

МИНОБРНАУКИ РОССИИ
федеральное государственное автономное образовательное учреждение
высшего образования
«Санкт-Петербургский политехнический университет Петра Великого»
Институт дополнительного образования
Высшая инженерная школа

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
АВТОМАТИЗАЦИЯ ТЕСТИРОВАНИЯ API ПРОЕКТА “ФИНАНСОВЫЕ
ИЗМЕНЕНИЯ СЧЕТА”
по программе профессиональной переподготовки:
«Разработчик прикладного программного обеспечения (Язык Java)»

Выполнил: Шиленко Николай Николаевич
Руководитель: Маслаков Алексей Павлович

Санкт-Петербург 2022

Обзор предметной области

ФИС – один из сервисов банка для внутренних потребителей.

ФИС получает информацию из автоматизированных банковских систем по счетам юридических лиц, обрабатывает полученную информацию и передает конечным потребителям посредством Kafka и REST API.

Потребители имеют возможность получать из ФИС следующую информацию:

- ▶ проводки;
- ▶ балансовая история;
- ▶ балансовый остаток, сумма и перечень блокировок, аресты, сумма овердрафта;
- ▶ документы на картотеках.

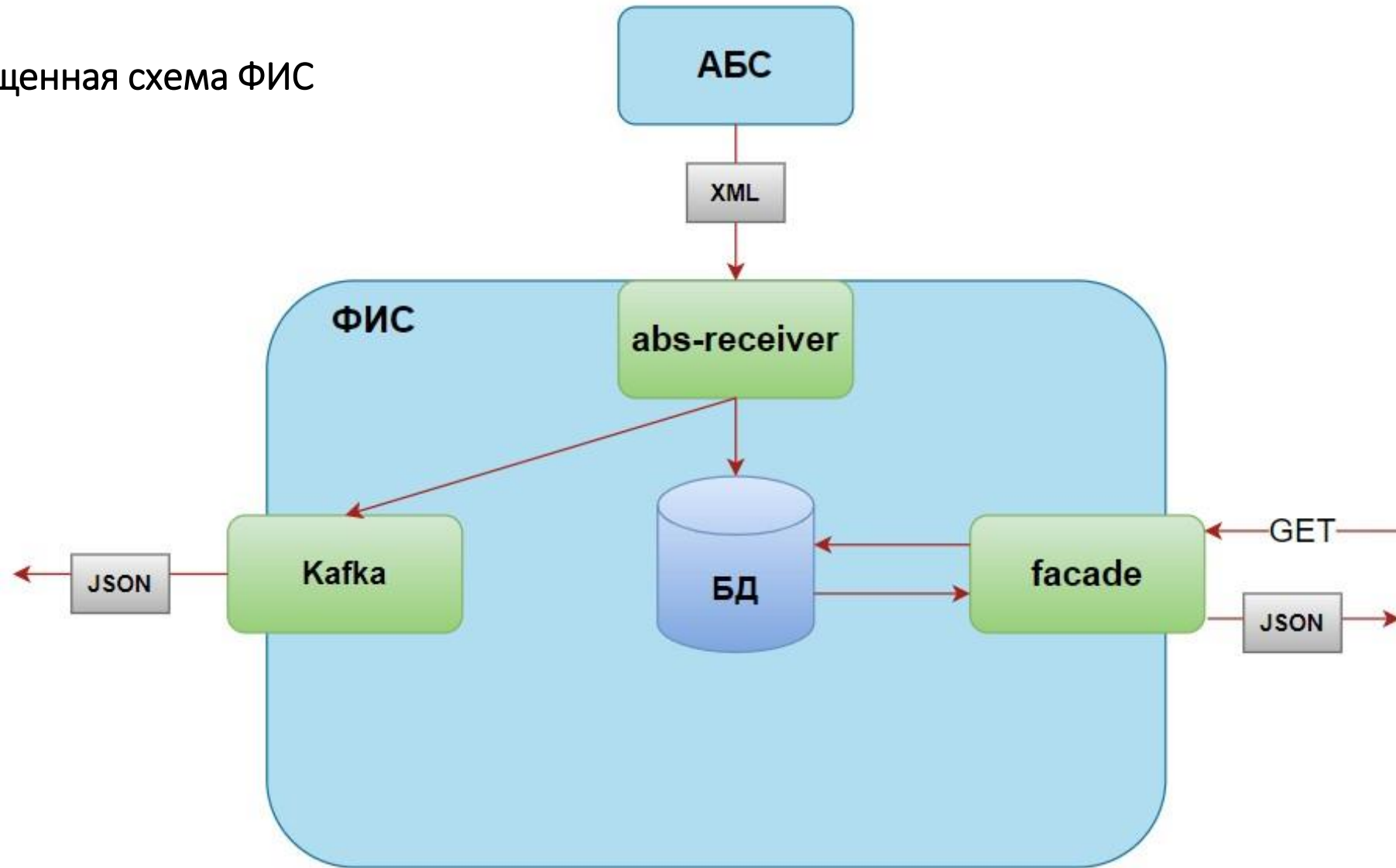
Цель работы

Автоматизация тестирования API ФИС

Задачи

- Изучить архитектуру ФИС
- Выявить требования к проекту автоматизации тестирования
- Составить техническое задание на основе требований
- Выполнить разработку согласно техническому заданию

Упрощенная схема ФИС



Требования

- Формирование документов в формате XML с требуемой структурой, описанной в соответствующих схемах XSD;
- Отправка документов в формате XML посредством HTTP запросов;
- Получение данных в формате JSON посредством HTTP запросов;
- Запуск и выполнение тестов с использованием фреймворка автоматизации тестирования;
- Выполнение тестов на удалённом сервере и локально;
- Обеспечить возможность запуска автоматизированных тестов на различных контурах банка.

Техническое задание

- ▶ Разработать DTO для описания сущности DocStatuses, представляющей проводки по счёту, получаемой из АБС в формате XML. Для разработки классов использовать схемы XSD;
- ▶ Разработать класс, позволяющий производить сериализацию объектов классов DTO сущности DocStatuses в строку в формате XML;
- ▶ Разработать класс, предоставляющий API для инициализации объектов классов DTO сущности DocStatuses, с целью использования его в классах с тестами;
- ▶ Разработать класс, предоставляющий API для выполнения HTTP запросов с методом POST в микросервис abs-receiver для отправки XML;
- ▶ Разработать DTO для описания сущности Movements, представляющей проводки по счёту, получаемые из ФИСа конечными потребителями в формате JSON. Для разработки классов использовать спецификацию OpenApi;
- ▶ Разработать класс, позволяющий производить десериализацию JSON в объекты классов DTO сущности Movements;
- ▶ Разработать класс, предоставляющий API для выполнения HTTP запросов с методом GET в микросервис facade для получения JSON;
- ▶ Разработать классы содержащие тесты, обеспечив возможность их запуска с помощью фреймворка для автоматизации тестирования;
- ▶ Разработать класс, предоставляющий высокоуровневый API для выполнения проверок в классах с тестами.

Lifecycle

- clean
- validate
- compile
- test

```
@XmlJavaTypeAdapter(value = LocalDateAdapter.class)
@XmlAttribute(name = "DocDate")
private LocalDate docDate;
```

deploy

Plugins

- clean (org.apache.maven.plugins:maven-clean-plugin:3.1.0)
- resources (org.apache.maven.plugins:maven-resources-plugin:3.0.2)
- site (org.apache.maven.plugins:maven-site-plugin:3.7.1)
- surefire (org.apache.maven.plugins:maven-surefire-plugin:2.22.1)

```
@JsonDeserialize(using = DateDeserializer.class)
@JsonSerialize(using = DateSerializer.class)
private LocalDate docDateDeadline;
```

```
@Test
public void shouldTurnoverToBeEquals() {
    DocStatusHelper docStatusHelper = new DocStatusHelper()
        .builder()
        .header_Oborot(100.2322222222)
        .initializeDefault()
        .build();
}
```

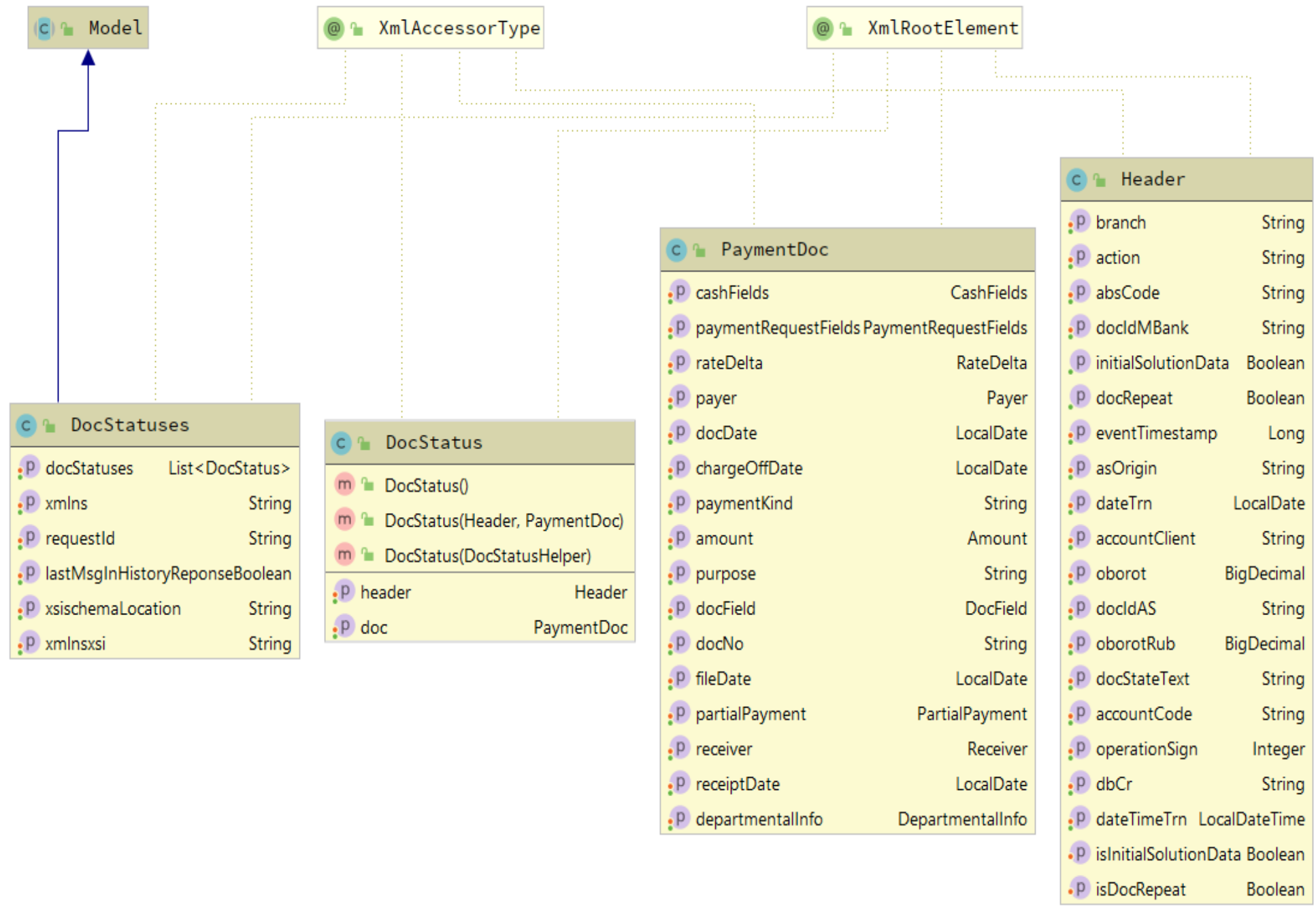
- com.fasterxml.jackson.datatype:jackson-datatype-jsr310:2.6.5
- com.fasterxml.jackson.core:jackson-core:2.6.5 (omitted for conflict with 2.13.0)
- com.fasterxml.jackson.core:jackson-databind:2.6.5 (omitted for conflict with 2.13.0)
- org.assertj:assertj-core:3.22.0
- org.apache.commons:commons-lang3:3.12.0

Используемые ТЕХНОЛОГИИ

- ▶ JDK 1.8
- ▶ RestAssured
- ▶ JAXB
- ▶ Jackson
- ▶ JUnit
- ▶ AssertJ
- ▶ Maven
- ▶ IDE IntelliJ IDEA
- ▶ GIT и GitHub.

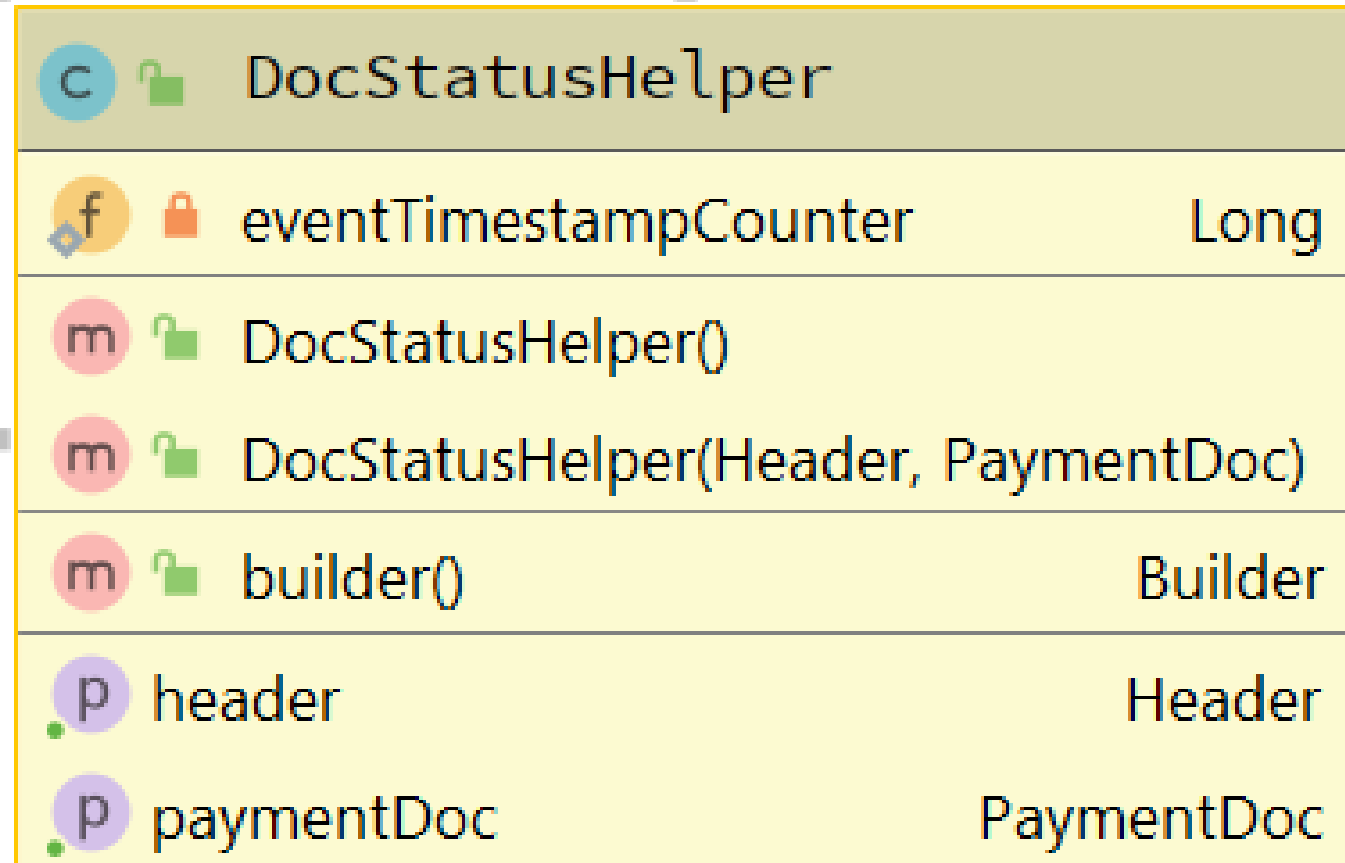
Диаграммы классов

Классы:
DocStatuses,
DocStatus,
Header,
PaymentDoc.



Диаграммы классов

Класс:
DocStatusHelper

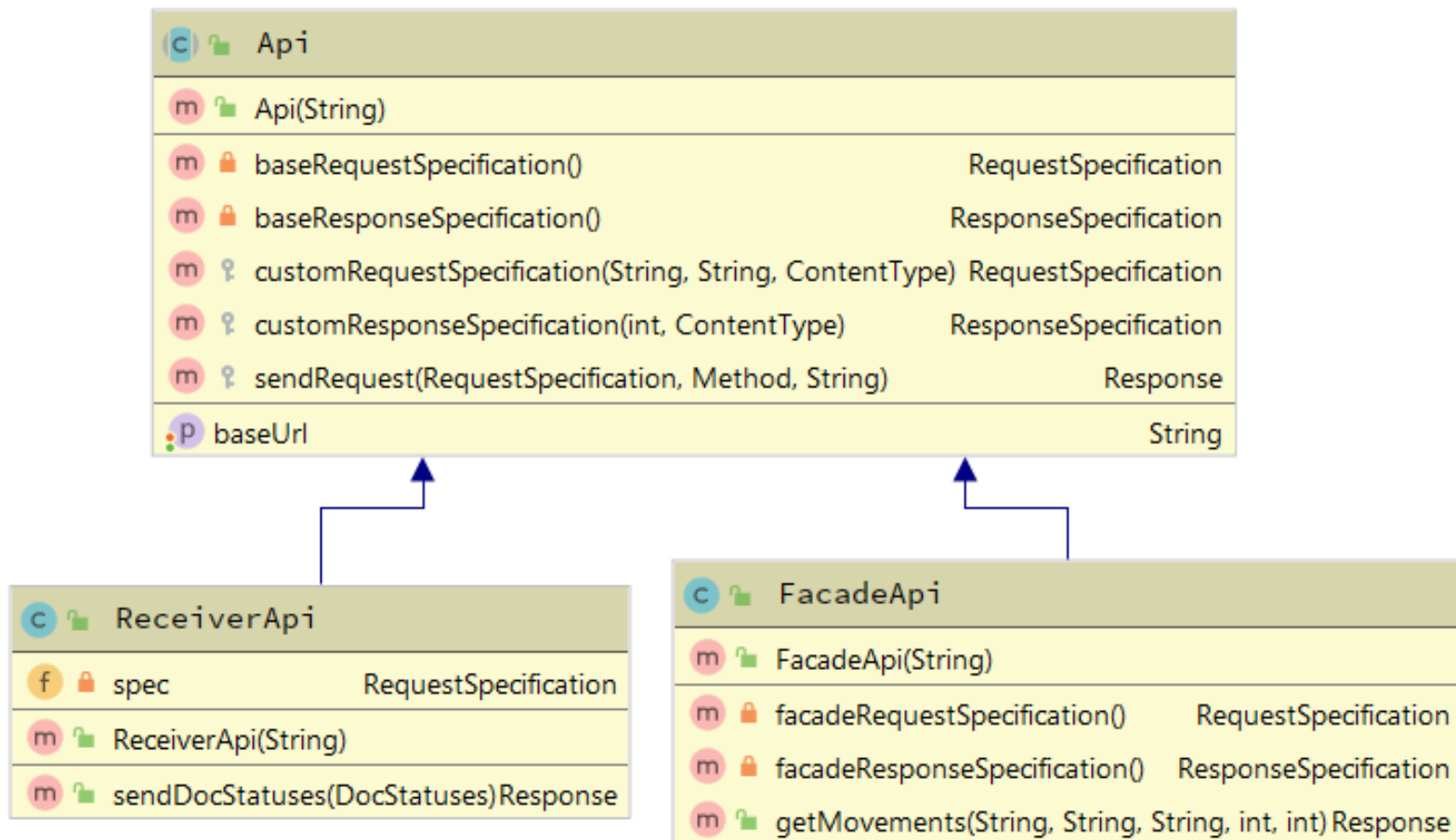


Powered by yFiles

Диаграммы классов

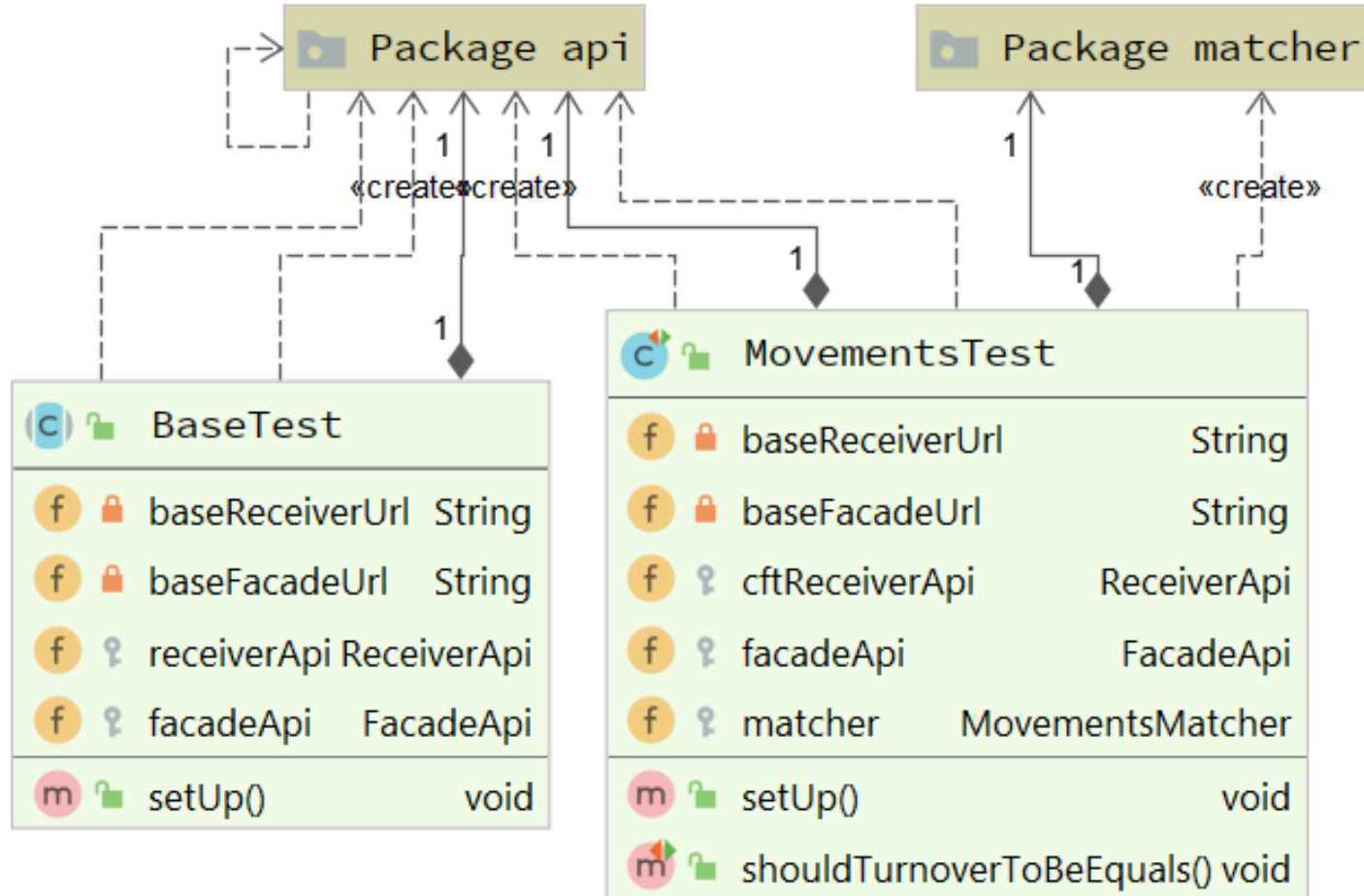
Классы:

Api,
ReceiverApi,
FacadeApi.



Диаграммы классов

Классы:
BaseTest,
MovemetsTest.



Powered by yFiles

Тестирование

Проводилось интеграционное тестирование класса DocStatusHelper позволяющего гибко и удобно инициализировать DTO объекты класса DocStatus для их последующей сериализации в строку в формате XML.

Выполнено:

- ▶ Тестирование методов внутреннего класса Builder используемых для инициализации объектов заданными значениями;
- ▶ Тестирование метода initializeDefault() внутреннего класса Builder используемого для инициализации объектов значениями по умолчанию;
- ▶ Тестирование цепочки вызовов методов используемых для инициализации объектов заданными значениями и метода initializeDefault() используемого для инициализации объектов значениями по умолчанию в различной последовательности;
- ▶ Тестирование сериализации полученных объектов в строку в формате XML со структурой соответствующей схемам XSD. В результате тестирования был выявлен ряд дефектов в работе данного класса. Дефекты были исправлены, что подтверждено повторным выполнением тестов.

Общее количество тестов – 42.

В результате тестирования был выявлен ряд дефектов в работе данного класса. Дефекты были исправлены, что подтверждено повторным выполнением тестов.

Заключение

В ходе работы были получены следующие результаты:

- ▶ Изучена архитектура проекта ФИС;
- ▶ Выявлены требования;
- ▶ Составлено техническое задание;
- ▶ Создан проект автоматизации тестирования, где построена надежная основа для дальнейшего расширения и добавления новых тестов.

Дальнейшее направление работы:

- ▶ Добавление DTO для описания других сущностей проекта ФИС: агрегаты, балансовая история, документы на картотеке;
- ▶ Добавление автоматизированных тестов на основе ручных тест-кейсов, описанных в Системе управления тестированием.

Спасибо за внимание!