

实验一报告

程序语言：html、css、vue

姓名：郑子睿

学号：22920212204317

专业：计算机科学与技术

一、问题描述

随机点名程序（越不来上课的人，被点中的概率越高；实现抽查问题、预警等功能）

二、算法思想

本实验采用html和CSS来实现网页的基本结构和样式，用VUE来实现网页的交互。

数据结构设计

```
1  学生对象
2  {
3      id        // 学号
4      name      // 名字
5      pro       // 缺勤次数
6      email     // 该学生的邮箱
7  }
8  current        // 当前点到的学生
9  initialData    // 初始的从文件中导入的数据
10 selectedList   // 已点名列表
11 absentList     // 缺勤列表
12 candidateList  // 待点名名单
13 randomList     // 用于随机点名的列表
```

实现随机点名

算法流程：

1. 每次按这样的方式生成randomList：遍历待点名名单candidateList，对于第*i*个元素向randomList中加入其缺勤次数+1个该元素。
2. 用js中的random方法生成一个在 $[0, randomList.length - 1]$ 的随机整数，将该整数作为下标对应randomList中的对象赋给current。
3. 当用户点击名字或者“开始/暂停”按钮之后就会暂停随机函数，此时的current就是被点到的名字。
4. 名字出来之后会在名字下方弹出一个框，选择是否缺勤。
 - 如果选择“再选一个”，则将该名字加入已点名单，并在待选名单中将这个名字删除。
 - 如果选择“缺勤”，则将该名字加入已点名单和缺勤名单，并在待选名单中将这个名字删除。

设置名单

导入一个txt，txt中有四列：学号，姓名，缺勤次数，邮箱，将文件中的数据解析为一个对象数组，存入 `initialData` 和 `candidateList` 中。

清空数据

将 `selectedList`、`randomList`、`candidateList`、`absentList` 都置为0，并将 `current` 显示为 请导入名单。

差别显示和预警

设定当学生缺勤三次，即取消考试资格。

在页面显示的缺勤名单中：

- 如果该生的缺勤次数小于3次，则显示为黄色，表示“警告”；
- 如果该生的缺勤次数等于3次，则显示为红色，表示“危险”；
- 如果该生的缺勤次数多于3次，则显示为灰色，表示“弃考”

导出数据

点击“导出数据”按钮，会将缺勤次数小于4次的学生的名单导出成txt文件，浏览器直接下载。

三、描述算法

```
1 // 滚动名字
2 scrollName() {
3   if 待选名单为空
4     this.scrollStatus <- false;           // 如果待选名单为空，则将滚动状态置为
False, 表示停止滚动
5     return;
6   else
7     if randomList不存在 // 如果还没有生成randomList，则生成
8       for i in candidateList // 遍历待选名单
9         for (let i = 0; i <= item.pro; i++) // 向randomList中加入
pro+1个对象
10          this.randomList.push(item);
11
12   setTimeout(() => { // 用计数器方法不断递归实现名字一直变化
13     let res = this.getRandom(0, this.randomList.length - 1);
14     this.current <- this.randomList[res];
15     if (this.scrollStatus) {
16       this.scrollName();
17     }
18   }, 100);
19 }
20
21 // 选中名字
22 selectName() {
23   if (当前正在滚动) {
24     this.scrollStatus = false; // 将scrollStatus置为false，停止滚动
25     setTimeout(() => {
26       this.selectedList.push(this.current); // 将当前选中的名字加入已选
名单
27       this.randomList = []; // 将randomList清空
28       this.candidateList = this.candidateList.filter(item => item !==
this.current);
```

```

29         }, 100) // 将当前名字从待选名单中清空
30     } else {      // 否则开始滚动
31         this.scrollStatus = true;
32         this.scrollName();
33     }
34 }

```

四、验证算法

具体代码

1. html的基本架构

```

1  <body>
2      <div id="app">
3          <main>
4              <div class="nameboard">
5                  <h1 @click="selectName">{{current.name}}</h1>
6                  <transition name="checkButton">
7                      <div v-show="!finishCheck && !scrollStatus"
class="absentCheck">
8                          <el-button type="danger" plain @click="isAbsent">缺
勤</el-button>
9                          <el-button type="success" plain
@click="selectName">再点一个</el-button>
10                     </div>
11                 </transition>
12             </div>
13             <div class="toolbar">
14                 <button @click="chooseFile">设置名单</button>
15                 <button @click="selectName">开始/停止</button>
16                 <button class="clearButton" @click="reset">清空数据</button>
17                 <input type="file" id="input" @change="handleFiles($event)"
accept=".txt" style="display: none;">
18                 <button @click="exportData">导出数据</button>
19             </div>
20             <div class="namelist">
21                 <div class="selectedStu">
22                     <p>已点名单</p>
23                     <ul>
24                         <li v-for="(u, i) of selectedList" :key="i">
{{u.name}}
25                     </li>
26                 </ul>
27             </div>
28             <div class="absentStu">
29                 <p>缺勤名单</p>
30                 <ul>
31                     <li v-for="(u, i) of absentList" :key="i"
: class="[u.pro > 3 ? 'grey' : '', u.pro <= 2 ?
'yellow' : '', u.pro === 3 ? 'red' : '']"
32                         style="border-radius: 15px; color: #fff; font-
weight: 700;">
33                         {{u.name}}
34                     </li>
35                 </ul>
36             </div>
37         </main>
38     </div>
39 </body>

```

```

38         </div>
39         <div class="candidate">
40             <p>特点名列表</p>
41             <ul>
42                 <li v-for="(u, i) of candidateList" :key="i">
43                     {{u.name}}
44                 </li>
45             </ul>
46         </div>
47     </div>
48 </main>
49 </div>
50 </body>

```

2. css样式表

```

1  * {
2      margin: 0;
3      padding: 0;
4  }
5  html,
6  body {
7      font-family: "Microsoft YaHei";
8  }
9  ul {
10     margin: 0;
11     padding: 0;
12     list-style-type: none;
13 }
14
15 h1 {
16     font-size: 100px;
17 }
18
19 main {
20     margin: 0 auto;
21     width: 1200px;
22     margin-top: 20px;
23     margin-bottom: 150px;
24     background-color: #fefefe;
25 }
26
27 main .nameboard {
28     height: 510px;
29     position: relative;
30     /* border: 1px solid blue; */
31     display: flex;
32     flex-direction: column;
33     justify-content: center;
34     align-items: center;
35     border-bottom: 3px solid #cacaca;
36     background-color: #fff;
37 }
38
39
40 main .nameboard .absentCheck {
41     position: absolute;

```

```
42     bottom: 100px;
43     margin-top: 70px;
44 }
45
46 .checkButton-enter-active, .checkButton-leave-active {
47     transition: opacity .5s;
48 }
49 .checkButton-enter, .checkButton-leave-to {
50     opacity: 0;
51 }
52
53 .toolbar {
54     margin-top: 25px;
55     margin-bottom: 20px;
56 }
57
58 button {
59     vertical-align: middle;
60     margin: 0 10px 1rem 0;
61     font-family: inherit;
62     padding: .85em 1em;
63     -webkit-appearance: none;
64     border: 1px solid transparent;
65     border-radius: 0;
66     transition: background-color .25s ease-out,color .25s ease-out;
67     font-size: .9rem;
68     line-height: 1;
69     text-align: center;
70     cursor: pointer;
71     background-color: #1779ba;
72     color: #fefefe;
73 }
74
75 .clearButton {
76     background-color: #767676;
77 }
78
79 .namelist div {
80     margin-bottom: 5px;
81 }
82
83 .namelist p {
84     color: #666;
85     margin-bottom: 20px;
86 }
87
88 .namelist ul {
89     display: flex;
90     flex-wrap: wrap;
91 }
92
93 .namelist li {
94     display: inline-block;
95     border: 1px solid #eaeaea;
96     color: #14679e;
97     line-height: 2;
98     padding: 0 10px;
99     margin-right: 10px;
```

```

100     margin-bottom: 10px;
101     cursor: pointer;
102 }
103
104 .namelist .red {
105     background-color: rgba(245, 108, 108, 1);
106 }
107
108 .namelist .yellow {
109     background-color: rgba(230, 162, 60, 1);
110 }
111
112 .namelist .grey {
113     background-color: rgba(118, 120, 122, 0.4);
114 }

```

3. vue的数据域

```

1  data: {
2      current: {name: '请导入名单'},
3      initialData: [],
4      selectedList: [],
5      absentList: [],
6      candidateList: [],
7      scrollStatus: true,
8      randomList: [],
9      finishCheck: true,
10 }

```

4. 导入文件

```

1  chooseFile() {
2      let inputFile = document.querySelector('#input');
3      inputFile.click();
4  },
5
6  handleFiles(e) {
7      let file = e.target.files[0];
8      if (file.type !== "text/plain") {
9          this.$message.error('只可读取txt格式的文件! ');
10         return;
11     }
12
13     let reader = new FileReader();
14     reader.readAsText(file, 'utf-8');
15     reader.onload = () => {
16         this.$message({
17             message: '文件导入成功! ',
18             type: 'success'
19         });
20         let res = reader.result;
21         res.split('\r\n').forEach((u) => {
22             if (u) {
23                 let array = u.split(' ');
24                 let obj = {
25                     id: array[0],

```

```

26         name: array[1],
27         pro: Number(array[2]),
28         email: array[3]
29     };
30     this.candidateList.push(obj);
31 }
32 })
33 // console.log(this.candidateList);
34 this.initialData = this.candidateList;
35 this.scrollStatus = true;
36 this.scrollName();
37 }
38 reader.onerror = () => {
39     this.$message.error('文件导入失败');
40 }
41 },

```

5. 随机函数

```

1 // 随机函数左闭右闭
2 getRandom(min, max) {
3     return Math.floor(Math.random() * (max - min + 1) + min);
4 },

```

6. 名字滚动函数

```

1 // 实现名字的变化
2 scrollName() {
3     if (this.candidateList.length === 0) {
4         this.scrollStatus = false;
5         this.$message.error('名单是空的');
6         return;
7     } else {
8         if (this.randomList.length === 0) {
9             this.candidateList.map((item) => {
10                 for (let i = 0; i <= item.pro; i++) {
11                     this.randomList.push(item);
12                 }
13             })
14         }
15         setTimeout(() => {
16             let res = this.getRandom(0, this.randomList.length - 1);
17             this.current = this.randomList[res];
18             if (this.scrollStatus) {
19                 this.scrollName();
20             }
21         }, 100);
22     }
23 },

```

7. 停止滚动点名

```

1 selectName() {
2     if (this.scrollStatus) {
3         this.scrollStatus = false;

```

```

4         this.finishCheck = false;
5         setTimeout(() => {
6             this.selectedList.push(this.current);
7             this.randomList = [];
8             this.candidateList = this.candidateList.filter(item => item !==
this.current);
9         }, 100)
10     } else {
11         this.scrollStatus = true;
12         this.scrollName();
13     }
14 },

```

8. 缺勤

```

1  isAbsent() {
2      console.log(this.current.id);
3      this.current.pro++;
4      this.absentList.push(this.current);
5      this.finishCheck = true;
6      this.scrollStatus = true;
7      this.scrollName();
8  },

```

9. 清空数据

```

1  reset() {
2      this.selectedList = [];
3      this.randomList = [];
4      this.candidateList = [];
5      this.absentList = [];
6      this.finishCheck = true;
7      this.current.name = '请导入名单';
8  },

```

10. 导出数据

```

1  exportRaw(name, data) {
2      var urlObject = window.URL || window.webkitURL || window;
3      console.log(data);
4      var export_blob = new Blob(data);
5      var save_link =
document.createElementNS("http://www.w3.org/1999/xhtml", "a")
6      save_link.href = urlObject.createObjectURL(export_blob);
7      save_link.download = name;
8      this.fakeClick(save_link);
9  },
10  fakeClick(obj) {
11      var ev = document.createEvent("MouseEvent");
12      ev.initMouseEvent("click", true, false, window, 0, 0, 0, 0, 0,
false, false, false, false, 0,
13          null);
14      obj.dispatchEvent(ev);
15  },
16  exportData() {

```

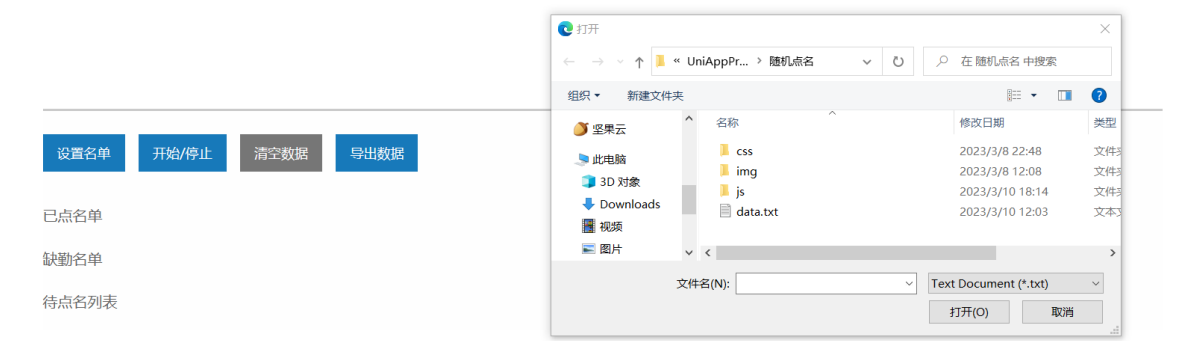


```
17     let strArray = [];  
18     this.initialData.map((u) => {  
19         if (u.pro <= 3)  
20             strArray.push(`${u.id} ${u.name} ${u.pro} ${u.email}\\r\\n`);  
21     })  
22     this.exportRaw('nameList.txt', strArray);  
23     this.$message({  
24         message: '名单导出成功',  
25         type: 'success'  
26     });  
27 },
```

结果展示

1.导入文件

请导入名单



2. 名字滚动

金育木



3. 点击名字暂停，出现选项

郭芳天

缺勤

再点一个

设置名单

开始/停止

清空数据

导出数据

已点名单

郭芳天

4. 点击缺勤后加入缺勤名单，差别显示

缺勤名单

郭芳天

吴美隆

吴思翰

林孟霖

吴韵如

蔡书玮

陈萱珍

彭郁婷

5. 导出文件，浏览器自动下载

下载

nameList.txt

打开文件

6. 导出文件和导入文件格式相同

导入文件

```
1 张吉惟 0 22920212204317@stu.xmu.edu.cn
2 林国瑞 2 2033541709@qq.com
3 林玟书 1 zheng2003jerry@126.com
4 林雅南 1 2033541709@qq.com
5 江奕云 0 zheng2003jerry@163.com
6 刘柏宏 3 22920212204317@stu.xmu.edu.cn
7 阮建安 1 2033541709@qq.com
8 林子帆 0 2033541709@qq.com
9 夏志豪 1 22920212204317@stu.xmu.edu.cn
```

导出文件

nameList (1).txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
1 张吉惟 0 22920212204317@stu.xmu.edu.cn
2 林国瑞 2 2033541709@qq.com
3 林玟书 1 zheng2003jerry@126.com
4 林雅南 1 2033541709@qq.com
5 江奕云 0 zheng2003jerry@163.com
6 刘柏宏 3 22920212204317@stu.xmu.edu.cn
7 阮建安 1 2033541709@qq.com
8 林子帆 0 2033541709@qq.com
9 夏志豪 1 22920212204317@stu.xmu.edu.cn
10 吉茹定 2 zheng2003jerry@163.com
```

导入文件原来的数据，在点名时缺勤

```
42 李成白 2 22920212204317@stu.xmu.edu.cn
```

导出的文件中缺勤次数+1

```
42 李成白 3 22920212204317@stu.xmu.edu.cn
```

7. 清空数据

名单是空的

请导入名单

设置名单

开始/停止

清空数据

导出数据

已点名单

缺勤名单

待点名列表

五、结论

该实验的难点在于不等概率点名，主要是在一个数组中每个学生的数量不同，使用随机数方法生成随机下标抽取。

本实验的亮点是使用了前端的一些技术实现了功能的可视化。