

РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ АВТОМАТИЧЕСКОГО ЗАПОЛНЕНИЯ ДОГОВОРОВ ГРАЖДАНСКО-ПРАВОВОГО ХАРАКТЕРА

Руководитель: Алпатов Даниил Геннадьевич
Выполнил: Семенов Дмитрий Алексеевич

Исходный код:
<https://github.com/dmtr-smnv/thesis.git>

Проблема

- Многие преподаватели Высшей инженерной школы являются внештатными сотрудниками
- Трудовые отношения с такими сотрудниками должны ежемесячно подтверждаться договором с указанием отработанного времени

Цель

- Автоматизировать процесс заполнения договоров на основе расписания и персональных данных внештатных сотрудников

Актуальное состояние

- На сервере Высшей инженерной школы реализован REST API, обрабатывающий 43 варианта GET-запросов и отдающий информацию в формате JSON
- Сервис отдает как одиночные объекты, так и массивы объектов в зависимости от запроса

Задачи

- Реализовать получение JSON-объектов от REST API Высшей инженерной школы
- Произвести необходимую обработку
- Реализовать заполнение договоров на основе полученных сведений

Программная реализация должна включать следующие компоненты

- HTTP-клиент для взаимодействия с REST API
- Преобразователь JSON-объектов в объекты предметной области: преподавателей и расписания
- Редактор документов формата Microsoft Word

Программная реализация основана на следующих библиотеках

- HTTP-клиент – на базе библиотеки Apache HttpClient
- Преобразователь JSON-файлов – на базе библиотеки Google GSON
- Редактор документов – на базе библиотеки Apache POI

Использованные технологии



Получение JSON-объектов

- Для функционирования приложения требуется получить два JSON-объекта: преподаватель и расписание
- JSON-объекты имеют следующую структуру:

{

«ключ»: значение,

«ключ»: значение

}

Получаемые JSON-объекты

- Объект преподавателя получается по запросу:

`https://api.avalon.ru/api/Teachers/{ID}`

- Объект расписания получается по запросу:

`https://api.avalon.ru/api/Schedules?startDate={startDate}&endDate={endDate}`

Объект преподавателя и расписания

- Объект преподавателя включает в себя 40 полей, из которых в работе приложения используются 5
- Объект расписания включает в себя 19 полей, из которых в работе приложения используются 9

Работа с JSON-объектами

- Преобразование полученных JSON-объектов в объекты предметной области «1-в-1» нецелесообразно, т.к. в этом случае сведения, содержащиеся в полях объектов предметной области были бы чрезмерными (не применимыми в данной программной реализации)
- Следовательно, необходима предварительная корректировка JSON-объектов
- Редактирование JSON-объектов производится через промежуточное преобразование в массив строк с копированием нужного количества параметров и последующим преобразованием в коллекции объектов предметной области

Работа с коллекцией преподавателей

- Исключение штатных преподавателей
- Исключение тех внештатных преподавателей, которые не работали в заданном временном интервале
- Расчет и инициализация поля отработанного времени на основе коллекции расписания

Работа с файлами договоров

- Замена текста производится посредством поиска маркера с его заменой имеющимися данными
- Стандартный вид маркера для замена, например, имени имеет следующий вид:

`[%USERNAME%]`

Общий алгоритм работы приложения

- Получения от пользователя граничных значений временного интервала в виде двух параметров запуска вида «ГГГГ-ММ-ДД»
- Получение от REST API перечня всех преподавателей, а также массива объектов расписания за данных период
- Исключение избыточных параметров из JSON-объектов
- Преобразование откорректированных JSON-объектов в коллекции объектов предметной области
- Обработка коллекции преподавателей с целью оставить в ней только внештатных преподавателей, отработавших в заданном интервале
- Подготовка перечня договоров для каждого объекта коллекции преподавателей путем замены маркерного значения на информацию, содержащуюся в полях объекта

UML-диаграмма классов



Тестирование программного продукта

- Тестирование производилось на основе подготовленного плана тестирования
- Было реализовано 7 тестовых сценариев (все 7 – позитивные)
- Тестовые сценарии охватывали вопросы пользовательского ввода (*классы эквивалентности – корректные и некорректные значения, корректный и некорректный порядок ввода значений*) и редактирования шаблона договора (*поиска и замены маркерного значения*)
- Тестирование показало соответствие разработанного программного продукта требованиям в части работы с пользовательским вводом и подготовки перечня договоров