

РЕАЛИЗАЦИЯ АЛГОРИТМА МАППИНГА ОБЪЕКТОВ ПЕРЕДАЧИ ДАННЫХ В БАЗОВЫЕ СУЩНОСТИ ПРОГРАММЫ

Медведская Ю.И.

Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь

Научный руководитель: Горбач А.П.. – магистр техн. наук, ст. преподаватель кафедры ПИКС

Аннотация. Алгоритм реализации маппинга объектов программы представляет собой создание абстрактных классов, реализующих автоматическое преобразование объекта одного типа к другому, избегая однотипного кода в серверной части программного средства.

Ключевые слова: маппинг, объекты передачи данных, java, абстрактный класс, программное средство

Введение. Маппинг данных – процесс сопоставления полей данных (определенных элементов источника или всего источника) и связанных с ними полей данных в другом месте назначения. То есть это установление соотношения между моделями данных, которые находятся в разных источниках или системах. Программное обеспечение и инструменты маппинга данных автоматически сопоставляют поля данных из одного источника данных в другой [1].

Основная часть. В ходе разработки программного средства по подбору и оценке автомобиля с пробегом была выявлена проблема сопоставления полей данных классов, выполняющих различные задачи. Существуют готовые решения по маппингу данных. В Java зависимость `modelmapper` предоставляет интерфейс, который позволяет не реализовывать маппинг вручную, однако использование большого количества импортируемых дополнительных зависимостей в разрабатываемое программное средство может привести к внутренним конфликтам, а также замедлить работу программы.

В разработанном программном средстве реализации онлайн-сервиса по подбору и оценке автомобиля были использованы три типа сущностей:

- сущности базы данных;
- сущности запросов;
- сущности ответов.

Сущности запросов и ответов относятся к объектам, используемых для получения данных из запроса и отправки ответа клиенту, который обращается к разработанному API. Такие объекты относятся к типу *DTO* – *Data Transfer Objects*. *DTO* называют *value-object* на стороне сервера, который хранит данные, используемые в слое представления. В случае фреймворка *Spring* такие объекты автоматически сериализуются и десериализуются [2].

Первым этапом реализации автоматического преобразования объектов из одного типа в другой было создание иерархии наследования классов, которая позволит сделать интерфейс маппера. Для объектов сущностей был создан абстрактный класс *AbstractEntity*, для *DTO* – *AbstractDTO*. Наследниками последнего класса являются так же абстрактные классы, которые уже определяют принадлежность к объекту запроса или ответа: *AbstractReq* и *AbstractResp*.

Следующим этом является создание интерфейса для преобразования объектов базовых сущностей в объекты типа *AbstractDTO*. Содержание такого интерфейса находится ниже:

```

public interface SimpleObjectMapper<E extends AbstractEntity, D extends AbstractDTO> {
    E toEntity(final D d);
    D toDto(final E e);
    List<D> toDtoList(final List<E> eList);
    Set<E> toEntitySet(Set<D> dtoSet);
}

```

Созданный интерфейс реализует абстрактный класс *SimpleAbstractMapper* с обобщёнными классами, который называются дженерики. Дженерики – особые средства языка *Java* для реализации обобщённого программирования: особого подхода к описанию данных и алгоритмов, позволяющего работать с различными типами данных без изменения их описания [3].

Созданный интерфейс и абстрактные классы реализуют преобразования только между обобщёнными типом *DTO* и базовыми сущностями разработанного программного средства. Для создания маппера между тремя типами сущностей были так же разработаны интерфейс и абстрактный класс. Интерфейс *FullAbstractMapper* наследует *SimpleObjectMapper*, а абстрактный класс реализует уже этот интерфейс и наследует реализацию *SimpleAbstractMapper*. На рисунке 1 представлена диаграмма классов, которая наглядно показывает систему наследования и использования классов.

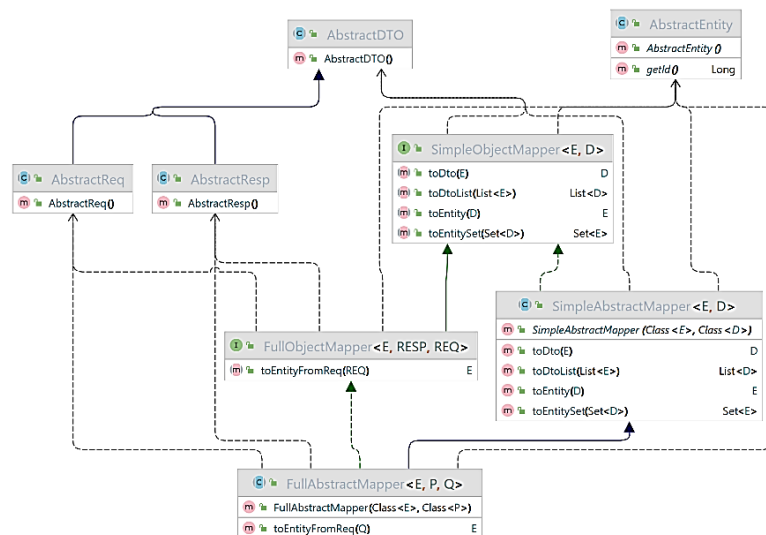


Рисунок 1 – Диаграмма классов

Одной из основных сущностей программного средства по оценке и подбору автомобилей с пробегом является сущность «Заказ на подбор». Для этой сущности существует три класса:

- *InspectionOrder* – базовая сущность заказа, связанная с таблицей в базе данных;
- *InspectionOrderResp* – сущность, отправляемая как ответ сервера;
- *InspectionOrderReq* – сущность, получаемая как запрос к серверу [4].

Для создания маппера для этих сущностей необходимо создать класс, который наследует абстрактный класс *FullAbstractMapper* с соответствующими классами, использованными в дженерике:

```

@Component
public class InspectionOrderMapper extends FullAbstractMapper<InspectionOrder,
InspectionOrderResp, InspectionOrderReq> {
    public InspectionOrderMapper() {
        super(InspectionOrder.class, InspectionOrderResp.class);
    }
}

```

Для вызова метода маппера, который преобразует объект из типа запроса в объект основной сущности БД, необходимо вызвать на экземпляре созданного класса маппера следующий метод:

```
InspectionOrder newOrder = inspectionOrderMapper.toEntityFromReq(orderReq);
```

Все данные полей, названия которых совпадает, будут автоматически переданы в класс, к которому приводится преобразование, это касается также и полей, которые представляют собой списки данных.

Заключение. Разработка собственного алгоритма маппинга объектов передачи данных позволяет без дополнительной нагрузки на разработанное программное средство автоматизировать преобразование классов из одного типа объекта в другой, избегая написания однотипного кода сеттинга данных, сохраняя использование паттерна объектов DTO, которые позволяют чётко разделить объекты на внутренние и внешние.

Список литературы

1. *The DTO Pattern (Data Transfer Object) | Baeldung [Электронный ресурс]* – Режим доступа: <https://www.baeldung.com/java-dto-pattern>. – Дата доступа: 24.03.2023.
2. *Naftalin, M. Java Generics and Collections / M. Naftalin, Ph. Walder.* – O'Reilly Media, Inc, 2007. – 280 с.
3. *Фаулер, М. UML. Основы, 3-е издание. Перевод А. Петухова / М. Фаулер.* – Пер. с англ. – СПб: Символ Плюс, 2004. – 192 с.
4. *Data Mapping: что такое маппинг данных, лучшие техники и инструменты – Mad Data [Электронный ресурс]*. – Режим доступа: <https://madata.agency/blog/data-mapping-luchshie-tekhniki-i-instrumenty>. – Дата доступа: 24.03.2023.

UDC 004.457

IMPLEMENTATION OF AN ALGORITHM FOR MAPPING DATA TRANSFER OBJECTS INTO THE BASIC ENTITIES OF THE PROGRAM

Medvedskaya Y.I.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Horbach A.P. – master of technical sciences, associate professor of the Department of ICSD

Annotation. The algorithm for implementing mapping of program objects is the creation of abstract classes that implement automatic conversion of an object of one type to another, avoiding the same type of code in the server part of the software.

Keywords: mapping, data transfer objects, java, abstract class, software tool