|  |
| --- |
|  |
|  |
| |  | | --- | |  | | МИНОБРНАУКИ РОССИИ | | Федеральное государственное бюджетное образовательное учреждение  высшего образования  **"МИРЭА - Российский технологический университет****"**  **РТУ МИРЭА** | |  | |  | |
| Институт информационных технологий (ИТ) |
| Кафедра инструментального и прикладного программного обеспечения (ИиППО) |

|  |
| --- |
| **ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ 2** |
| **по дисциплине** |
| **«Программное обеспечение интеллектуальных систем**»  *Изучение Jboss Drools* |

|  |  |
| --- | --- |
|  |  |
| Выполнил студент группы ИКБО-02-17 | *Крамаренко З.В* |
| Приняла | *Зорина Н.В.* |

|  |  |  |
| --- | --- | --- |
| Работа выполнена | «\_\_\_» \_\_\_\_\_\_\_\_2020 |  |
|  |  |  |
| «Зачтено» | «\_\_\_» \_\_\_\_\_\_\_\_2020 |  |

**Цель:** Научиться составлять бизнес правила в Drools и применять их в Java приложениях.

**Задание:**

1. Установить drools.
2. Создать несколько бизнес правил.
3. Продемонстрировать их работу.

**Ход работы**

Было принято решение использовать встроенный в Intellij Idea drools плагин.

В качестве темы была выбрана «Акционные предложения в книжном магазине».

Сначала необходимо создать классы, которые будут представлять бизнес объекты:

Класс Client представляет клиентов, у них есть ID, имя, почта, переменная отвечающая за наличие бонусов, а также список приобретенных книг.

Класс Book – книга. У книги есть ID, автор, название, цена, кол-во на складе. Также Book ссылается на класс Client.

Далее создадим несколько правил.

Предположим в книжном магазине ко Дню Рождения Харуки Мураками устраивают акцию: скидка на все книги автора 50 %.

rule "Birthday Promotion"

when

b : Book( author() = "Haruki Murakami")

then

newprice = (int) (b.price \* 0.5);

System.out.println("50% discount on the author's . New price: " + newprice);

end

Распродажа книг, кол-во которых на складе превышает 7 штук. Тогда действует скидка в размере 20%.

rule "Stock Rule"

when

b : Book( amount() > 7 )

then

newprice = (int) (b.price \* 0.8);

System.out.println("Sale! 20% discount for books from stock. New Price: " + newprice);

end

На складе накопилось много книг автора Марка Леви, поэтому магазин делает скидку 80% на книги данного автора, если их кол-во на складе превышает 10 штук. Помимо этого дарит бонус на следующею книгу.

rule "Sale.Bonus"

when

b : Book( author() = "Mark Levi" && amount() > 10)

then

newprice = (int) (b.price \* 0.2);

System.out.println("80% off book and next purchase bonus. New Price: " + newprice);

b.client.bonus = true;

System.out.print("Bonus: " + b.client.bonus);

end

Если клиент имеет бонус, то ему делается скидка 10% на любую книгу, не участвующею в акции.

rule "Bonus"

when

c : Client( bonus() = true)

then

newprice = (int) (c.book.price \* 0.9);

System.out.println("Bonus from a previous purchase: " + newprice);

end

Далее попробуем протестировать наши правила. Подключим все необходимые библиотеки и создадим сессии.

package com.javainuse.main;  
import com.javainuse.main.\*;  
import org.junit.runner.RunWith;  
import java.awt.\*;  
import java.awt.print.Book;  
KieServices ks = KieServices.Factory.get();  
KieContainer kContainer = ks.getKieClasspathContainer();  
KieSession kSession = kContainer.newKieSession ("ksession-rules");  
kSession = kContainer.newKieSession("ksession-rules");  
@RunWith(DroolsJUnitRunner.class)  
@DroolsFiles(value = "Rules.drl", location = "/drl/")

public class Test {  
 @DroolsSession  
 StatefulSession session;

Создадим двух клиентов.

Client a = new Client(1, "Den","a@gmail.com", false);  
Client b = new Client(2, "Hope","b@mail.com", false);

Сначала пропишем тест для проверки акции ко Дню Рождения.

*//акция ко дню рождения автора (скидка 50%)*{  
 Book b1 = new Book (001, "Haruki Murakami", "1Q84", 400, 5, a);  
 kSession.*insert*(b1);  
 kSession.*fireAllRules*();  
}

Далее для распродажи со склада.

*//Распродажа! Скидка 20% на книги со склада*{  
 Book b3 = new Book (003, "Son Won-Pyung", "Almond", 313, 10, a);  
 kSession.*insert*(b3);  
 kSession.*fireAllRules*();  
}

После тест для распродажи книг Марка Леви.

*/Распродажа книг Марка Леви с большим кол-вом на складе. Скидка 80% и бонус на следующую покупку*{  
 Book b2 = new Book (002, "Mark Levi", "P.S. from Paris", 257, 13, b);  
 kSession.*insert*(b2);  
 kSession.*fireAllRules*();  
}

И в заключении тест для проверки бонусной системы.

*//Бонус со следуюзей покупки*{  
 Book b4 = new Book (004, "Jeffrey Young", "Get out of the vicious circle", 500, 2, b);  
 kSession.*insert*(b4);  
 kSession.*fireAllRules*();  
}

Проверим работу наших правил:

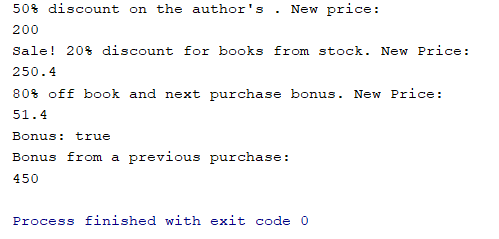


Рисунок 1. Выход программы

Правила сработали так, как ожидалось.

**Вывод**

В ходе выполнения лабораторной работы мы научились составлять бизнес правила в Drools и применять их в Java приложениях.

Drools – хороший инструмент, чтобы отделить бизнес логику программы от основного кода. Важно заранее продумывать архитектуру классов, чтобы в дальнейшем можно было создавать комплексные правила. Правила выполняются в порядке, в котором они встречаются в файле, поэтому можно создавать цепочки правил, и они будут выполняться последовательно.

**Список литературы**

1. Mauricio Salatino. Mastering JBoss Drools 6 for Developers/ - М.: Изд-во Книги по требованию, 2016

2. Петер Флах. Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из данных / - М.: Изд-во ДМК Пресс, 2019

3. IBM. Реализация бизнес-логики при помощи процессора правил Drools / URL: <https://www.ibm.com/developerworks/ru/library/j-drools/index.html>

4. Материалы лекций