке 13.

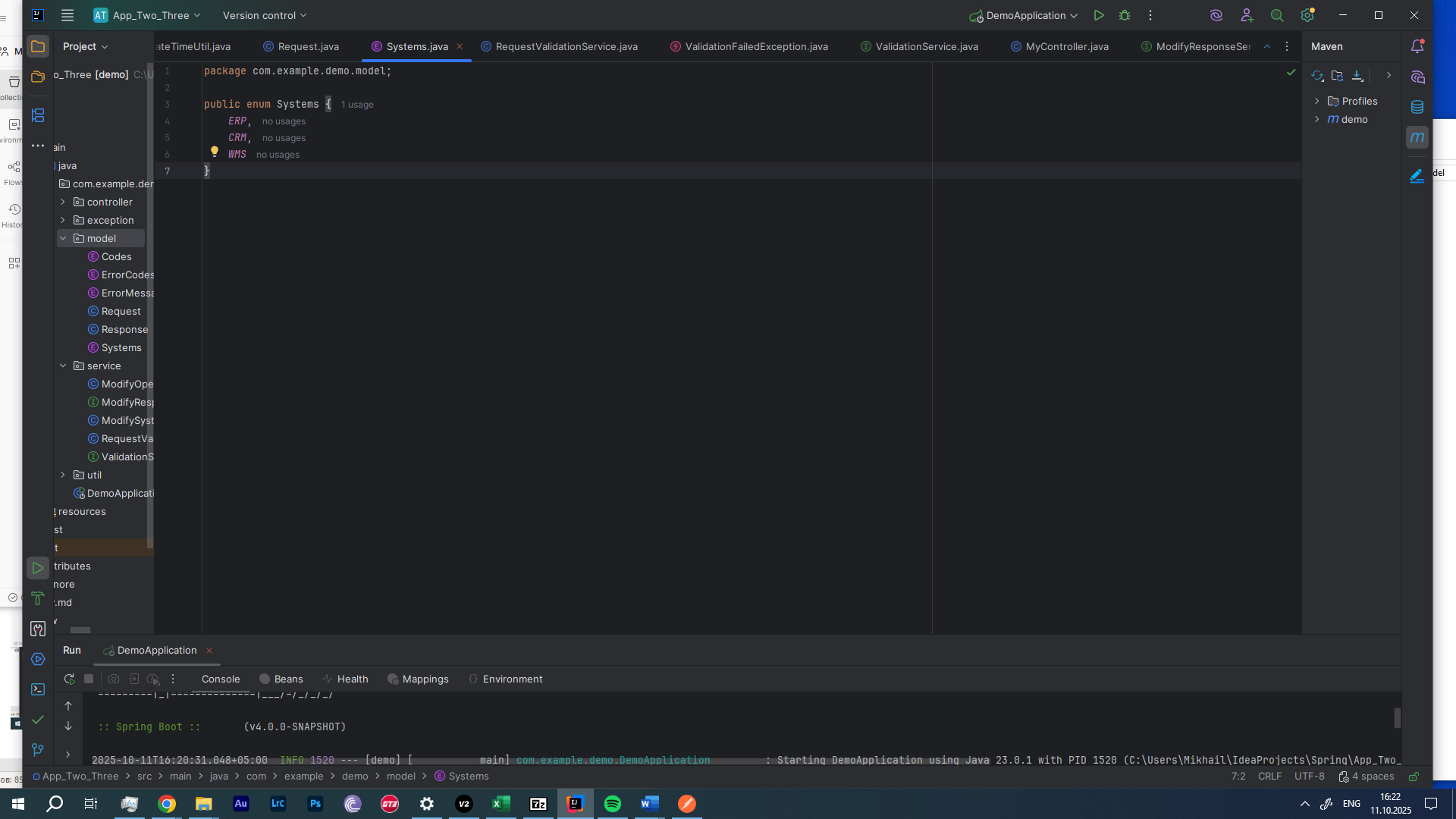


Рисунок 13 – Код класса Systems

Для тестировки через Postman, требуется изменить запросы, что мы и делаем – а, именно, заменяем поле systemName. Процесс тестов показан на рисунке 14.

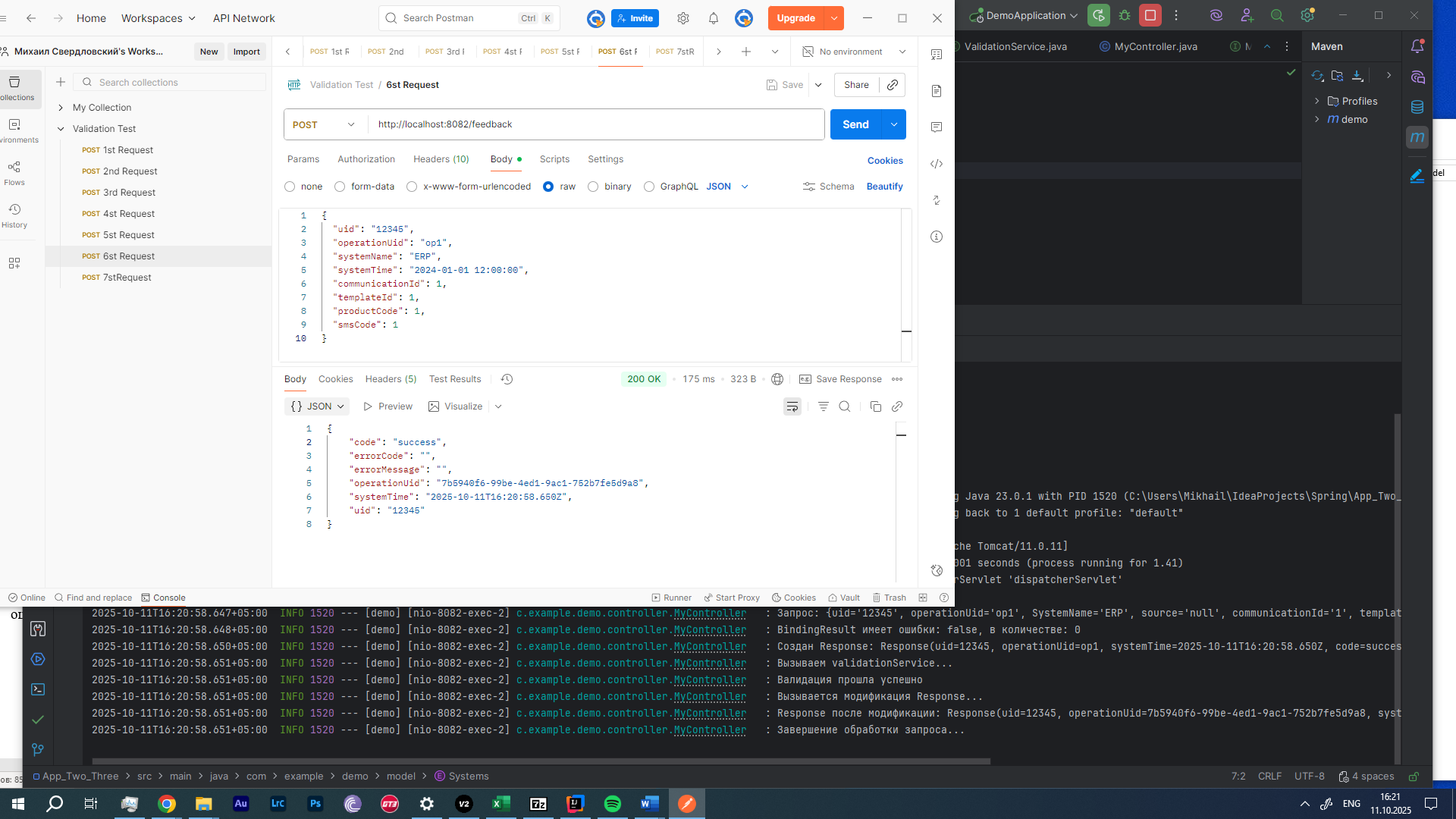


Рисунок 14 – Кейс успешного теста

**Задание 3. Ответы на вопросы.**

1 Что такое ERP система? Приведите примеры.

Это система планирования ресурсов предприятия, которая включает все бизнес-процессы в одну среду (например, закупки, бухгалтерия, аналитика, проекты и пр.)

2 Что такое WMS система? Приведите примеры.

Это система управления складом. Обычно управляет приемками и отгрузом товаров, комплектацией заказов, инвентаризаций и др.

3 Какие уровни логгирования в @Slf4j, в чем их отличие?

а) ошибки – при невозможности выполнить операцию;

б) предупреждения – при некритичных обрабатываемых ошибках;

в) информация – для отслеживания основного хода программы;

г) отладка – для диагностики и выявления проблем в программе;

д) трассировка – для детальной отладки и сложных проблем.

**Вывод**

В результате лабораторной работы был доработан простой REST сервис.