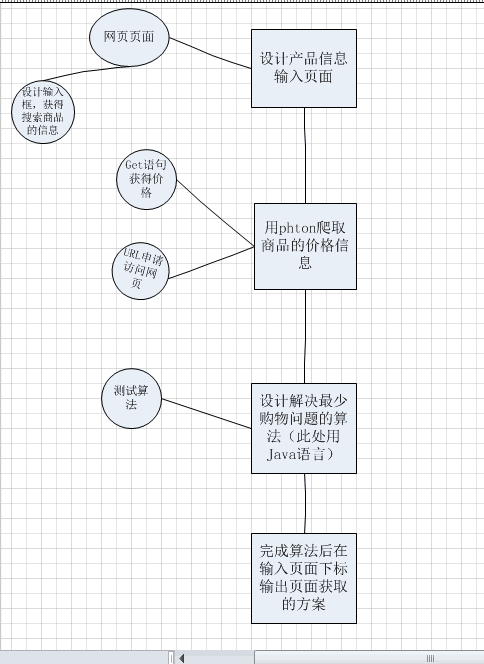
**1.小组成员及分工**

|  |  |
| --- | --- |
|  | 分工 |
| 组长：张梦华201706401229 | 写代码 |
| 组员：陈承娟201706401232 | 写代码 |
| 组员：农添添201706401227 | 算法设计 |
| 组员：李文 201707406146 | 需求分析 |
| 组员：黄兰花201706401246 | 文档撰写 |
| 组员：杜静娴201706401241 | 测试代码 |

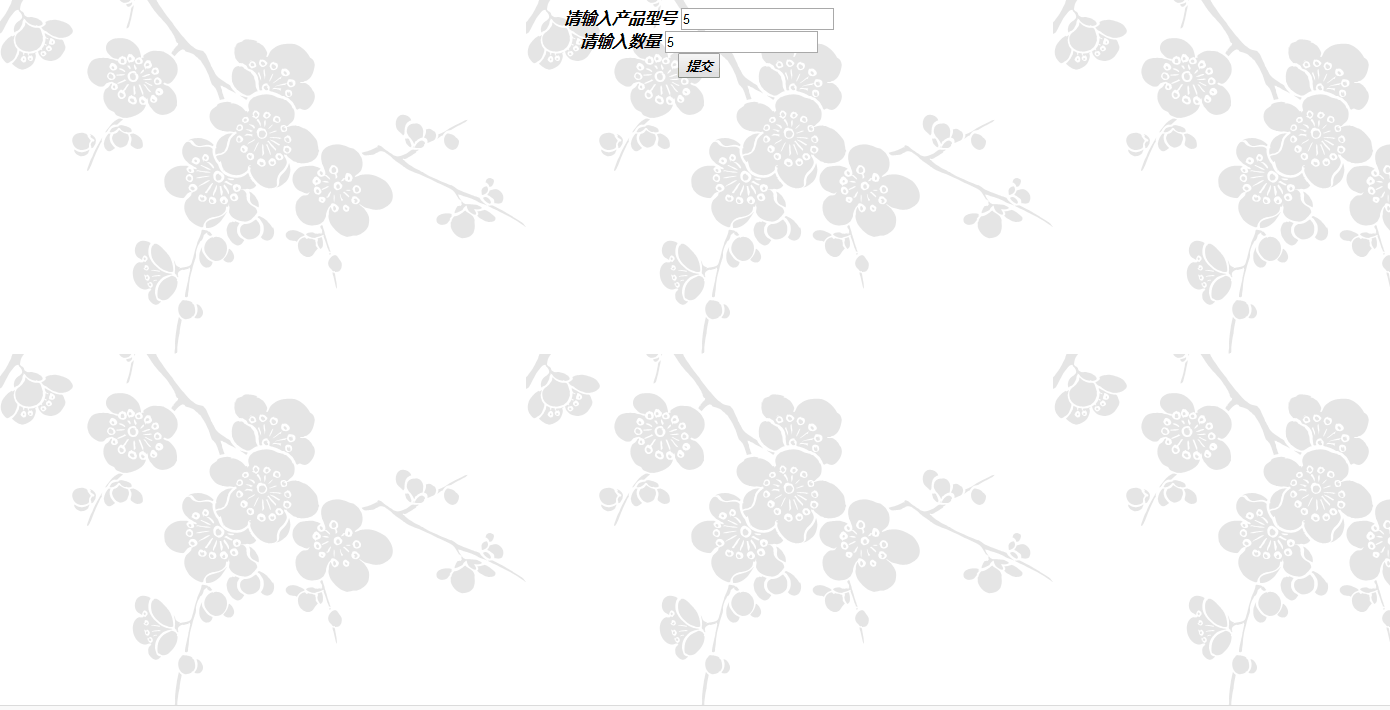
**2.需求分析**

随着网络技术的发展，网上购物成为很多人的首选，对于宅男宅女来说这更是一种便利的快捷方式。而作为用户，自然也希望在网上购物得到更好的体验。用户当然希望用最少的钱买到最实惠的商品。所以，根据这些情况，我们可以通过自己所学的知识去设计一个小程序，当用户想要花最少的钱买到最实惠的商品时，就可以通过这个小程序，在这个小程序里输入想要购买的商品及数量，然后这个小程序就会根据这些信息计算出花费最少的购物方案及费用，然后将结果返还。

**3.问题建模**



1. **产品信息页面代码及页面效果**

****

<!DOCTYPE HTML>  
<!DOCTYPE HTML>  
<html>  
<body background="h1.png">   
<div>    
  <div align="center"><strong> <em>请输入产品型号</em>  
    <input  type="text" id="pname" value="5">  
  </strong></div>  
</div>  
<div>   
  <div align="center"> <em><strong>请输入数量 </strong>  
    </em>  
    <input type="text" id="pnum" value="5">  
  </div>  
</div>  
<div id="mycontainor"> </div>  
<div>   
   <div align="center">  
    <button type="button" onclick="addProduct()">

<strong><em>提交</em></strong></button> </div>  
</div>  
<div align="center"> </div>  
<script>  
 function addProduct(){  
 var strl = '<div> 请输入产品型号

<input type="text" id="p1name" value="5"></div>'

strl +=  '<div> 请输入量

<input type="text" id="p1num" value="5"></div>'  
document.getElementById("mycontainor").innerHtml = strl;

}

</script>  
</body>  
</html>

1. **算法设计**

1、商品间有关联，用以明确这个商品可以与哪些商品搭配销售。  
2、搭配商品的定价策略，2件减多少，3件减多少。  
3、推荐优先策略，价格最低、最省钱。  
4、参考京东推荐模块效果。  
优惠钱数 满XX-X;(适用所有商品)  
优惠数量 满YY-Y;(适用单一商品，钱多的服从钱少的  
就比如 一个牙刷3块，他是满5个-1个的加钱，然后我买了1个牙刷，4个牙膏，牙膏加钱>牙刷，  
那就相当于5个牙刷，还是-去一个牙刷的加钱。)  
特殊规则。但是按照上面如果牙膏也是5-1，那还是减一个牙刷的加钱，  
如果牙膏是6-1，那就不优惠，如果牙膏是4-1，那就要先算牙膏的4-1，因为  
这样省钱多。  
  
拿数量算完了之后再和所有商品的满多少-多少计算，哪个-的多用哪个。

1. **算法分析**

算法的时间复杂度为O(N\*N)

**7.算法实现（这个算法只是单纯的实现最少购物问题，不是整体的）**

class Commodity{  //创建内部类  
  int piece;//购买数量  
  int price;//购买价格  
 }  
 public class ZuiShaoFeiYongGouWu {  
  private static int MAXCODE = 999;//商品编码的最大值  
  private static int SALECOMB = 99;//优惠商品组合数   
  private static int KIND = 5; //商品种类  
  private static int QUANTITY = 5; //购买某种商品数量的最大值  
  private static int b;//购买商品种类数  
  private static int s;//当前优惠组合数  
  private static int[] num = new int[MAXCODE+1];//记录商品编码与商品种类的对应关系  
  private static int[] product = new int[KIND+1];//记录不同种类商品的购买数量  
  private static int[][] offer = new int[SALECOMB+1][KIND+1];//offer[i][j]: 商品组合的优惠价(j=0)；某种优惠组合中某种商品需要购买的数量(j>0)  
  private static Commodity[] purch = new Commodity[KIND+1];//记录不同商品的购买数量和购买价格  
  private static int[][][][][] cost = new int[QUANTITY+1][QUANTITY+1][QUANTITY+1][QUANTITY+1][QUANTITY+1];//记录本次购买的总花费  
  public static void main(String[] args){ //主方法   
   init();  
   comp(1);  
   out();  
  }  
  private static void minicost(){  
   int i,j,k,m,n,p,minm;//i某种优惠组合中某种商品需要购买的量,j商品组合的优惠价,k购买该种商品总数，p每件商品价格( 单买各种商品，使用套餐买各种商品)  
     minm = 0;    
    for(i=1; i<=b; i++)         
      minm += product[i]\*purch[i].price;  
    for(p=1; p<=s; p++){  
      i = product[1] - offer[p][1];  
      j = product[2] - offer[p][2];  
      k = product[3] - offer[p][3];  
      m = product[4] - offer[p][4];  
      n = product[5] - offer[p][5];  
    if(i>=0 && j>=0 && k>=0 && m>=0 && n>=0 &&

cost[i][j][k][m][n]+offer[p][0] < minm)//更新最优值  
      minm = cost[i][j][k][m][n] + offer[p][0];

}

cost[product[1]][product[2]][product[3]][product[4]][product[5]]=minm;  
   }  
 private static void init(){    
   Scanner input = new Scanner(IMG_256System.in);  
    int i,j,n,p,t,code;     
    for(i=0; i<100; i++)        
    for(j=0; j<6; j++)       
    offer[i][j] = 0;  
  for(i=0; i<6; i++){   
    purch[i] = new Commodity();  
    purch[i].piece = 0;  
    purch[i].price = 0;  
    product[i] = 0;  
 }  
    b = input.nextInt();  
    for(i=1; i<=b; i++){  
     code = input.nextInt();  
     purch[i].piece = input.nextInt();  
     purch[i].price = input.nextInt();  
     num[code] = i；  
   }         
    s = input.nextInt();  
     for(i=1; i<=s; i++){  
     t = input.nextInt(); //从键盘输入t  
     for(j=1; j<=t; j++){  
     n = input.nextInt(); //从键盘输入n  
     p = input.nextInt(); //从键盘输入p,此商品组合的优惠价  
     offer[i][num[n]] = p;  
  }  
     offer[i][0] = input.nextInt();//offer[i][0]是第i种优惠组合的价格    }  
   }  
   private static void comp(int i){ //迭代计算！  
   if(i > b){ // 说明B种商品的product[]已经确定,如果i>b,则执行minicost()方法       
     minicost();  
     return;  
   }  
    for(int j=0; j<=purch[i].piece; j++){  
    product[i] = j;//第i种商品取值范围为[0,purch[i]]  
    comp(i+1);   //第i+1种商品  
 }      
}  
    private static void out(){  
    System.out.println(cost[product[1]][product[2]][product[3]][product[4]][product[5]]);//输出最少购物费用  
 }  
}

**8.说明原因及困难：**

原因：一个周的时间要做出关于计算最少购物消费一个小程序，它里面还包含了很多我们没用系统的去学习过的知识，其困难是可想而知的。HTML知识，在制作输入界面的网页时，文字的格式、插入跳转链接等都不能熟练的操作设计。困难：没有专门学过网页设计的课程，对网页设计知识的掌握有所欠缺，制作起来有点困难。例如HTML知识，在制作输入界面的网页时，文字的格式、插入跳转链接等都不能熟练的操作设计。其实上学期的实践周我们做的就是网页设计这方面的内容，但是老师给我们讲的都是一些比较通俗易懂的，比较有效的，在短时间内快速速成学习的知识，我们大部分都是找到自己喜欢的一个网页的模板，然后去爬取它代码，接着根据它的模板将它改成我们想要的样子，把图片和文字更改成自己的，最后进行微调。当然，也有同学自己另外设计的。我们并没有深入的研究过CSS和HTML这些关于网页设计的知识，真是没有什么知识储备。

关于如何用phton去爬取商品的价格信息，我们也是遇到的重大难题，我们的组员认真的去讨论这问题，也在网上找了很多的资料，但是我们大部分都看不懂。我们也问了老师，弄懂了一点，但还是有很多不懂。如：在获取价格的时候，我们已经找到某商品的价格的元素源代码，但不知道怎么在HTML里用getElementById或者用getElementByClassName来获取该商品的价格。在组装URL上也有困难。

**9.解决办法：**

一是利用网络教学资源，认真深入的去学习这方面的知识打好基础，自学网页设计知识，使用Python采集京东或者淘宝网上的数据，再对这些数据代码进行分析，根据题目的要求，运用所学的html和css知识设计产品信息输入页面，最后测试代码；二是请教同学或老师，向他们寻求一些帮助。

**10.实践总结**  
 通过此次实践周我们了解到自己的知识基础很差，专业技能掌握得也不够深。我们也明白了团队之间的合作重要的重要性，明确的分工有利与我们工作的开展。虽然此次实践同的任务我们没有完点，但我们也尽了自己最大的努力去做。我们从中学到了很多的东西，也明白了自己的不足的地方。在今后的学习里我们会查漏补缺，充实自己的，学好专业知识，掌握专业技能。在学好理论知识的同时，我们也会将理论知识与实践结合，运用到实践中去，让自己能学有所用。I