**Lab Exercise 10- Using Collections (contains, memberOf) in Drools**

**Objective:**

* To demonstrate how to use the contains and memberOf operators in Drools to work with collections and perform various operations such as checking membership and containment.

**Step 1: Set Up the Project**

* Ensure you have a Maven project set up with Drools dependencies. You can use your existing project setup or create a new one.

**Step 2: Define Java Model Classes**

**Create a User Class:**

In the src/main/java/com/example/model directory, create a User.java file with the following content:

package com.example.model;

import java.util.Set;

public class User {

private String username;

private Set<String> roles;

public User(String username, Set<String> roles) {

this.username = username;

this.roles = roles;

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

public Set<String> getRoles() {

return roles;

}

public void setRoles(Set<String> roles) {

this.roles = roles;

}

@Override

public String toString() {

return "User{username='" + username + "', roles=" + roles + "}";

}

}

**Step 3: Create DRL Rules with Collection Operations**

**Create the Rule File:**

In the src/main/resources directory, create a file named userRolesRules.drl with the following content:

package com.example.rules

import com.example.model.User

rule "Admin Role Check"

when

$user : User(roles contains "ADMIN")

then

System.out.println($user.getUsername() + " has ADMIN role.");

end

rule "User Role Check"

when

$user : User(roles contains "USER" || roles contains "GUEST")

then

System.out.println($user.getUsername() + " has either USER or GUEST role.");

End

rule "No Admin Role"

when

$user : User(roles not contains "ADMIN")

then

System.out.println($user.getUsername() + " does not have ADMIN role.");

end

**Explanation of the Rules:**

* Admin Role Check: Checks if the roles set of the User object contains the "ADMIN" role using the contains operator.
* User Role Check: Checks if the roles set of the User object contains either "USER" or "GUEST" using the memberOf operator.
* No Admin Role: Checks if the roles set of the User object does not contain the "ADMIN" role.

**Step 4: Update the Main Application**

Modify the DroolsTest Class:

Update the DroolsTest.java file to demonstrate collection operations:

package com.example;

import com.example.model.User;

import org.kie.api.KieServices;

import org.kie.api.runtime.KieContainer;

import org.kie.api.runtime.KieSession;

import java.util.HashSet;

import java.util.Set;

public class DroolsTest {

public static void main(String[] args) {

// Load the knowledge base

KieServices ks = KieServices.Factory.get();

KieContainer kContainer = ks.getKieClasspathContainer();

KieSession kSession = kContainer.newKieSession("ksession-rules");

// Create sample users

Set<String> roles1 = new HashSet<>();

roles1.add("ADMIN");

roles1.add("USER");

Set<String> roles2 = new HashSet<>();

roles2.add("USER");

Set<String> roles3 = new HashSet<>();

roles3.add("GUEST");

User user1 = new User("alice", roles1);

User user2 = new User("bob", roles2);

User user3 = new User("charlie", roles3);

// Insert users into the session

kSession.insert(user1);

kSession.insert(user2);

kSession.insert(user3);

// Fire all rules

kSession.fireAllRules();

// Dispose the session

kSession.dispose();

}

}

**Step 5: Run the Application**

**Compile and Run:**

Compile and run the DroolsTest class.

**Expected Output:**

The console should display output similar to the following:

alice has ADMIN role.

alice has either USER or GUEST role.

bob has either USER or GUEST role.

charlie has either USER or GUEST role.

bob does not have ADMIN role.

charlie does not have ADMIN role.