

UI 界面的设计与开发

摘要：近两年来，各式各样的网络电视盒子丰富了电视观众的选择，而各类中间件简化了电视盒子的程序开发，使得普通的程序员也可以在此平台方便的设计开发软件，以满足多样的需求。极大的市场潜力意味着有着大量的相应软件需求，通过设计开发互动点播系统，可以提前接触到这个流行的技术领域，对于我们了解有关电视盒子应用开发有极大的帮助。

关键词：电视盒子；定制芯片；互动；中间件；服务器

Abstract: In the past two years, all kinds of network TV box enriched TV audience's choice, and all kinds of the middleware simplifies the application development of TV box, make ordinary programmers can also in the platform software design and development, to meet the diverse needs. Great market potential means that there are a large number of corresponding software requirements, through the design and development on demand system, can come into contact with the popular technology in advance, for us to understand about TV box application development is of great help.

Key words: Custom chip box; interactive television; middleware; server

1 UI 界面的设计与开发

1.1 编写目的

UI 是 **User Interface**，即用户界面的简称。而 UI 设计指的是对软件的人机交互、操作逻辑以及界面的美观程度的整体设计。一个好的 UI 设计，不仅可以让软件变得更加有个性和品位，还有让软件的操作变得更加人性化，简单舒服的任务。

UI 设计大体包括：一，图形设计，即传统意义上的美工。二，交互设计，也就是软件的操作逻辑，操作流程之类的。三，用户测试，用来测试交互设计的合

理性以及图形设计的美观性。所以这里，就从这三个方面来阐述 界面的设计与开发。

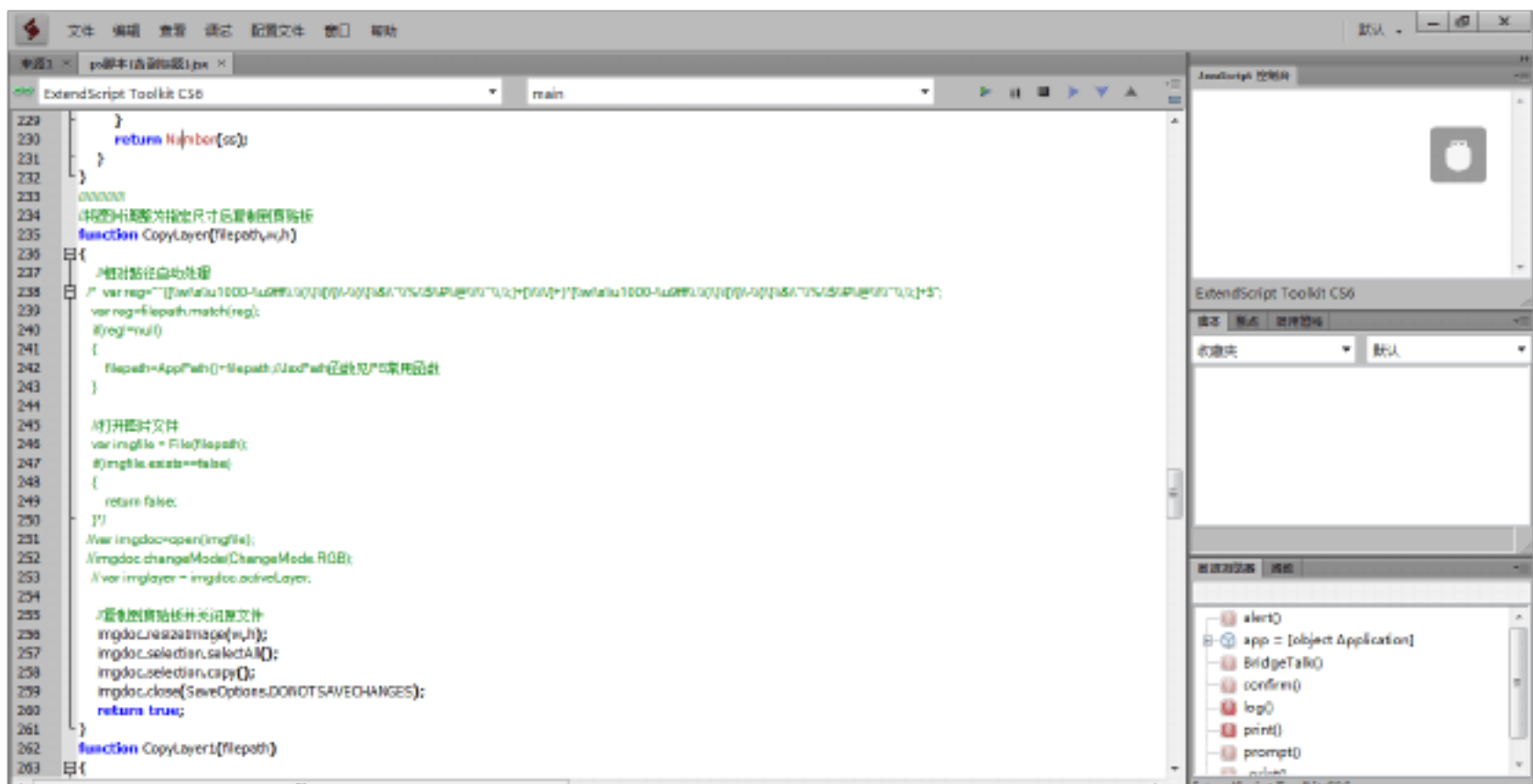
1.2

UI 界面设计的第一步，就是就是图形设计。我们的项目前端主要使用 HTML+CSS来实现效果呈现，所以在图形设计时候，主要使用 PhotoShop 来对图片进行切片和处理，用文本编辑器对图片进行排版设计。因为涉及到大量重复的图片处理步奏，我还特别为此用 applescript 编写了简易的图片批处理脚本，方便在 PhotoShop 上的处理，提高图片设计效率。

使用 PhotoShop，给我们进行图形处理提供了极大的帮助，PhotoShop 的特点有一下这些：



图 在 PS 中进行图形设计



使用 applescript 编写的 PS 脚本

1、像素层级的编辑

像素是组成图片的最基本元素。Photoshop 可以做到像素层级的编辑，功能强大。另外，除了编辑现有图片外，用户还可以用 Photoshop 制作图像，例如加入字体、向量图案等等。

2、图层功能

图层功能让图片既可以分层编辑，也可以合并图层进行统一修改。在同一个图层内，用户可使用 Layer Mask 功能选取套用编辑的部分。

3、动作功能

动作功能可记录一连串的编辑动作，然后应用在不同图片中。

当然 PhotoShop 的缺点也很明显：批量处理不方便。Photoshop 单张图片的编辑处理功能比较强大，其本身也是为处理单张图片而设计。虽然可以使用 camera RAW plug in 来处理 RAW 文件，但 Photoshop 本身的批量处理功能并不算强大。

所以针对这个缺点，我们根据实际需求，为其开发了简易的批处理工具，与 PhotoShop 相结合，就能很好的实现我们对于图形设计的需求了。

第二步，为了在项目中使用设计好的图形，必须借助 HTML+CSS 来呈现效果。HTML 指的是超文本标记语言，标准通用标记语言下的一个应用。超文本”

就是指页面内可以包含图片、链接，甚至音乐、程序等非文字元素。超文本标记语言的结构包括“头”部分（英语：Head）、和“主体”部分（英语：Body），其中“头”部提供关于网页的信息，“主体”部分提供网页的具体内容。CSS指的是层叠样式表，是一种用来表现HTML或XML等文件样式的计算机语言。我们将图形放在HTML的body里，用CSS来控制要呈现的效果，就能实现我们所需要的图形设计效果：

```
<!--节目列表-->
<div id=          style= position:absolute; left:386px; top:258px; width:772px;
height:76px; font-size:20px; color:#ffffff; visibility:hidden; >正在请求数据，请稍等！
</div>
<div style= position:absolute; left:100px; top:189px; width:280px; height:300px; >
    <div class=          ><img id=          src=          width=
height=          />
    </div>
</div>
<div style= position:absolute; left:400px; top:189px; width:480px; height:300px; >
    <div class=          ><img id=          src=          width=
height=          />
    </div>
</div>
</div>
```

节选片段选自项目首页的HTML代码块，是节目列表的图形设计与展示。第一行的“<!--节目列表-->”表示以下模块代码行是属于节目列表功能的函数块。<div>标签是层叠样式表中的定位技术，用来为HTML文档内的内容提供结构和

背景的元素。 中的 id 是 HTML 的选择器,id 选择器允许以一种独立于文档元素的方式来指定样式,在这块代码中,我指定 class 为 的<div>存放设计好的第一张图片,并在 里赋予它 CSS 的属性,使图片对于页面相对定位,并规定内部包含的标签,确定图片的相对路径和相对位置。

大量的图片都是通过这种方式定位在项目上,错综复杂的图片就构成了这个项目的图形设计。

1.3

想要做好 UI 的交互设计,必须要求熟悉网站,能独立完成项目开发,网页设计和相关操作从事 UI 设计网站界面必须知道的知识。现场生产过程,例如,网站整体页面布局和网站知识的本质,能够满足现场整体统一。设计一个网站并不仅仅是把信息,如图片和文本信息,音频和视频的页面如此简单,但是应该开始从用户的实用性,清楚的设计网站的性质,性质不同的设计风格,如政府机构是一个社会人们互相交流,信息公告,发布提供类网站,主要用于政府的公共可以学到一些重要的措施,重要的发布文档,和促进社会道德、伦理等等。

1.3.1 交互设计技术介绍

在该项目里面,我们主要使用 JavaScript 作为主要的交互设计语言,通过 JavaScript 来控制用户的操作与实际函数的映射和逻辑的设计。

JavaScript 一种直译式脚本语言,是一种动态类型、弱类型、基于原型的语言,内置支持类型。它的解释器被称为 JavaScript 引擎,为浏览器的一部分,广泛用于客户端的脚本语言,最早是在 HTML (标准通用标记语言下的一个应用) 网页上使用,用来给 HTML 网页增加动态功能。JavaScript 是一种属于网络的脚本语言,已经被广泛用于 Web 应用开发,常用来为网页添加各式各样的动态功能,为用户提供更流畅美观的浏览效果。通常 JavaScript 脚本是通过嵌入在 HTML 中来实现自身的功能的。

1.3.2 交互设计功能思路

由于本项目的运行平台是电视盒子,所以在用户实际运用时候,是通过遥控

器而不是鼠标或者键盘对其进行操控。所以说，该项目的实际交互设计需要考虑的第一个因素就是如何模拟用户通过遥控器进行自然的操作，而不会感到不正常。

第二个因素是要贴合考虑到操作的方便性和合理性，让用户通过尽可能少的步奏操作，完成想要的动作。

想要解决第一个问题，可以参考曾经智能 DVD 机的操作方式，观察他们的界面安排和操作方式，通过对比和学习，就可以摸索出一套比较通用的操作规范，不同于电脑端或者移动端的操作方式，只要将这套操作规范转换成实际的代码，就能顺利的实现模拟遥控器的操作方式。同时，我还将可能会用到的遥控器功能键以及与其对应的实际键值进行了搜集和整理。

1.1 遥控器按键与对应实际键值

按键描述	实际键值
数字 1	49
数字 2	50
数字 3	51
数字 4	52
数字 5	53
数字 6	54
数字 7	55
数字 8	56
数字 9	57
数字 0	48
* (信息)	848
# (状态/输入法)	572
红	832
黄	834
蓝	835

第二个问题，我们小组进过探讨和研究，决定使用当前社会上比较流行的栅格化原则来进行交互设计。

我们的项目，规定了像素分辨率为 1280*720。所以我们参照 960 栅格化系统的系统指定，设计出了 1280 栅格系统。通过这个 1280 栅格系统，就可以设计出同时具备方便性和合理性的交互逻辑，达到我们的目标预期。

1.3.3 交互设计代码分析

```
function grdEvent(event){
    var keycode = event.which;
    Utility.println(
        +keycode);
    switch(keycode){
        case 1:
            // up
        case 38:
            if(tipShowFlag | moreInfoShowFlag) return 0;
            if(area == 1) {
                lastArea = area;
                changeArea(0);
            } if(area == 2 && listPos < 3) {
                lastArea = area;
                changeArea(1);
            } else if(area == 2 && (listPos == 3 | listPos == 4)) {
                changeListPos(-3);
            }
            else if(area == 2 && listPos == 5) {
```

```
        changeListPos(-1);  
    } else if (area == 2 && listPos == 6) {  
        changeListPos(-4);  
    } return 0;  
    break;  
}
```

节选的函数 `grdEvent` 代码块主要负责接收用户所按按键的真实键值，并执行对应的响应函数。通过这个函数，用户每次按键之后想要的反应都会实时响应，达到用户的需求。下面具体介绍代码的功能实现：

`var keycode = event.which;`”底层硬件集成的中间系统会通过接收器接收到用户通过遥控器按下的键位，并将具体键值传递给 `event`，并通过变量 `keycode` 接收。`Utility.println(` `+keycode);`
”`Utility.println` 方法可以将后面括号内的内容在串口输出打印。

接下来的 `switch(keycode){...}` 是编程语言经典的多分枝选择语句，准确的使用 `switch...case...` 语句能有效的避免冗长的 `if..elseif..else` 代码块。圆括号内的 `keycode` 就是指前文所指的真正键值，通过一条条匹配对应键值的函数方法即可实现用户需求。

代码块中出现大量类似 `if(tipShowFlag || moreInfoShowFlag) return 0;` 的语句，`return 0` 的作用是跳出当前循环，而放在 `if()` 的后面意思就是当括号内成立时候，就跳出当前语句。作用就是在执行对应键值操作的方法之前先判断，当前状态是否不能响应这个动作，比如正在加载数据或者有上层 `panel` 覆盖的时候，就不能响应确认按键的动作。如果不加这种判断，就很有可能出现逻辑错误，影响用户使用体验。这同时也是开发过程中最经常出现的问题之一，需要有清晰的思路已经多次的检查排漏才能确保不会出现此类问题。

```

    changePos(_num){
if(area == 1) {
    menuPos += _num;
    if(menuPos < 0) menuPos = 6;
    if(menuPos > 6) menuPos = 0;
    $(      ) style.left = 99 + 158 * menuPos +      ;
} else if(area == 0) {
    lastPos = menuPos;
    menuPos += _num;
    if(menuPos < 7) menuPos = 10;
    if(menuPos > 10) menuPos = 7;
    $(lastPos).style.visibility =      ;
    $(menuPos).style.visibility =      ;
}
}

function changeArea(_area){
    area = _area;
    if(area == 0) {
        if(lastArea == 1 || menuPos != 6) {
            menuPos = 7;
            $(      ) style.visibility =      ;
            $(menuPos).style.visibility =      ;
        }
    } else if(_area == 1) {
        area = _area;
        $(      ) style.visibility =      ;
    }
}

```

```

if(lastArea == 0 && (menuPos == 7 | menuPos == 8)) {
    lastPos = menuPos;
    menuPos = 5;
    $(      ) style.left =      ;
    $(lastPos).style.visibility =      ;
} else if(lastArea == 2 && listPos == 2) {
    lastPos = listPos;
    menuPos = 5;
    $(      ) style.left =      ;
    $(      +lastPos).style.visibility =      ;
}
}
}

```

节选代码块包含两个有关交互设计的函数 `changePos` 和 `changeArea`，从名字就可以很容易看出这两个函数是移动焦点位置和切换显示区域的函数。



用 `area` 来为项目划分区域

我们的项目，是将页面分成多个区域，用 `area` 来区分。因为每个区域的交互逻辑都有不同，划分区域就可以清楚的将不同的逻辑代码放在不同的地方，一方

面可以提高开发效率，减少开发时间。二是代码分离开，方便排查错误，这样当测试时候是在哪一区域出现逻辑交互错误，就可以快速定位代码位置，知道错误的大体位置。

同理，在每个区域里的每个相对位置我们用 `XXXPos` 来表示，比如区域 `area0` 中有四个区域（搜索、收藏、历史、首页），于是我们就用 `menuPos0~menuPos3` 来标示。这样，`if(menuPos < 0) menuPos = 3; if(menuPos > 3) menuPos = 0;` 就可以表示在循环中，如果当前焦点位置从第四个首页再往右移动，即 `menuPos=3` 再加1，就自动切换回最左边的搜索，即 `menuPos=0`。这种切换方式代码量非常少，而且逻辑清晰，非常适合在我们这种简单的小程序里面使用。在项目中，大量这种类型的代码的调用，构成了项目最基础的交互移动系统，互相联系又不相互影响，强化了系统的鲁棒性。

`$().style.visibility =` 和 `$().style.visibility =` 是通过html控制括号内对象类选择器的CSS。顾名思义可以看得出来，`visibility`指代的是对象是否可见。通过对其属性进行赋值`hidden`和`visible`，就可以轻松实现控制对象的出现和隐藏。如果同时让相邻的两个相同元素一个显示，一个消失，则可以出现仿佛是这个元素从一个位置移动到另一个位置的假象。这是一种编程的技巧，我们可以在每个焦点可能会移动的位置都事先隐藏一个焦点元素，这样，只要当系统监测到用户想要移动焦点位置的意图时候，通过`hidden/visible`的方法，就可以轻松实现。与之相对应的是

```
$( ) style.left = 396 + 150 * listPos + ;
```

```
$( ) style.top = ;
```

这种方法，只生成一个焦点元素，每次监测到用户移动焦点意图的时候，去计算焦点坐标变化，再让焦点移动到结束位置。这种做法需要在每次焦点移动时候做全面判断移动规则，然后用对应的变换函数去计算移动后的坐标位置。

这两种做法优缺点非常明显。前者优点是没有任何延迟，在我们项目不支持多线程的硬件基础上，同时多个元素的移动不会出现卡顿或延迟的感觉，缺点就

是前端页面代码比较冗长，每个对位都会嵌入同样的焦点图片标签。后者优点是前端代码清楚，只有一个个位置，没有大量重复的代码。但是缺点是后台逻辑部分有大量判断计算函数，而且开发难度较大，容易出现错误和 。最佳的方法是两种相结合，移动规则单一且量大的时候用第二种方法，位置少但是移动不规律时候后第一种方法。这样就可以很好的实现交互设计的需求了。

网站界面的设计从以前的只重技术不重设计正逐渐改善为两者兼顾，设计的重要性和技术性是同等重要的。如果只重技术而设计的不好，那么即使技术性再强也达不到好的用户体验效果，从而不能满足一个网站总的功能的发挥，这样就影响了网站的价值，只有将技术性和设计效果结合起来才能达到更好的用户体验，更合理的人机交互，实现网站存在的价值。因此，网站界面的 UI 设计对于网站的建设非常重要。

我们的项目很大程度是靠视觉呈现给用户使用与感受的，所以UI开发也是重中之重。前期我们做了大量的讨论与准备，研究过当前市场上比较流行的项目和开发方式，在开发中也是慢慢摸索，常用的工具有如PhotoShop、HBuilder之类的都是在使用中渐渐熟悉的。不过，在探索和实践中，我们也深切体会到UI设计的重要性，同时也能感受到用户正真的需求，设计出最合适的交互方式，也为接下来的页面逻辑设计打好了基础。

2 页面逻辑的设计与实现

2.1 编写目的

逻辑设计就是把一种计划、规划、设想通过视觉的形式通过概念、判断、推理、论证来理解和区分客观世界的思维传达出来的活动过程。逻辑设计比物理设计更理论化和抽象化，关注对象之间的逻辑关系，提供了更多系统和子系统的详细描述。

而我们的项目最终完成之后也是交由用户使用，一个完整且没错误的页面逻辑会是用户的使用体验达到完美，所以页面逻辑设计也是项目里非常重要的一个部分，在本段会介绍我们项目主要页面逻辑的设计与实现。

网格是用竖直或水平分割线将布局进行分块，把边界，空白和栏包括在内，以提供组织内容的框架的辅助性工具。设计中不能忽视的网格，从印刷物料到网页页面，还是最新的 Windows8 系统界面，无论我们的设计如何，网格在设计中必然存在。网格可以使页面布局显得紧凑而且稳定，为设计师在设计站点时提供一个逻辑严谨的模板。网格也可以使版式设计有连贯性和韵律，让那些看上去相当小并且毫无联系的元素也可以在网格的布局里占有自己的一席之地。但是网格不只是让设计师一味地按照规律来枯燥的设计，只是在一个合理的运用下构造和设计出好的作品来，适时地打破规则也能有意外的惊喜。

2.3 数据传递的逻辑设计

我们项目的数据是通过 HTTP 的 **ajax** 与服务器进行交互，传递数据的格式主要由 **Xml** 转 **Json** 来进行。

ajax 是指一种创建交互式网页应用的网页开发技术。通过在后台与服务器进行少量数据交换，**AJAX** 可以使网页实现异步更新。这意味着可以在不重新加载整个网页的情况下，对网页的某部分进行更新。传统的网页（不使用 **AJAX**）如果需要更新内容，必须重载整个网页页面。**Ajax** 的核心是 **JavaScript** 对象 **XmlHttpRequest** 该对象在 **Internet Explorer 5** 中首次引入，它是一种支持异步请求的技术。简而言之，**XmlHttpRequest** 使您可以使用 **JavaScript** 向服务器提出请求并处理响应，而不阻塞用户。

Http 定义了与服务器交互的不同方法，最基本的方法有4种，分别是 **GET**, **POST**, **PUT**, **DELETE**。**URL** 全称是资源描述符，我们可以这样认为：一个 **URL** 地址，它用于描述一个网络上的资源，而 **HTTP** 中的 **GET**, **POST**, **PUT**, **DELETE** 就对应着对这个资源的查，改，增，删4个操作。到这里，大家应该有个大概的

了解了， 一般用于获取/查询资源信息，而 POST 一般用于更新资源信息。

```
function getListProgData() {  
    listProgs = [];  
    //clearListPrograms();  
    if (listAjaxObj == null) {  
        listAjaxObj = new ajaxClass();  
        listAjaxObj.frame = window;  
    } else {  
        listAjaxObj.requestAbort();  
    }  
    $(  
    ) style.visibility =  
    ;  
    listAjaxObj.successCallback = function(_xmlHttp, _params) {  
        var data = parseDom(this.getResponseXML());  
        var jsonData = parseJson(data.toJson());  
        if (typeof(jsonData) != | jsonData == null) {  
            $(  
            ).innerText = 获取数据失败! ;  
            $(  
            ) style.visibility =  
            ;  
            if (!oneRequestFinish) oneRequestFinish = true;  
            else $(  
            ) style.visibility =  
            ;  
            return; }  
        var key =  
        ;  
        if (key in jsonData) {  
            var totalResult = parseInt(jsonData.FolderContents[0].totalResults, 10);  
            listServiceId = jsonData.FolderContents[0].FolderFrame[0].serviceId;  
            listChargeMode =  
            jsonData.FolderContents[0].FolderFrame[0].chargeMode;
```



```

if (totalResult > 0) {
    /理论上来说应该没有电影返回，可以不用考虑。

    var series = jsonData.FolderContents[0].ChildFolder;
    var movies = jsonData.FolderContents[0].SelectableItem;
    /添加一个属性标识是电视剧还是电影

    if (typeof series != 'undefined') {
        for (var i = 0; i < series.length; i++) {
            series[i].type = 'TV';
        }
    } else series = [];

    if (typeof movies != 'undefined') {
        for (var i = 0; i < movies.length; i++) {
            movies[i].type = 'Movie';
        }
    } else movies = [];

    listProgs = series.concat(movies);
    //hashTableObj.put(_params[0], listProgs);

    currPage = Number(currPage);
    initListPrograms();

    $( "#listProgs" ).style.visibility = 'visible';
    $( "#errorMsg" ).style.visibility = 'hidden';
} else if ( "message" in jsonData) {
    var message = jsonData.NavServerResponse[0].message;
    var code = jsonData.NavServerResponse[0].code;
    $( "#errorMsg" ).innerText = "发生错误！错误码：" + code;
    $( "#listProgs" ).style.visibility = 'hidden';

    if (!oneRequestFinish) oneRequestFinish = true;
} $( "#errorMsg" ).style.visibility = 'visible';

```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/466134025121010121>