|  |
| --- |
|  |
|  |

**MBTI职业性格测试系统**

**需求&设计说明书**

**拟 制 人：**

**审 核 人：**

**[]**

**北京汇才同飞教育科技有限公司**

**文件变更记录**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 版本号 | 日期 | 修改人 | 摘 要 | 审核人 | 备注 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

1 引言 1

1.1 编写目的 1

2 系统需求 1

2.1 系统流程 1

2.2 答卷 2

3 系统设计 3

3.1 项目名称 3

3.2 系统分析 3

3.3 核心技术 4

3.3.1 实体类设计 4

3.3.2 文件数据格式 6

3.3.3 解析json技术Gson 7

3.4 包结构 8

3.5 接口和类的详细设计 11

3.5.1 表现层App类描述 11

3.5.2 逻辑层描述 12

3.5.3 数据层描述 14

4 项目拓展 15

4.1 拓展项 15

4.1.1 时间控制 15

4.1.2 随机出题 15

# 引言

## 编写目的

**<文档主要描述了此项目目前阶段需求，为详细设计或者开发提供设计和开发依据>**

# 系统需求

## 系统流程



## 答卷

参考《MBTI职业性格测试题》

# 系统设计

## 项目名称

MBTI

## 系统分析

题目文件

List<题目>

循环展示题目

收集结果,将结果保存在List

[T,I,P,E,T,P…………]

分析结果，得到结论：ISTP

根据ISTP选择性格文本

结果文件

List<结果>

## 核心技术

### 实体类设计

|  |  |  |
| --- | --- | --- |
| 问题类：Question | | |
| 属性名 | 类型 | 说明 |
| num | int | 题目序号 |
| title | String | 题目标题(题目内容) |
| options | List<Option> | 题目选项 |

|  |  |  |
| --- | --- | --- |
| 选项类：Option | | |
| 属性名 | 类型 | 说明 |
| title | String | 选项内容 |
| score | String | 选项得分 J或P |

题目文件的内容：

|  |
| --- |
| 当你要外出一整天，你会#计划你要做什么和在什么时候做%J#说去就去%P  ………………  ……………… |

93行

|  |
| --- |
| Question q=new Question();  q.setNum(1);  q.setTitle(“当你要外出一整天，你会”);  Option o1=new Option();  o1.setTitle(“计划你要做什么和在什么时候做”);  o1.setScore(“J”);  Option o2=new Option();  o2.setTitle(“说去就去”);  o2.setScore(“P”);  List<Option> lst=new ArrayList<Option>();  lst.add(o1);  lst.add(o2);  q.setOptions(lst);  一个题目的所有信息都在对象q中 |

|  |  |  |
| --- | --- | --- |
| 结果类：Result | | |
| 属性名 | 类型 | 说明 |
| num | String | 结果编号：ISFJ |
| content | String | 结果内容 |

### 文件数据格式JSON

参考《MBTI职业性格测试题》.pdf将考题生成json数据文件

创建question.txt文件

[ {

num : 1,

title : "当你要外出一整天，你会",

options : [ {

title : "计划你要做什么和在什么时候做",

score : "J"

}, {

title : "说去就去",

score : "P"

} ]

}, {

num : 2,

title : "你认为自己是一个",

options : [ {

title : "较为随兴所至的人",

score : "P"

}, {

title : "较为有条理的人",

score : "J"

} ]

}]

创建result.txt文件

[ {

num : "ESTP",

content: "1.擅长现场实时解决问题— 解决问题者。 \n2.喜欢办事并乐于其中及过程。"

}, {

num : "ISFP",

content: " 1.外向、和善、接受性、乐于分享喜乐予他人。 \n2.喜欢与他人一起行动且促成事件发生，在学习时亦然。"}

]

### 解析json技术Gson

#### 创建Gson实例

Gson gson = **new** Gson();

#### 将JSON格式的字符串转换成对象

String str=”{num:1,title:"当你要外出一整天，你会",options:[{title:" 计划你要做什么和在什么时候做，",score:"J"},{title:"说去就去",score:"P"}]}”;

Question question= gson.fromJson(str, Question.**class**);

此写法，在本项目中不使用。

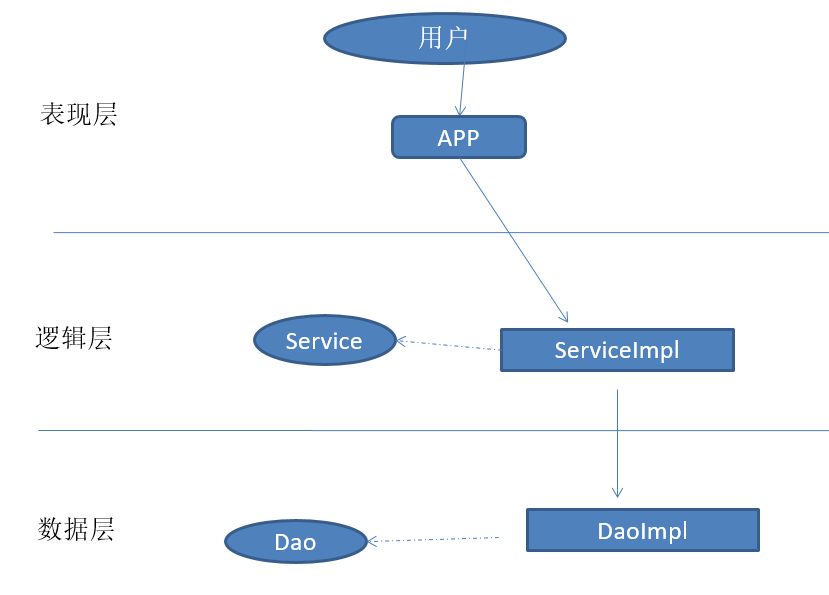
#### 将JSON格式的字符串转换成List集合

String str="[{},{},{},{},…………]";

List<Question> qs  
=gson.fromJson(str,**new** TypeToken<List<Question>>(){}.getType());

## 包结构

三层架构



白色：包

黄色：类

红色：接口

## 接口和类的详细设计

### 表现层App类描述

#### App

App类是系统的入口，包含main方法。

#### 方法

|  |  |  |  |
| --- | --- | --- | --- |
| 方法名 | 输入(参数) | 输出(返回值) | 说明 |
| main |  |  | 系统的入口  创建App类实例，调用start方法 |
| start |  |  | **系统流程的控制方法：**  **1、读取题目信息**  **2、展示题目信息**  **3、收集用户答案**  **4、展示性格分析结果** |

### 逻辑层描述

#### Service接口

Service是逻辑层的接口，包含的方法如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 方法名 | 输入(参数) | 输出(返回值) | 说明 |
| readQuestion |  | List<Question>  封装题目信息的List对象 | 读取题目内容 |
| parseResult | List<String>  封装用户答案的List对象（包含93个字母） | String  用户性格分析结果 | 根据用户答案分析得到性格结果 |

#### ServiceImpl实现类

ServiceImpl类是逻辑层接口的实现类，实现了Service接口。

逻辑层主要实现了系统的业务逻辑功能。

#### ServiceImpl方法

|  |  |  |  |
| --- | --- | --- | --- |
| 方法名 | 输入(参数) | 输出(返回值) | 说明 |
| readQuestion |  | List<Question>  封装题目信息的List对象 | 1. 调用数据层读取文件中的json数据 2. 将json数据解析成封装Question对象的List对象 3. 返回List对象 |
| parseResult | List<String>  封装用户答案的List对象 | String  用户性格分析结果 | 1. 传入封装用户答案的List对象 2. 统计List对象中的数据，得到【性格分类】 3. 调用数据层读取Result信息 4. 根据【性格分类】在Result信息中找到对象的分析结果，并返回 |

### 数据层描述

#### Dao接口

Dao是数据层的接口，包含的方法如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 方法名 | 输入(参数) | 输出(返回值) | 说明 |
| read | String  读取文件的路径 | String  文件中的文本信息 | 读取文件内容 |

#### DaoImpl

DaoImpl类是数据层接口的实现类，实现了Dao接口。

数据层主要实现了数据访问业务，通过读取磁盘中的文件或访问数据库得到系统需要的数据信息。

#### DaoImpl方法

|  |  |  |  |
| --- | --- | --- | --- |
| 方法名 | 输入(参数) | 输出(返回值) | 说明 |
| read | String  读取文件的路径 | String  文件中的文本信息 | 1. 读取文件信息，得到字节数组 2. 将字节数组转换成字符串，并返回 |

# 项目拓展

## 拓展项

### 随机出题

可以打乱题目顺序，随机出题。

### 输入校验

用户只能输入ABab，如果输入错误，需要提示错误，重新录入；

如果三次输入错误，测试停止，重新开始测试。

### 测试结果持久化

测试完成后，询问用户是否保存测试结果；

如果用户选择“保存测试结果”，将测试结果保存到指定路径（项目路径/save目录）中，文件类型为txt，文件名称为“MBTI职业性格测试20200313用户姓名”；

“20200313”为当前年月日的组合，用户姓名为进入系统时，用户输入的姓名。

### 展示测试历史

用户进入系统后，展示测试历史。

要求展示最后三次保存的测试记录。

展示效果如下：

1. 张三 20200315
2. 李四 20200312
3. 王五 20200313

按测试时间倒序展示。

不足三次的，有几次，展示几次。

用户在控制台输入序号后，展示性格测试结果。

### 时间控制

要求：答题时间控制在30分钟内:当用户输入选项时，距开始时间超出了30分钟，提示测试结束。

System.exit(0)

### 返回上一题目

在用户答题过程中，已经完成第5题作答，但是想重新选择第5题答案，可以输入【重选】，重新输入第5题答案。

完成操作后，继续展示第6题，继续答题

### 返回指定题目，修改答案

在用户答题过程中，已经完成第5题作答，但是想重新选择第3题答案，可以输入【重选+题号】（例如：重选3），重新输入第3题答案。

完成操作后，继续展示第6题，继续答题