## 什么是贴现计算器

它是一种用户计算银行承兑汇票贴现利息的计算器。

用户输入信息包括：

1.票面金额：持票人的银行承兑汇票上记载的“出票金额”。

2.月利率：持票人在申请贴现当日金融机构所报的银行承兑汇票贴现率，月利率以千分率%0来计算。

3.贴现日：持票人向金融机构申请银行承兑汇票贴现当日。

4.到期日：银行承兑汇票票面上的“汇票到期日”。

5.调整天数：根据中国人民银行《支付计算办法规定》：“承兑人在异地的，贴现，转贴现和再贴现的期限以及贴现利息的计算应加上3天的划款日期”；贴现银行与承兑银行如果属于异地，贴现天数应当再加上3天。

6.年利率：汇票面值 × 年贴现率x% × （贴现日－承兑汇票到期日）的天数 / 360。

计算器显示信息包括：

1.计息天数：贴现日到汇票兑换日之间的相隔的天数。

2.贴现利息：在人民银行现行的再贴现利率的基础上进行上浮，贴现的利率是市场价格，由双方协定，但最高不能超过现行的贷款利率。

3.贴现金额：汇票收购者在收购汇票时，实际付给汇票出售者的现金金额。

计算公式：

贴现利息=汇票金额\*贴现天数\*（月贴现率/30）

实付贴现金额=汇票金额-贴现利息

计息天数=到期日-贴现日+调整天数

## 数据计算确定公式

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 票面金额（万元） | 10 | 25 | 40 | 100 |
| 月利率（千分率） | 4.2 | 3.8 | 4.4 | 4.5 |
| 贴现日 | 2019，6，25 | 2019，5，30 | 2018，8，27 | 2018，9，27 |
| 到期日 | 2019，8，29 | 2019，9，30 | 2019，8，27 | 2019，9，27 |
| 调整天数 | 3 | 4 | 0 | 3 |
| 计息天数 | 5+31+29+3=68 | 1+30+31+31+30+4=127 | 365 | 368 |
| 贴现利息 | 952 | 4021.67 | 21413.33 | 55200 |
| 贴现金额 | 99048 | 245978.33 | 378586.67 | 944800 |
| 计算出错 | 贴现日不算 | 精确到小数点后两位（四舍五入） |  |  |

## 功能模块

1. 主模块：计算器功能模块；
2. 挂失票据查询模块；
3. 参考信息展示模块；Shibor信息，当日的贷款利率，当日人民银行再贴现利率。

## 代码开发计划

前端：使用JSP

后端：使用MySQL数据库存储数据。

框架：使用SpringMVC或者SpringBoot

扩展：使用python。

## 开发环境配置：

完成此项目需要的软件工具包括：tomcat作为web服务器，记事本作为JSP页面的编程工具，IDEA作为servlet的编程工具。

配置tomcat：在官网上下载tomcat并且配置环境变量CATALINA\_HOME；CATALINA\_BASE它们分别指向tomcat的安装目录和tomcat的工作目录。

# 第一阶段完成主模块功能计算器：

## 第一步：使用JSP完成前端页面的编写。

1. 确定使用哪种编码方式？

使用UTF-8编码方式。

A）调整文件编码：使用记事本（不能使用写字板）来打开文件，另存为编码方式选择UTF-8。默认的编码方式为ANSI。

B）设定浏览器解析网页时使用的编码方式为UTF-8。

<%@ page language="java" contentType="text/html; charset=UTF-8"  
 pageEncoding="UTF-8"%>。

1. 将前端的用户输入数据进行收集

使用form标签来完成数据的收集和传递工作，在form中设置5个input type=”text”用来完成数据的收集工作。

在数据收集的过程中要保证数据收集的有效性和可用性，故决定了使用特殊的手段保证数据收集过程中尽量不出现错误。

在贴现日，到期日的input中使用了type=“date”的方式来保证数据的可用性。

生成日历效果

使用被html5支持的input标签中的属性type=”date”

明确被input type=“data” 所提交的数据类型和使用方式。

提交的数据类型为被格式化成字符串的date实例，date提供了将date对象转为字符串的方法，也提供了将字符串（该字符串必须是由date对象通过格式化转换而来的）重新转为date的方法。

1. 前端页面布局设置

大标题：<h1>并且设置居中。

副标题：应该使用<h3>,并设置为居左。

数据输入部分：使用<table>标签和<form>标签相结合的方式设计，其中form主要负责数据的收集，而table负责页面排版。

默认情况下每行之间的间距太小，想要加大间距 ：cellspacing=”20”此属性增加单元格之间的间距，故它不但增加了<td>之间的距离，而且增加了<tr>之间的距离。

调整table让它看起来像是居中的状态：

首先设置width为100%，这保证了整个table在中间位置。

然后调整显示内容的位置，<td>的width值可以根据需要设置为45%和55%。

### 4. 添加保护手段，增强程序的健壮性

1.设置输入框，只能输入数字，不能输入别的数据类型。

使用input中的pattern属性来限制输入内容。

解决方案：问题解决使用了正则表达式 pattern="([0-9]\*\.[0-9]+)|([0-9]\*)"title="只能输入数字"。在HTML中也有正则表达式，而且和Java中的正则表达式有些相同[0-9]表示输入数据可以是数字0-9，\*表示可以匹配0次或者多次；\.：反斜杠+点表示匹配字符点.，因为点在正则表达式中有其他的特殊含义，所以需要\来进行转义；[0-9]+：表示0-9这些数字可以被匹配1到多次；这是考虑到了小数的情况。后面的“|“表示的含义为“或”。

[0-9]\*：表示可以输入整数。故整个正则表达式的含义可以理解为允许输入的数据类型为数值类型。

在票面金额，月利率input中限制只能输入数字型数据，在调整天数input中限制只能输入整数数据。

2.限制文本框的必须具有输入内容

解决方案：查找input中的属性，发现属性required：可以解决上面的问题。

设置required的属性值为：required = “required”。

3.为方便用户的使用和保持程序的健壮性要在页面刷新的时候将光标放到第一个输入框

解决方案：使用input标签中的autofocus属性，该属性含义为当页面刷新时光标自动放置到input中，并且每个页面只能设置一个该属性。

### 5.数据流向后端

在前端收集完数据后可以在前端进行处理，也可以在后端进行处理，但是选用JSP页面后，根据JSP页面的本质为一个servlet类的特性我们将会选择把数据传输到后端，然后进行处理。

1. 数据由前端流向后端。

在form的标签中使用action=“HelloForm”，method=“POST”。Action=“HelloForm”表示将数据提交到HelloForm.class中，该class文件位于/ROOT/WEB-INF/classes文件夹下，并且该文件夹为存放class文件的根目录。Method=“POST”表示使用POST的方式来提交数据。点击input type=“submit”的按钮

1. 将前端和后端联系起来。

第一种联系：让前端JSP页面明白action=“HelloForm”指向的是那个文件，将HelloForm.class文件放入ROOT/WEB-INF/classes文件夹下，tomcat会自动以该文件夹为根目录寻找class文件，如果在classes文件夹（classes文件夹本就为用户所建）下又新建文件夹f1，把HelloFrom.class文件放入到f1中则action=“f1.HelloForm.class”。

第二种联系：让浏览器明白输入localhost：8080/HelloForm，要将访问交给HelloForm.class处理，使用注解的方式，[在HelloForm.java中添加@WebServlet("/HelloForm")](mailto:在HelloForm.java中添加@WebServlet(%22/HelloForm%22))。

注意：2.-》第二种联系【在HelloForm.java中加入注解后必须将web.xml中的metadate-complete=“true”，修改为false。因为metadate-complete表示该web.xml文件是否完全，如果为true则会忽略所有的注解。】

### 6.在httpservlet中完成数据处理然后传输到前端

Httpservlet是一个可以处理HTTP请求的servlet，它本质上是一个类，该类间接继承了servlet接口（servlet就是一个接口interface）。开发人员如果想要处理HTTP请求，就应该直接继承该类，不要继承servlet接口。

在httpservlet中有service方法，该方法可以判断传入的请求是POST类型的还是GET类型的，如果是POST类型就调用doPost方法，如果是GET类型的就调用doGet方法。

开发人员进行开发时只需重写doPost，或doGet方法。

1. 在doGet方法中获取前端提交来的数据

使用request.getParameter(“input的name值”)。其中request是JSP页面的9个隐式对象之一，是javax.servlet.http.httpservletRequest类的一个对象，每当客户请求一个JSP页面时，JSP就会产生一个新的request对象来代表这个请求。

Request.getParameter()；使用该方法得到的数据为String类型，使用request.getAttribute()得到的数据为对象类型。

1. 将request.getAttribute()传递过来的String类型的数据转为运算时需要的类型。

将String转为float，使用Float.parseFloat(String s);

将String转为Data，使用SimpleDateFormat，首先：知道由Date转为String时使用的SimpleDateFormat，SimpleDateFormat ft = new SimpleDateFormat("yyyy-MM-dd");然后使用SimpleDateFormat中的parse（String）方法将String转为Date。

b\_Dday = ft.parse(discount\_Day)；

1. 计算天数

根据两个日期距离1970-01-01：00：00：00.000（标准日期）的毫秒数的差值来计算两个日期之间相距的天数。

Date.getTime()方法将会返回日期Date距离标准日期的毫秒数。

**long** from = b\_Dday.getTime();  
**long** to = b\_Eday.getTime();  
days = (**int**) ((to-from)/(1000\*60\*60\*24));

1. 计算出结果，计算公式已经在前面写出此处不再讨论。