УДК 004.45

МОДЕЛИРОВАНИЕ И СОЗДАНИЕ ВИРТУАЛЬНЫХ МОДЕЛЕЙ ОБЪЕКТОВ ЖЕЛЕЗНОДОРОЖНОГО ПУТИ НА ЯЗЫКЕ JAVA

А. А. Абрамов, В. Г. Кобак
Донской государственный технический университет (г. Ростов-на-Дону, Российская Федерация)

В области управления железнодорожными составами большие происходит постепенное планомерное снижение доли участия человека. В то же время последовательность действий в данной отрасли при обнаружении неисправностей и дальнейшие действия имеют строго определенный порядок. В статье указывается основной подход к описанию полигона данных и правил работы с железнодорожными объектами. Результат работы реализован на языке Java с использованием фреймворка Spring.
Ключевые слова: Java, Apache Maven, полигон, правило, Jackson, Spring framework.

UDC 004.45

SIMULATION AND CREATION OF VIRTUAL MODELS OF RAILWAY TRACK OBJECTS IN JAVA

A. A. Abramov, V. G. Kobak
Don State Technical University (Rostov-on-Don, Russian Federation)

The share of human participation in the field of train management is gradually and systematically declining. The rules for detecting faults, as well as correct work, have a strict procedure. The article shows the main approach to the description of the polygon data and the rules of work with objects. The result of the work is implemented in Java using Spring framework.
Keywords: Java, Apache Maven, polygon, rule, Jackson, Spring framework.

Введение. Железнодорожный транспорт — это многоотраслевое хозяйство, представляющее собой огромный по протяженности конвейер, бесперебойная и безаварийная работа которого зависит от функционирования каждой из его составных частей. Железнодорожный транспорт общего пользования включает в себя железнодорожные пути, устройства электроснабжения, сети связи, системы сигнализации, централизации и блокировки, сооружения, оборудование, парк подвижного состава (локомотивы, вагоны), раздельные пункты, станции и узлы и др. [1].

Актуальность данной работы обусловлена тем, что в сети Интернет нет ни одного решения по формализации правил обработки данных, касающихся ситуаций, возникающих на железнодорожных участках. Возложенная на человека ответственность по контролю должна плавно переходить в зону контроля вычислительной техники.

Цель работы состоит в том, чтобы реализовать возможность контроля за железнодорожными объектами на основании тех правил, которые выставляет заказчик, зачастую не имеющий знаний в области программирования.

В рамках статьи решались следующие задачи:

- выбор языка Java и системы сборки проектов Maven;
- описание виртуальных объектов железнодорожной сети;
- описание синтаксиса и правил;

• выбор структуры приложения.

Выбор языка Java и система сборки проектов Maven

Выбор языка на этапе создания проекта является важным условием дальнейшего успеха конечного продукта. Каждый язык программирования представляет собой инструмент, имеющий наравне с преимуществами и свои недостатки. Ключевыми требованиями стали: возможность поддержки и развития проекта на десятки лет вперед, более низкий порог вхождения в целевую архитектуру новых разработчиков и широкий инструментарий по автоматизации выявления потенциальных ошибок в программном коде на этапе создания программного средства. Поставленным целям и задачам язык Java полностью удовлетворяет.

Язык Java, согласно рейтингу TIOBE, на март 2019 является самым популярным языком программирования [2]. Такая популярность на коммерческом рынке обеспечивается значительным числом факторов: большим количеством готовых решений, удобным сопровождением крупных проектов, наличием garbage collector, хорошо проработанной многопоточностью и кроссплатформенностью JVM. Создание любых проектов не обходится без системы сборки проектов.

Apache Maven — фреймворк для автоматизации сборки проектов на основе описания их структуры в файлах на языке POM, являющемся подмножеством XML [3].

В файлах описания проекта содержится его спецификация, а не отдельные команды выполнения. Все задачи по обработке файлов, описанные в спецификации, Maven выполняет при помощи последовательности встроенных и внешних плагинов. На рис. 1 показана диаграмма использования систем сборки на коммерческом рынке [4].

Основополагающим принципом выбора стали следующие факторы:

• стандартизация и упрощение процесса сборки;

• все зависимости автоматически записываются в скрипты Maven;

• загрузка зависимостей происходит автоматически;

• легкая интеграция в существующие IDE.

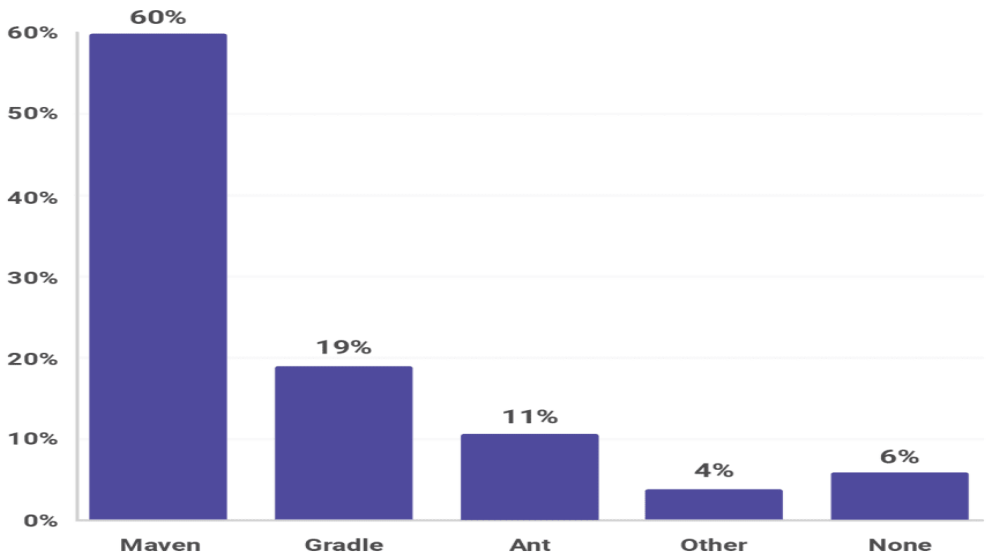


Рис. 1. Доля использования систем сборки на рынке в 2018г.

Виртуальные объекты железнодорожной сети, области и их описание.

В качестве виртуальных объектов системы железнодорожного транспорта выделены такие, как [5]:

1) светофор (Sv);
2) переезд (Pr);
3) путь (Pt);
4) стрелка (St) [6];
5) участок приближения/ удаления (Up);
6) маркер (Mr);
7) завершающий объект (Ls).

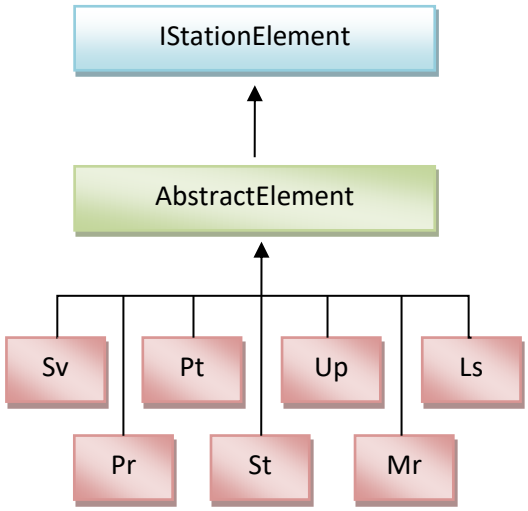


Рис. 2. Иерархия объектов

Виртуальные объекты с 1 по 5 пункт являются отображением реального физического элемента или комплекса элементов. Маркер служит для информирования дежурного или диспетчера о состоянии области контроля. Завершающий объект присутствует на всех областях и играет роль соседа объекта без связи в четном или нечетном направлении.

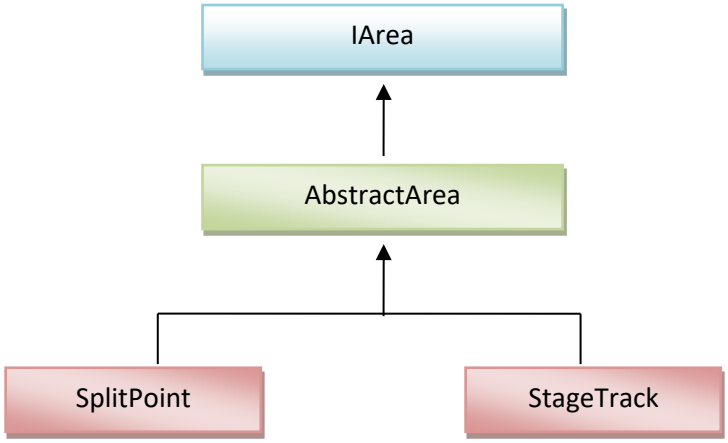


Рис. 3. Иерархия областей

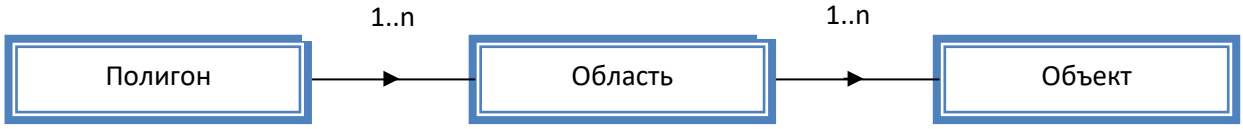
Каждый объект обязан хранить уникальный код в той области, к которой он принадлежит. Уникальность области на полигоне обеспечивается кодом и шестизначным номером единой сетевой разметки. Выделим следующие непересекающиеся области [6]: раздельный пункт (SplitPoint) и путь перегона (StageTrack).

Описание полигона представлено в виде тэгов вложенности (xml файл).

```
<polygon>
    <area code ="" esr="" type="">
        <pt/sv/up/pr/mr code="" evenLink="" oddLink="" evenArea="" oddArea=""/>
        <st code="" evenLinkP="" evenLinkM="" oddLinkP="" oddLinkM=""/>
        <party codeFirst="" codeSecond="" />
    </area>
</polygon>
```

Для разбора структуры данных и создания объектов модели использована библиотека Jackson [7]. Каждый класс — POJO (Plain Old Java Object) класс.

Структурно можно описать получившийся объект — «полигон» — следующим образом:



Описание, синтаксис и работа правил

При работе с железнодорожными объектами необходимо придерживаться строгих правил функционирования системы. Каждый участок индивидуален и требования, выставляемые заказчиком, могут варьироваться. Чтобы проводить индивидуальную настройку, было выработано понятие «правило» (Rule). Каждое правило обладает свойствами:

1) имеет предусловие возникновения и постусловие выполнения;
2) объект исследования, к которому применимо правило, уникален;
3) каждое правило — функция.

Созданы функциональные правила проверки:

1) И (And);
2) ИЛИ (Or);
3) ЕСЛИ…ТО... ИНАЧЕ (If);
4) Конечное (Simple).

Каждое правило имплементирует функциональный интерфейс IFunction с функцией проверки check(). Описание правил представлено в виде тэгов вложенности (xml файл) со следующей структурой:

```
<rules>
    <rule name="" type"" area="">
        <group>
            <element code=""/>
        </group>
        <precodition>
            | function |
        </precondition>
        <postcodition>
            | function |
        </postcondition>
    </rule>
</rules>
```

Группа объектов внутри | functions | описывается рекурсивным образом:

<function> ::= <if> | <and> | <or> | <print> | <check> | <timer> | <setstate>

<print>, <check>, <timer>, <setstate> ::= nil

<if> ::= <condition> & <then> | <condition> & <then> & <else>
<or>, <and> ::= (<condition>)$^{+}$
<condition>, <then>, <else> ::= <function>

Для считывания данных используется библиотека Jackson с реализацией POJO классов, характеризующих структуру вложенности тэгов по принципу, описанному выше. Стоит отметить, что наличие тэга *condition* не избыточно, поскольку он направлен на соблюдение строгого порядка выполнения операций и избавляет от излишнего расширения атрибутов тэга, специфичных для конкретной функции.

Также стоит отметить, что формирование правил для конкретного объекта производится только после того, как был создан полигон данных. Каждый объект в теге *group* автоматически применяет описанные в правиле условия относительно самого себя.

Каждое такое правило обеспечивает условие безопасности ГОСТ 33 896–2016 Логический анализ эксплуатационных событий и действий персонала [8].

Структура приложения

Приложение создается внутри Spring-контекста. Внутри главной функции производится запуск Spring-приложения с аннотацией @SpringBootApplication [10]. Настройка @ComponentScan приводит к поиску всех конфигураций для старта. В пакет поиска встраиваются такие бины, как ISateSender, IPlygon, List<IRule>, IMessageHolder, что отображено на рис. 4.

Блок IN/OUT — внешний контроллер, который производит обработку электрических сигналов, приходящих с полигона. Приходящий сигнал по входному каналу информирует IPolygon об изменении состояния объекта. Изменение состояния фиксируется. Далее метод информирует всех слушателей (подписантов, реализующих интерфейс IStateListener) для последующей обработки результата.

Объект Rule на этапе создания правила подписывается и относительно переданного объекта проверяет предусловие выполнения. В случае, если описание <precondition> выполнилось успешно, задача по постобработке <postcondition> перенаправляется в пул потоков. Логика обработки функции с тегом <setstate> специфична тем, что происходит запрос на изменение состояния в контроллер посредством делегирования сообщения объекту IStateSender.

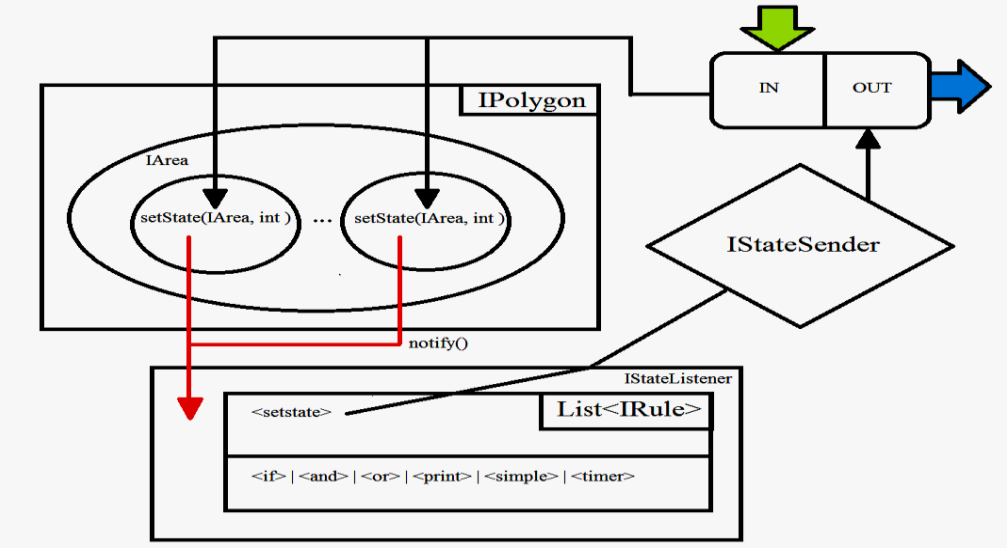


Рис. 4. Назначение бинов и работа приложения

Заключение. Таким образом, полученная структура данных, описанные правила позволяют настроить участок, указав связи между объектами как одной области, так и смежных областей; сформулировать правила работы участка без знаний в области программирования. Приложение имеет возможность гибкого расширения функционала за счет сформированной логики работы, построенной на интерфейсах, а не на реализации. Собранный проект можно с легкостью установить на множество известных операционных систем для которых существует JVM.

Библиографический список

1. Железнодорожный транспорт [Электронный ресурс] / Российские железные дороги. — Режим доступа : http://rzd.company/index.php/ Железнодорожный_транспорт (дата обращения : 19.03.2019).

2. TIOBE Index for January 2020 [Электронный ресурс] / TIOBE. — Режим доступа : https://tiobe.com/tiobe-index/ (дата обращения : 17.03.2019).

3. Apache Software Foundation [Электронный ресурс] / Apache Maven Project. — Режим доступа : http://maven.apache.org/ (дата обращения : 20.03.2019).

4. JVM Ecosystem report 2018 – About your Tools [Электронный ресурс]/ SNYK.IO. — Режим доступа : https://snyk.io/blog/jvm-ecosystem-report-2018-tools (дата обращения : 21.03.2019).

5. О железнодорожном транспорте в Российской Федерации : федер. закон от 10 января 2003 года №17-ФЗ [Электронный ресурс] / Режим доступа: http://www.consultant.ru/document/cons_doc_LAW_40443/4ceedc6beeab98acfcffe6b042e41a8319e1c922/ (дата обращения : 24.03.2019).

6. Раздельный пункт [Электронный ресурс] / Википедия. — Режим доступа : https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B7%D0%B4%D0%B5%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D0%BF%D1%83%D0%BD%D0%BA%D1%82 (дата обращения : 21.03.2019).

7. Jackson [Электронный ресурс] / Gutlib.com. — Режим доступа : https://github.com/FasterXML/jackson (дата обращения : 20.03.2019).

8. ГОСТ 33 896–2016. Системы диспетчерской централизации и диспетчерского контроля движения поездов. Требования безопасности и методы контроля [Электронный ресурс] / Электронный фонд правовой и нормативно-технической информации. — Режим доступа : http://docs.cntd.ru/ document/1200144933 (дата обращения : 20.03.2019).

Об авторах:

Кобак Валерий Григорьевич, профессор кафедры «Программное обеспечение вычислительной техники и автоматизированных систем» Донского государственного технического университета (344000, РФ, г. Ростов-на-Дону, пл. Гагарина, 1), доктор технических наук, valera33305@mail.ru

Абрамов Анатолий Анатольевич, магистрант Донского государственного технического университета (344000, РФ, г. Ростов-на-Дону, пл. Гагарина, 1), tolik.abramoff@yandex.ru