# 单项选择题测试系统函数设计

# 详细设计

1.1类设计

#### 1.1.1基本用户类

类属性

受保护属性：

身份定义 **identity** 类型**string**

用户名 **userName** 类型 **string**

密码 **password** 类型 **string**

类方法

公共方法：

bool changePassword(string userName, string oldPassword)

{

//参数

//userName 用户名， oldPassword 旧密码

//与系统数据文件内的用户名密码去核验旧密码的准确性

//密码正确，要求用户输入新密码并确认新密码，修改成功返回true，修改失败返回false

//密码错误，拒绝修改，返回false

}

string getUserName()

{

//返回用户名  
}

string getpassword()

{

//返回密码  
}

bool cancelAccount(int instruct)

{

//参数 instruct 指令

//指令 0 取消注销，自动返回上一级

//指令 1确认注销，进行文件操作

//执行成功返回true失败返回false

}

string getIdentity() //获取身份标签

{

//返回身份标签  
}

#### 1.1.2管理员类(继承基本用户类)

类方法

公共方法：

void menu()

{

//管理员菜单

//"================================================"

//"= 1.查看题库 =\n";

//"= 2.增加题目 =\n";

//"= 3.删减题目 =\n";

//"= 4.用户管理 =\n";

//"= 5.清空题库 =\n";

//"= 6.密码修改 =\n";

//"= 7.退出系统 =\n";

//"================================================\n";

//选择相关选项进行功能跳转

}

bool viewQuestions()

{

//查看题库，列出所以题库内容

//成功返回true失败返回false

}

bool addQuestions(题目类实例 problemData )

{

//参数 problemData 题目数据

//系统判断数据的完整性与正确性，若合法进行存储数据操作

//系统判断数据合法情况

//数据不合法，系统驳回，拒绝保存，系统反馈保存失败数据不合法

//数据合法，系统生成题目编号并写入题库文件写入

//写入成功，反馈保存成功

//写入失败，反馈保存失败及失败原因

//成功返回true失败返回false

}

int deleteQuestions(int num)

{

//参数 num需要删除的题目编号

//系统判断编号的真实性与合法性

//编号判断情况

//编号输入不合法，系统驳回,返回错误码-1：输入不合法

//编号输入合法但题库中不存在此编号的题目，删除题目失败，返回错误码-2，题目不存在

//编号合法并真实进入删除操作

//删除成功，系统反馈删除成功,返回0

//删除失败，系统反馈删除失败，返回错误码-3

}

void userManagement(string userName, CAdmin admin, string adminName)

{

//用户名 userName

//管理员类实例 admin

//管理员名 adminName

//用户管理菜单

//1.重置密码

//2.删除账户

//3.返回上级

//选择相关选项进行功能跳转

}

bool clearQuestions(int instruct, string path)

{

//题库文件路径 path

//参数 instruct 指令

//指令 0 取消清空，自动返回上一级

//指令 1确认清空，进行文件操作

//执行成功返回true失败返回false

}

bool resettingUserPassword(string Name, string newPassword, string path) //重置用户密码

{

//用户名 Name

//新密码 newPassword

//用户文件路径 path  
}

#### 1.1.3普通用户类(继承基本用户类)

类方法

公共方法：

void menu()

{

//用户菜单

// "================================================\n";

// "= 1.开始测试 =\n";

// "= 2.修改密码 =\n";

// "= 3.注销账号 =\n";

// "= 4.退出系统 =\n";

// "================================================\n";

//选择相关选项进行功能跳转

}

bool testQuestions(int num)

{

//参数num:抽题数量

//参数合法性判断

//超过题库最大题目数量，返回-1

//完全合法，进行测试

//题目测试

//读取用户输入

//合法判断

//非法输入，系统驳回，要求重新输入

//合法输入，进行评分

//未出错返回0未知错误返-2

}

#### 1.1.4题目类

类属性

私有属性：

编号 num 类型 string

题目 title 类型string

选项 A B C D, 类型string

公共属性：

答案 answer 类型 char

类方法

公共方法：

Bool setTitle(string \_Num)

{

//题目 \_Num

//写入题目编号

//成功返回true失败返回false

}

bool setTitle(string \_title)

{

//题目 \_title

//写入题目

//成功返回true失败返回false

}

bool setA(string \_A)

{

//答案\_A

//写入A答案

//成功返回true失败返回false

}

bool setB(string \_B)

{

//答案\_B

//写入B答案

//成功返回true失败返回false

}

bool setC(string \_C)

{

//答案\_C

//写入C答案

//成功返回true失败返回false

}

bool setD(string \_D)

{

//答案\_D

//写入D答案

//成功返回true失败返回false

}

bool setAnswer(string isAnswer)

{

//标准答案\_isAnswer

//写入标准答案

//成功返回true失败返回false

}

string getNum()

{

//返回题目编号

}

string getTitle()

{

//返回题目

}

string getA()

{

//获得A答案

//返回A答案

}

string getB()

{

//获得B答案

//返回B答案

}

string getC()

{

//获得C答案

//返回C答案

}

string getD()

{

//获得D答案

//返回D答案

}

string getAnswer()

{

//返回题目标准答案

}

1.2全局函数

int registerNum(string userName ) //注册

{

//参数userName 用户名

//系统进行用户名查重

//用户名已存在返回错误码-1

//写入用户文件

//写入失败系统反馈写入失败返回错误码-2

//写入成功返回0

}

int logOn(string userName, string password) //登录

{

//参数userName 用户名， password 密码

//系统进行用户名校验与密码核对

//用户名和密码都正确，进行身份判断

//管理员身份，返回1

//普通用户身份，返回2

//未知身份，返回0

}

void systemMenu() //主菜单

{

// "================================================\n";

// "= 1.注册账号 =\n";

// "= 2.登录系统 =\n";

// "= 3.退出系统 =\n";

// "================================================\n";

}

void unregisteredUserMenu() //未注册用户菜单

{

//"================================================\n";

// "= 1.重新输入 =\n";

// "= 2.注册账号 =\n";

// "= 3.退出系统 =\n";

// "================================================\n";

}

void exitSystem()

{

//程序正常退出

}

void systemSleep(int stime) //延迟等待

{

//延迟时间stime

}

异常处理函数

//异常退出系统

void exitErrorTheSystem()

{

//程序异常退出

}

//身份验证异常

void unknownIdentity()

{

// "未知身份，登录失败，请联系统管理员\n";

// "按Q退出程序，按C继续输入";

}

//刷新缓冲区

void refreshBuffer()

{

//将文件指针重新指向文件开头,达到刷新缓冲区效果

//rewind(stdin);

}

bool judgmentString(string& str) //判断是否为字符串

{

//str 字符串

//判断是否为英文字母字符串

//符合条件返回true

//不符合条件返回false

}

bool judgmentNumber(string& str) //判断是否为数字

{

//str 字符串

//判断是否为数字字母字符串

//符合条件返回true

//不符合条件返回false

}

bool openMenu(string& userName, string& userType) //开启菜单

{

//用户名 username

//用户类型 userType

//根据用户类型打开相应菜单

//运行正常返回true

//运行不正常返回false

}

bool returnMenu(string& userName, string& userType) //返回上一级

{

//用户名 username

//用户类型 userType

//返回上一个菜单项

//运行正常返回true

//运行不正常返回false

}

string getpassword() //加密输入

{

//接受键盘输入并加密输入内容

}

void Random(int\* arr, int num, int minNum, int maxNum)

{

//生成范围在minNum~maxNum的随机数

//存储随机数群的数字指针 arr

//随机数数量num

//题目最小编号minNum

//题目最大编号 maxNum

}

int cmpfunc(const void\* a, const void\* b) //题目编号排序方式

{

//编号a a

//编号b b

//返回排序规则

}

char getUserAnswer() //读取用户输入的答案

{

//获取用户输入的答案

//返回用户输入的答案

}

bool getProblemData(vector<CProblem>\* data, int\* maxNum)//获取题库数据

{

//题库数据指针 data

//最大题目编号指针 maxNum

//运行正常返回true

//运行不正常返回false

}

string strToLower(string& str) //字符串转换成小写

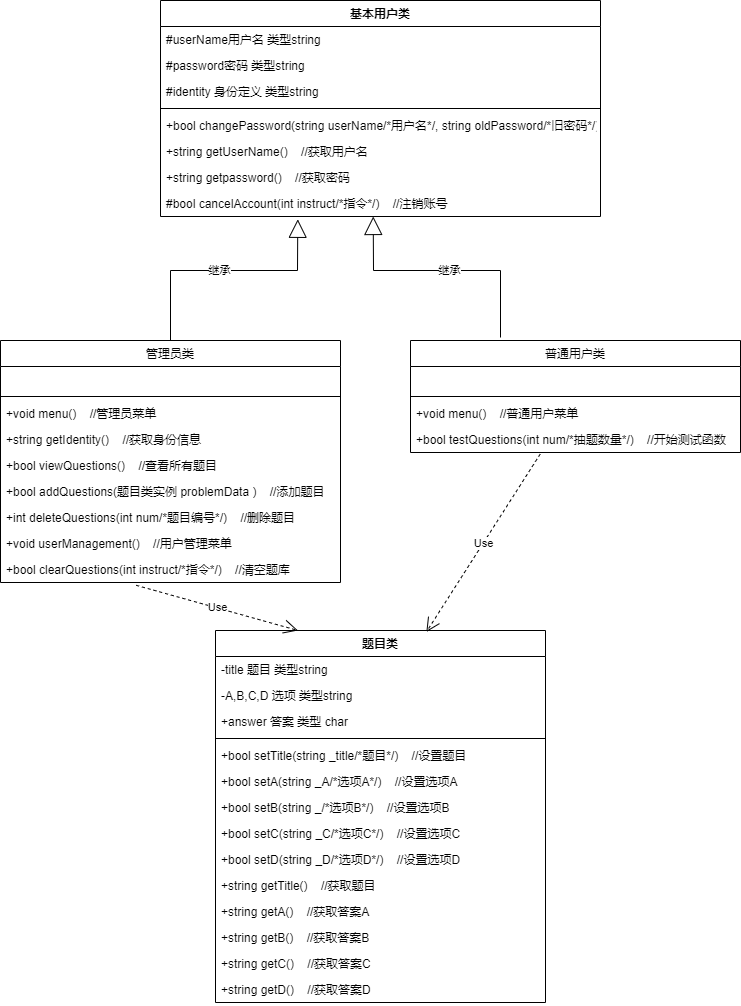
{

//字符串

//字符串转换为小写

//返回转换后的字符串

}

1. 类图