单元测试报告

目录

完整作业见homework.html

输入：

```
2016-06-02 20:00~22:00 7
2016-06-03 09:00~12:00 14
2016-06-04 14:00~17:00 22
2016-06-05 19:00~22:00 3
2016-06-06 12:00~15:00 15
2016-06-07 15:00~17:00 12
2016-06-08 10:00~13:00 19
2016-06-09 16:00~18:00 16
2016-06-10 20:00~22:00 5
2016-06-11 13:00~15:00 11
```

输出：

```
[Summary]

2016-06-02 20:00~22:00 +210 -240 -30
2016-06-03 09:00~12:00 +420 -180 +240
2016-06-04 14:00~17:00 +660 -600 +60
2016-06-05 19:00~22:00 +0 -0 0
2016-06-06 12:00~15:00 +450 -300 +150
2016-06-07 15:00~17:00 +360 -200 +160
2016-06-08 10:00~13:00 +570 -330 +240
2016-06-09 16:00~18:00 +480 -300 +180
2016-06-10 20:00~22:00 +150 -120 +30
2016-06-11 13:00~15:00 +330 -200 +130

Total Income: 3630
Total Payment: 2470
Profit: 1160

end
```

单元分布

设计背景：

小明打算成立羽毛球俱乐部，不定期组织羽毛球活动。参加活动的人向小明支付费用，小明向羽毛球馆预定球场并支付费用。帮助小明结算收入。

设计分析：

设计网页，使用javascript实现计算。设计单元如下：

1. 网页布局，获取输入内容并输出。
2. 依据年、月、日计算星期数
3. 依据星期数、场地的租用时间段计算一个场地一日的开销
4. 依据人数计算需要租用的场地数
5. 合计收入和支出的输出以及最后的盈利

单元1

1.1单元描述

网页使用html来实现，主要为了获取输入内容并按照原格式输出

1.2单元结构

分为标题、输入提示、输入框textarea、输出四部分

1.3单元控制流图及主要代码

.html文件见test1.html

```
<!DOCTYPE html>
<html>
<head>
    <title>homework-zc</title>
    <meta charset="utf-8">
    <style type="text/css">
        body{font-family:  theta
            /*Arial,Lucida,Verdana,"宋体","微软雅黑",Helvetica,sans-serif*/;}
    </style>
</head>
<body>
<h1>羽毛球俱乐部收支费用</h1>
<p>输入格式为{活动时间yyyy-MM-dd HH:mm~HH:mm} {人数}</p>
<p>例如：2016-06-02 20:00~22:00 7</p>
<textarea id="input" rows="8" cols="40" placeholder="请在此输入！" wrap="hard"></textarea>
<input type="button" name="button" value="提 交" id="btn">
<p id="summary"></p>
<script type="text/javascript">
    var input=document.getElementById("input"),
        btn=document.getElementById("btn"),
        summary=document.getElementById("summary");
    var strs= new Array(); //定义一数组来储存每一行
    var i;
    var str="[Summary]<br><br>";//用于输出的文本字符串

    //提交按钮事件
    btn.onclick=function(){
        generateSummary(input);
    }

    //生成总计
function generateSummary(input){
    strs=input.value.split("\n"); //字符分割，一行分割为三个数组
        for (i=0;i<strs.length ;i++ )
        {
            str=str.concat(strs[i]);
            str=str.concat("<br> ");//分割后的字符输出
        }
        str=str.concat("<br>end");
        summary.innerHTML=str;

}
</script>
</body>
</html>
```

1.4测试数据及结果

单行输入：

```
2016-06-03 09:00~12:00 14
```

输出：

```
[Summary]

2016-06-03 09:00~12:00 14

end
```

多行输入：

```
2016-06-02 20:00~22:00 7
2016-06-03 09:00~12:00 14
2016-06-04 14:00~17:00 22
2016-06-05 19:00~22:00 3
2016-06-06 12:00~15:00 15
2016-06-07 15:00~17:00 12
2016-06-08 10:00~13:00 19
2016-06-09 16:00~18:00 16
```

输出：

```
[Summary]

2016-06-02 20:00~22:00 7
2016-06-03 09:00~12:00 14
2016-06-04 14:00~17:00 22
2016-06-05 19:00~22:00 3
2016-06-06 12:00~15:00 15
2016-06-07 15:00~17:00 12
2016-06-08 10:00~13:00 19
2016-06-09 16:00~18:00 16

end
```

单行、多行都以原格式输出，因此测试成功！

单元2

2.1单元描述

使用substr()获取需要的字符段，使用parseInt()转成数值，创建计算星期数的函数countWeekday()。 计算星期数使用Zeller公式

2.2单元结构

函数的参数1：cent 世纪-1

函数的参数2：yy 年份后两位

函数的参数3：mm 月份

函数的参数4：day 日期

返回：weekday 星期数0-6

2.3单元控制流图及主要代码

.html文件见test2.html

在for中添加

```
cent=parseInt(strs[i].substr(0,2));//世纪
cent=parseInt(strs[i].substr(0,2));//世纪
```

```
yy=parseInt(strs[i].substr(2,4));//年份后两位
mm=parseInt(strs[i].substr(5,2));//月
day=parseInt(strs[i].substr(8,2));//日
countWeekday(cent,yy,mm,day);//计算星期几  weekday
```

在`<script>`标签中添加函数

```
//计算星期
function countWeekday(cent,yy,mm,day){
    //蔡勒（Zeller）公式
    //蔡勒（Zeller）公式：w=y+[y/4]+[c/4]-2c+[26(m+1)/10]+d-1
    //w：星期；c：世纪-1；y：年（两位数）；m：月（m大于等于3，小于等于14，即在蔡勒公式中，某年的1、2月要看作上一年的13、14月来计算，比如2003年1月1日

    if(mm==1) mm=13;
    else if(mm==2) mm=14;
    weekday=yy+Math.floor(yy/4)+Math.floor(cent/4)-2*cent+Math.floor(26*(mm+1)/10)+day-1;
    weekday=Math.floor(weekday%7);
}
```

更改输出

```
str=str.concat(strs[i]+"星期："+weekday);
```

2.4测试数据及结果

多行输入：

```
2015-03-01 09:00~12:00 14
2015-12-04 14:00~17:00 22
2016-06-05 19:00~22:00 3
2016-06-06 12:00~15:00 15
2016-06-07 15:00~17:00 12
2017-06-08 10:00~13:00 19
2016-10-17 10:00~13:00 19
```

输出：

```
[Summary]

2015-03-01 09:00~12:00 14星期：0
2015-12-04 14:00~17:00 22星期：5
2016-06-05 19:00~22:00 3 星期：0
2016-06-06 12:00~15:00 15星期：1
2016-06-07 15:00~17:00 12星期：2
2017-06-08 10:00~13:00 19星期：4
2016-10-17 10:00~13:00 19星期：1

end
```

输入不同日期，对照万年历逐个检查输出正确，因此测试成功！

单元3

3.1单元描述

计算所租用的一个场地一日的花费

3.2单元结构

函数的参数1：weekday 星期数0-6

函数的参数2：start 开始时间

函数的参数2：end 结束时间

返回：dayprice 一个场地一日的开销

3.3单元控制流图及主要代码

.html文件见test3.html

注意：由于`dayprice`采用的是累加的方式来计价，因此获取每行数据后应对`dayprice=0;`初始化

```
function chargeStandard(weekday,start,end){
    if(weekday>=1&&weekday<=5){
        //周一到周五
        while(start!=end){
            if(start>=9&&start<12){
                dayprice=dayprice+30;
                start=start+1;}
            else if(start>=12&&start<18){
                dayprice=dayprice+50;
                start=start+1;}
            else if(start>=18&&start<20){
                dayprice=dayprice+80;
                start=start+1;}
            else if(start>=20&&start<22){
                dayprice=dayprice+60;
                start=start+1;}
        }
    }
    else{
        //周六及周日
        while(start!=end){
            if(start>=9 && start<12){
                dayprice=dayprice+40;
                start=start+1;}
            else if(start>=12&&start<18){
                dayprice=dayprice+50;
                start=start+1;}
            else if(start>=18&&start<22){
                dayprice=dayprice+60;
                start=start+1;}
        }
    }
}
```

3.4测试数据及结果

选择周一到周日的不同以及交叉的时间段进行测试。

输入：

```
2016-06-02 20:00~22:00 7
2016-06-03 09:00~12:00 14
2016-06-04 14:00~17:00 22
2016-06-05 19:00~22:00 3
2016-06-06 12:00~15:00 15
2016-06-07 15:00~17:00 12
2016-06-08 10:00~13:00 19
2016-06-09 16:00~18:00 16
2016-06-10 20:00~22:00 5
2016-06-11 13:00~15:00 11
```

输出：

```
[Summary]
```

```
2016-06-02 20:00~22:00 7 星期数：4 dayprice：120
2016-06-03 09:00~12:00 14 星期数：5 dayprice：90
2016-06-04 14:00~17:00 22 星期数：6 dayprice：150
2016-06-05 19:00~22:00 3 星期数：0 dayprice：180
2016-06-06 12:00~15:00 15 星期数：1 dayprice：150
2016-06-07 15:00~17:00 12 星期数：2 dayprice：100
2016-06-08 10:00~13:00 19 星期数：3 dayprice：110
2016-06-09 16:00~18:00 16 星期数：4 dayprice：100
2016-06-10 20:00~22:00 5 星期数：5 dayprice：120
2016-06-11 13:00~15:00 11 星期数：6 dayprice：100

end
```

对照时间段、星期数，逐个检查dayprice的输出正确，因此测试成功！

单元4

4.1单元描述

计算需要租用的场地数

4.2单元结构

函数的参数1：member 报名人数

返回：games 需要租用的场地数

4.3单元控制流图及主要代码

.html文件见test4.html

```
//计算场数
function countGames(member){
    t=Math.floor(member/6),x=Math.floor(member%6);
    if(t==0&&x<4) games=0;
    else if(t==0&&x>=4) games=1;
    else if(t==1) games=2;
    else if((t==2||t==3)&&x<4) games=t;
    else if((t==2||t==3)&&x>=4) games=t+1;
    else games=t;
}
```

4.4测试数据及结果

测试的数据应涉及订场策略的六种情况。

输入：

```
2016-06-02 20:00~22:00 7
2016-06-03 09:00~12:00 14
2016-06-04 14:00~17:00 22
2016-06-05 19:00~22:00 3
2016-06-06 12:00~15:00 15
2016-06-07 15:00~17:00 12
2016-06-08 10:00~13:00 19
2016-06-09 16:00~18:00 16
2016-06-10 20:00~22:00 5
2016-06-11 13:00~15:00 11
```

输出：

```
[Summary]

2016-06-02 20:00~22:00 7 场数：2
```

```
2016-06-03 09:00~12:00 14 场数：2
2016-06-04 14:00~17:00 22 场数：4
2016-06-05 19:00~22:00 3 场数：0
2016-06-06 12:00~15:00 15 场数：2
2016-06-07 15:00~17:00 12 场数：2
2016-06-08 10:00~13:00 19 场数：3
2016-06-09 16:00~18:00 16 场数：3
2016-06-10 20:00~22:00 5 场数：1
2016-06-11 13:00~15:00 11 场数：2

end
```

对照场订场策略，逐行检查场数输出正确，因此测试成功！

单元5

5.1单元描述

计算每次活动的收入、支出、盈利。

5.2单元结构

收入：`30*member`

支出：`dayprice*games`

盈利：`(30*member-dayprice*games)`

5.3单元控制流图及主要代码

.html文件见test5.html

总收入、总支出、总盈利

```
income+=30*member;//收入
payment+=dayprice*games;//支出:所需场地的费用=场地*场地日单价

profit=income-payment;//盈余
```

5.4测试数据及结果

输入：

```
2016-06-02 20:00~22:00 7
2016-06-03 09:00~12:00 14
2016-06-04 14:00~17:00 22
2016-06-05 19:00~22:00 3
2016-06-06 12:00~15:00 15
2016-06-07 15:00~17:00 12
2016-06-08 10:00~13:00 19
2016-06-09 16:00~18:00 16
2016-06-10 20:00~22:00 5
2016-06-11 13:00~15:00 11
```

输出：

```
[Summary]

2016-06-02 20:00~22:00 7 星期数：4 dayprice：120 场数：2 每日收入：+210 每日支出：-240 每日收益-30
2016-06-03 09:00~12:00 14 星期数：5 dayprice：90 场数：2 每日收入：+420 每日支出：-180 每日收益 +240
2016-06-04 14:00~17:00 22 星期数：6 dayprice：150 场数：4 每日收入：+660 每日支出：-600 每日收益 +60
2016-06-05 19:00~22:00 3 星期数：0 dayprice：180 场数：0 每日收入：+0 每日支出：-0 每日收益0
2016-06-06 12:00~15:00 15 星期数：1 dayprice：150 场数：2 每日收入：+450 每日支出：-300 每日收益 +150
```

```
2016-06-07 15:00~17:00 12 星期数：2 dayprice：100 场数：2 每日收入：+360 每日支出：-200 每日收益 +160
2016-06-08 10:00~13:00 19 星期数：3 dayprice：110 场数：3 每日收入：+570 每日支出：-330 每日收益 +240
2016-06-09 16:00~18:00 16 星期数：4 dayprice：100 场数：3 每日收入：+480 每日支出：-300 每日收益 +180
2016-06-10 20:00~22:00 5 星期数：5 dayprice：120 场数：1 每日收入：+150 每日支出：-120 每日收益 +30
2016-06-11 13:00~15:00 11 星期数：6 dayprice：100 场数：2 每日收入：+330 每日支出：-200 每日收益 +130

Total Income: 3630
Total Payment: 2470
Profit: 1160

end
```

对照范例，逐行检查输出正确，因此测试成功！