**1.环境搭建**

1.1 新建maven工程

1.2 配置activiti-cfg.xml

1.3 初始化数据库

**2.绘制工作流流程图bpmn**

2.1 安装acrivitiDesigner插件

2,2 生成bpmn 和 png 文件

注意三点：

整个流程图的id和名称，在act\_re\_procdef 表中 id 是流程的NAME，名称是KEY

每个任务的id和名称，

每个任务的执行者

**3.在系统中进行流程部署相关操作**

3.1 创建流程定义对象

获取工作流引擎，将工作流流程图资源，工作流名称注册到流程部署对象中，创建流程部署对象。

具体表现：

1.将流程注册时信息保存到流程部署表 act\_re\_deployment，流程定义表 act\_re\_procdef，流程资源表 act\_ge\_bytearray中；

2.可以通过部署对象，获取id 和 名称

3.2 删除流程定义（删除此定义下的所有流程实例）

获取该实例的流程定义id，记录在act\_re\_procdef表中

参考：

ProcessVariablesOptTest.testDeleteProcessDefinition

**4.创建流程实例对象**

流程定义部署以后，一个流程实例对象，相当于一次完整的流程。

参考：

StartProcessTest.testStartProcessTest1Process

**5.查询流程实例相关属性，操作流程变量，完成当前id任务**

5.1 查询流程实例相关属性

5.1.1 查询流程实例当前任务

ProcessOptTest.testQueryProcessTest1Task - 获取当前任务相关信息

ProcessOptTest.testQueryProInstanceState -获取当前任务名字

5.1.2 通过当前任务，查询流程实例相关属性

"流程定义ID:" + task.getProcessDefinitionId()

"流程实例ID:" + task.getProcessInstanceId()

"执行对象ID:" + task.getExecutionId()

"任务ID:" + task.getId()

"任务名称:" + task.getName()

"任务的办理人:" + task.getAssignee()

"任务的创建时间:" + task.getCreateTime()

5.1.3 通过当前流程实例，查询流程实例相关属性

"流程定义的ID:" + processInstance.getProcessDefinitionId()

"流程实例的ID:" + processInstance.getProcessInstanceId()

5.2 操作流程变量

流程变量的作用域:只对应一个流程实例 ,也就是说各个流程实例的流程变量是不相互影响的(不可共享)

设置流程变量有三种方式：

\* 单个设置

\* map集合设置

\* javaBean设置(序列化) - 相当于第1种方式，只是value可以是一个bean对象

5.2.1【添加】

5.2.1.1 在启动流程实例时设置

通过.startProcessInstanceByKey(processDefinitionKey,variables)

声明周期：

在流程实例执行过程中，使用

在完成任务后，这个变量会消失

参考：

ProcessVariablesOptTest.testSetVariablesStartProcessProcess

5.2.1.2 在RuntimeService时设置（同4.2.1.1，会覆盖4.2.1.1的值）

通过.startProcessInstanceByKey(processDefinitionKey,variables)

声明周期：

在流程实例执行过程中，使用

在完成任务后，这个变量会消失

参考：

ProcessVariablesOptTest.testSetVariablesInRuntimeServiceProcess

5.2.1.3 在TaskService时设置

通过.getTaskService().setVariables(taskId , variables) 和

通过.taskService.setVariable(taskId, variableName, value)

声明周期：

会保存到运行时变量表act\_ru\_variable中

参考：

ProcessVariablesOptTest.testSetVariablesInTaskProcess

5.2.1.4 在任务执行完毕时设置

通过.getRuntimeService().setVariables(executionId, variables)

声明周期：

在完成任务后，会保存到历史变量表中

参考：

ProcessVariablesOptTest.testSetVariablesStartProcessProcess

5.2.1.5 setVariable() 和 setVariableLocal的区别

setVariable：设置流程变量的时候，流程变量名称相同的时候，后一次的值替换前一次的值，而且可以看到TASK\_ID的字段不会存放任务ID的值

setVariableLocal：使用setVariableLocal说明流程变量绑定了当前的任务，当流程继续执行时，下个任务获取不到这个流程变量，需要查询历史的流程变量。

5.2.2【查询】

5.2.2.1 通过RuntimeService获取变量

通过processEngine.getRuntimeService().getVariables(executionId)

参考：

ProcessVariablesOptTest.findOneProcessRuntimeServiceVariables

5.2.2.2 通过TaskService获取变量

通过processEngine.getTaskService().getVariables(taskId)

参考：

ProcessVariablesOptTest.findOneProcessTaskServiceVariables

5.2.2.3 通过HistoryService获取流程实例中的所有历史变量

参考：

ProcessVariablesOptTest.findOneProcessHistoryVariables

5.3 完成当前id任务

选择流程实例中任务id 完成 ，taskId指代act\_ru\_task 中 EXECUTION\_ID\_字段值

具体执行内容：

1.act\_ru\_variable和act\_ru\_task 中数据会清空

2.act\_hi\_taskinst 中的任务会添加end时间

3.当前完成的任务如果有流程变量会存放到act\_hi\_varinst

4.act\_hi\_procinst 中的流程实例会添加end时间

5.act\_hi\_actinst会添加完成的任务记录

----------------------------------------------

注意变量

流程实例ID ProcessInstanceId act\_ru\_task PROC\_INST\_ID\_

流程定义ID ProcessDefinitionId act\_ru\_task PROC\_DEF\_ID\_

流程执行ID executionId act\_ru\_task EXECUTION\_ID\_

流程任务ID taskId act\_ru\_task ID

四个工作流引擎Service API

getRepositoryService()

getRuntimeService() 跟任务处理相关的服务类

getTaskService() 跟任务处理相关的服务类

getHistoryService() 跟历史相关的服务类

--------------------------------------------------

疑问：

1.在.getRuntimeService().startProcessInstanceByKey(processDefinitionKey, variables); 和

pe.getRuntimeService().setVariable(executionId , variableName , value)添加有什么不同？

其实是一样的，而且如果相同的key 会被覆盖，统一使用processEngine.getRuntimeService().getVariables(executionId);方式获取流程变量

2.executionId 是如何获得

通过查找当前执行的TaskService，创建一个任务查询

3.act\_hi\_actinst 如何使用？

-------------------------------------------------------

**5.用户、用户组设置**

用户任务分配主要有三种方式

\* 将任务分配个某个人

\* 将任务分配个某群人

\* 将任务分配个某个角色(组)

-------------------------------------------

疑问：

1.taskCandidateUser 应该如何使用

2.

**6.事件监听器**