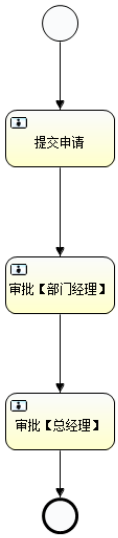
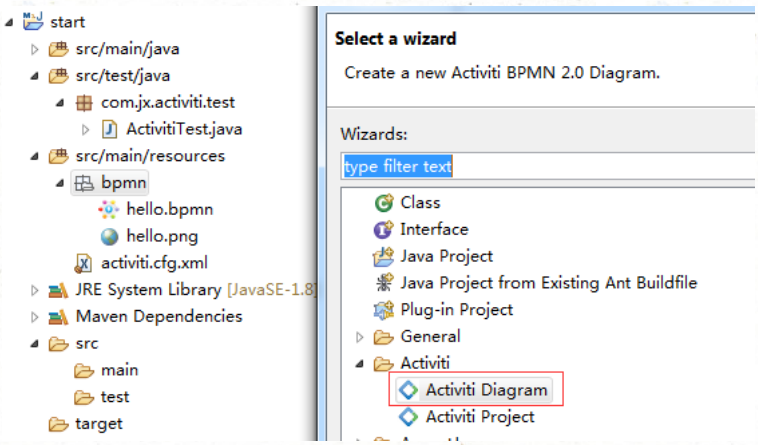
目标：完成如下图一个流程，张三提交申请，李四（部门经理）审批，王五（总经理）审批



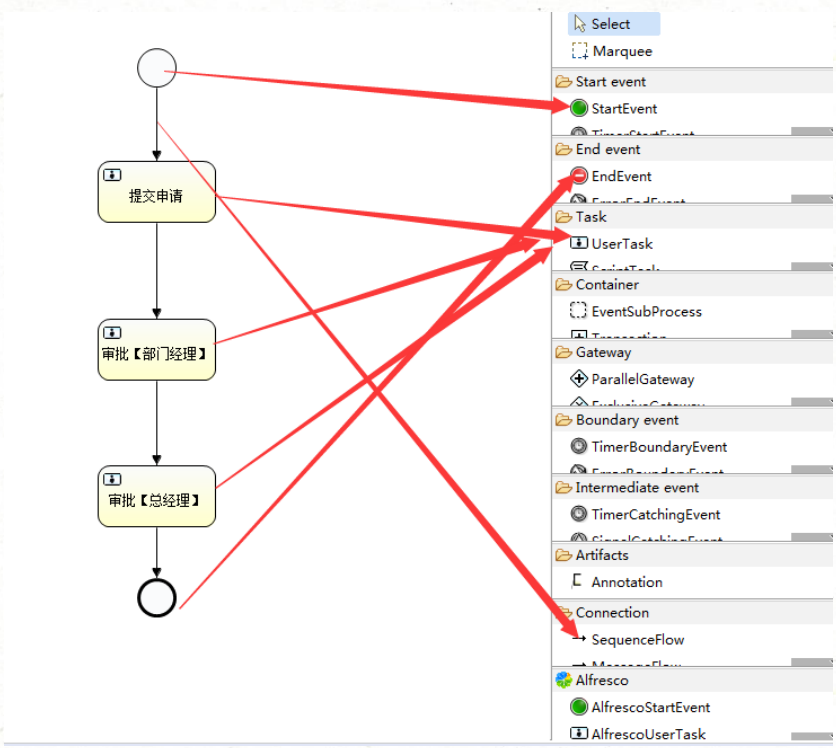
1　画流程图：

打开eclipse的propperties视图 --》window - show view - properties

在 resources下新建一个package：bpmn ， 然后在里面新建一个activiti的流程文件 hello.bpmn



流程图说明：



最开始是一个开始标记，开始标记通过一个流程连线到第一个用户任务（提交申请），之后又通过一个流程连线到第二个用户通过一个流程连线到第三个用户任务（审批【总经理】），最后通过一个流程连线到结束标记

提交申请通过properties视图，在 tab General 填写了name 为提交申请， 在 Main config 中 Assignee 填写了任务处理人 张

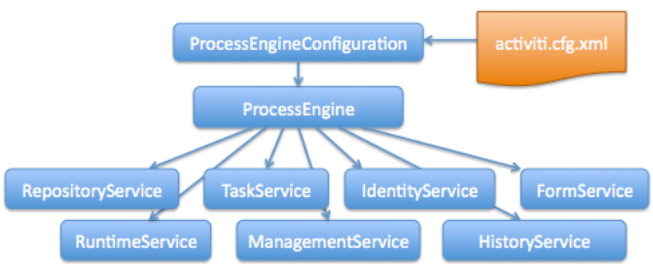
同样 在第二个（审批【部门经理】），第三个（审批【总经理】）用户任务 填写了名称与任务处理人

最后点击流程图的空白处 在 process 这一栏的 ID 填写 helloworld Name 填写 helloworldProcess ，然后点击保存即可

hello.bpmn
3.28KB

2 执行流程

引擎API 是与 Activiti 交互最常用的方式。主要的出发点是 ProcessEngine （流程引擎），可以使用配置一节中介绍的几种方式对其创建。由ProcessEngine ，可以获取到含有工作流/BPM 方法的不同服务。ProcessEngine 以及那些服务对象都是线程安全的。所以可以为整个服务器保留一份它的引用。



```
ProcessEngines.getDefaultProcessEngine().getRuntimeService();
ProcessEngines.getDefaultProcessEngine().getRepositoryService();
ProcessEngines.getDefaultProcessEngine().getTaskService();
ProcessEngines.getDefaultProcessEngine().getManagementService();
ProcessEngines.getDefaultProcessEngine().getIdentityService();
ProcessEngines.getDefaultProcessEngine().getHistoryService();
ProcessEngines.getDefaultProcessEngine().getFormService();
```

RepositoryService	**管理流程定义**
RuntimeService	**执行管理，包括启动、推进、删除流程实例等操作**
TaskService	**任务管理**
HistoryService	**历史管理(执行完的数据的管理)**
IdentityService	**组织机构管理**
FormService	一个可选服务，任务表单管理
ManagerService	

执行代码：

```
package com.jx.activiti.test;

import java.util.List;

import org.activiti.engine.ProcessEngine;
import org.activiti.engine.ProcessEngineConfiguration;
import org.activiti.engine.ProcessEngines;
import org.activiti.engine.repository.Deployment;
import org.activiti.engine.runtime.ProcessInstance;
import org.activiti.engine.task.Task;
import org.junit.Before;
import org.junit.Test;

public class ActivitiTest {

    @Test
    public void init() {
        System.out.println("初始化activiti 23张表");
        ProcessEngineConfiguration.createProcessEngineConfigurationFromResource("activiti.cfg.xml").buildProcessEngine(
    }

    private static ProcessEngine processEngine = null;

    @Before
    public void initProcessEngine() {
        //这里默认的也会去找resources目录下的activiti.cfg.xml配置文件加载
```

```
        processEngine = ProcessEngines.getDefaultProcessEngine();
    }

    @Test
    public void deploymentProcessDefinition(){
        System.out.println("部署流程");
        Deployment deployment = processEngine.getRepositoryService()//与流程定义和部署对象相关的Service
                                .createDeployment()//创建一个部署对象
                                .name("hello入门程序")//添加部署的名称
                                .addClasspathResource("bpmn/hello.bpmn")//从classpath的资源中加载，一次只能加载一个文件
                                .addClasspathResource("bpmn/hello.png")//从classpath的资源中加载，一次只能加载一个文件
                                .deploy();//完成部署
        System.out.println("部署ID："+deployment.getId());//1
        System.out.println("部署名称："+deployment.getName());//helloworld入门程序
    }

    @Test
    public void startProcessInstance(){
        System.out.println("启动流程实例");
        //流程定义的key
        String processDefinitionKey = "helloworld";//就是hello.bpmn在xml视图中的 process 中的 id属性
        ProcessInstance pi = processEngine.getRuntimeService()//与正在执行的流程实例和执行对象相关的Service
                                .startProcessInstanceByKey(processDefinitionKey);//使用流程定义的key启动流程实例，key对应hel
key值启动，默认是按照最新版本的流程定义启动
        System.out.println("流程实例ID:"+pi.getId());//流程实例ID    101
        System.out.println("流程定义ID:"+pi.getProcessDefinitionId());//流程定义ID   helloworld:1:4
    }

    @Test
    public void findMyPersonalTask(){
        System.out.println("查询当前人的个人任务");
        String assignee = "王五";
        List<Task> list = processEngine.getTaskService()//与正在执行的任务管理相关的Service
                                .createTaskQuery()//创建任务查询对象
                                .taskAssignee(assignee)//指定个人任务查询，指定办理人
                                .list();
        if(list!=null && list.size()>0){
            for(Task task:list){
                System.out.println("任务ID:"+task.getId());
                System.out.println("任务名称:"+task.getName());
                System.out.println("任务的创建时间:"+task.getCreateTime());
                System.out.println("任务的办理人:"+task.getAssignee());
                System.out.println("流程实例ID："+task.getProcessInstanceId());
                System.out.println("执行对象ID:"+task.getExecutionId());
                System.out.println("流程定义ID:"+task.getProcessDefinitionId());
                System.out.println("########################################################");
            }
        }
    }

    @Test
    public void completeMyPersonalTask(){
        System.out.println("完成我的任务");
        //任务ID
        String taskId = "302";
```

```
            processEngine.getTaskService()//与正在执行的任务管理相关的Service
                    .complete(taskId);
            System.out.println("完成任务：任务ID："+taskId);
        }
    }
```