1. 规则引擎
2. 为何使用规则引擎？

我们的业务逻辑是根据具体的业务需求来设置业务规则的，如果直接将业务规则硬编码实现，那么之后的维护就十分麻烦，且修改代码之后重启服务才能生效。而规则引擎能够很好地解决这个问题。

1. 规则引擎概述

规则引擎，全称为业务规则管理系统（Business Rule Management System, BRMS），规则引擎的主要思想是将程序中的业务决策部分分离出来，并使用预定义的语义模块编写业务决策（业务规则），由用户或开发者在需要时进行配置。

规则引擎不是一个具体的框架，而是一类系统，目前的规则引擎产品有：Drools, VisualRules, iLog等。规则引擎接受数据输入，解释业务规则，并根据业务规则作出业务决策。

系统中引入规则引擎后，业务规则不再以代码的形式保存在系统中，取而代之的是处理规则的规则引擎，业务规则存储在规则库中，独立于程序。业务人员可以管理业务规则，如增删改查等等。

1. 规则引擎的优势
2. 业务规则与系统代码分离，实现业务规则的集中管理
3. 在不重启服务的情况下扩展和维护业务规则
4. 使用规则引擎提供的规则编辑工具，使得业务规则变得简单。
5. 规则引擎应用场景

规则引擎是用于一些复杂的业务规则且会频繁变动的系统。

1. 风险控制系统（风险贷款，风险评估）
2. 反欺诈项目（银行贷款，征信验证）
3. 决策平台系统（财务计算）
4. 促销平台系统（打折，满减）
5. Drools介绍

Drools是一款由JBoss组织提供的基于Java开发的开源规则引擎，将业务规则以规则脚本的形式存放在文件或者特定的存储介质中（数据库），使得业务规则的变更不需要修改项目代码。重启服务器就可以生效。

1. Drools的使用

在项目中使用Drools时，可以单独地使用，直接导入maven坐标即可：

<dependency>

<groupId>org.drools</groupId>

<artifactId>drools-compiler</artifactId>

<version>7.6.0.Final</version>

</dependency>

在IDEA中已经集成了Drools插件了，可以非常方便地使用。

1. Drools开发步骤
2. 获取KieServices
3. 获取KieContainer
4. KieSession
5. Insert fact
6. 触发规则
7. 关闭KieSession