简易员工工号管理系统

# 考试注意事项

1. 请仔细阅读规格文档, 务必看清楚要求再开始答题。预计理解题目规格文档需花费半小时左右。
2. 注意：编译不通过得零分，建议定期编译运行所写代码。
3. 每道题已经提供初步的框架，及基本功能的测试用例，请在此框架上继续完成代码。
4. 试题框架提供的源代码，考生可自行补充成员方法、成员变量或新的类，但不能改变已有函数的签名信息（即输入参数、函数名、返回值、修饰符等）。
5. 随试题提供的代码框架使用UTF-8编码，请调整Idea的编码字符集为UTF-8，以防止源文件中的中文出现乱码。
6. 只允许使用JDK自带的API，不得使用第三方库。
7. 除题目明确要求需要考虑的异常情况、输入校验外，其他异常情况和输入校验一律不需要考虑。
8. 随试题提供的测试用例仅用于基本功能的验证，非评分依据，即使全部通过也不意味着考试通过。
9. 最终答题代码，务必遵从系统的要求提交，要求如下:
10. 请将工程目录下的src文件夹压缩后上传；
11. 测试截图

压缩文件夹命名规则：小组\_校\_姓名 二组\_南邮\_张三

# 任务说明

本任务是开发一个“简易员工工号管理系统”，对所有员工账号进行添加、查询、申请变更等操作，每一名员工都有唯一的一个工号。

## 1.1 系统规则

1、初始化系统

系统初始化后清空所有的员工数据。

2、创建员工档案

根据员工的名称和员工类型，创建员工并为员工分配工号。

3、更新员工工号

为现有的一个员工申请和分配一个新的工号，使用新工号替换该员工的老工号。

4、更新员工状态

当员工离职时，更新该员工的状态，以表示其离职。

5、查询员工信息

根据员工的名称，查询该员工的信息。

## 1.2 系统约束

1. 考生需要实现系统初始化、创建员工、更新员工工号、更新员工状态和查询员工工号等功能。
2. 每个员工信息都包含以下字段，各字段的约束条件请见下面表格：

|  |  |  |  |
| --- | --- | --- | --- |
| **字段** | **对应变量名** | **取值** | **举例说明** |
| 员工姓名 | employeeName | 1. 姓名拼音 2. 取值为长度不超过18个字符的小写字母，不包含空格、@、#等特殊字符; | 例如，某员工姓名叫张三，则员工姓名的测试用例输入值为：zhangsan = zhangsan |
| 员工类型 | employeeType | 正式员工取值为：true  合作员工取值为：false |  |
| 在职状态 | Status | 在职员工取值为：true  离职员工取值为：false |  |
| 员工工号 | employeeID | 1. 员工有两类，一类是正式员工，一类是合作员工；所有员工完整工号包含两部分：字母标志部分+数字标志部分； 2. 字母标志部分取值为员工姓氏的首字母（大写），其中合作员工工号前两位字母固定为WX（大写）； 3. 数字标志部分为4位整数，取值范围为1000~9999（含1000和9999，测试用例输入保证不会越界，考试不必关注）；所有员工的工号中数字标志部分必须唯一； 4. 数字标志部分从1000开始递增使用，即公司第一名入职员工工号的数字标志部分为1000， 第二名入职员工工号的数字标志部分为1001，以此类推； 5. 员工离职后，其对应的数字标志部分的数值被释放，可以重新投入使用，重新分配工号时，从最小的数值进行分配； | 正式员工工号举例：zhangsan，其工号为：Z1010；  合作员工工号举例：chenyu，其工号应为：WXC1011；  第一位入职的正式员工wangsan，其工号为： W1000；  第二位入职的合作员工zhangwu，其工号为：WXZ1001；  如果wangsan离职，则第三位入职的正式员工kaixin的工号应为：K1000。 |

# 操作说明

所有功能测试使用单元测试

## 系统初始化 (10分)

**方法定义：**

**public** **void** init();

**功能说明**：

系统初始化后清空所有数据

**参数说明：**

无

**输出说明：**

无

## 创建员工档案 (20分)

**方法定义：**

/\*\*

\* 请实现创建新员工档案功能

\* @param employeeName 员工姓名(输入侧保证合法性)。

\* @param employeeType 员工类型，

\* @return 返回该员工的完整工号字串

\*/

**public String** createEmployee (String employeeName, boolean employeeType)

**功能说明**：

实现创建员工功能，返回创建员工的完整工号，注意：创建员工后，员工为在职状态。

**参数说明：**

员工姓名(employeeName)，员工类型(employeeType) （可使用随考题提供的测试用例保证合法性）。

**输出说明：**

员工工号（employeeID）（考生实现自动为入职员工生成工号功能）。

## 更新员工工号 (30分)

**方法定义：**

/\*\*

\* 请实现变更工号的功能

\* @param oldEmployeeID 待变更员工ID

\* @param idNumber 申请的新工号数字部分，为4位数数字，输入侧保证在1000-9999之间，程序中建议对合法性进行检测

\* @return 准许变更，返回true， 否则，返回false)

\*/

**public boolean** updateEmployeeID (String oldEmployeeID, int idNumber)

**功能说明**：

对系统中存在并且是在职状态的员工可以变更工号。

* 1. 系统不存在的、或已离职的、或合作员工，不得进行变更申请，返回false，即需要对待变更工号进行合法性检测；
  2. 只准许在职且正式员工申请变更工号；
  3. 员工只能申请变更工号的数字标志部分，变更成功后旧工号数值被释放到资源池，可供其它员工使用；
  4. 数字标志部分取值有如下来源：其它员工离职之后被释放的数值，或者其他员工变更申请后被释放的数值，故需要对申请的新工号数字部分进行合法性检测；
  5. 满足变更条件，则准许变更，更新员工新工号信息，返回true；否则返回false

**参数说明：**

待变更员工工号，申请变更的新工号数字部分

**输出说明：**

true or false

## 更新员工状态 (30分)

**方法定义：**

/\*\*

\* 请实现根据工号更新员工在职/离职状态信息功能。

\* @param employeeID 员工完整工号（例如，z1002）

\* @param newStatus: 将在职/离职状态修改为 newStatus

\* @return true or false 参见1.2章节系统约束

\*/

**public boolean**  updateStatus(String employeeID boolean newStatus)

**功能说明**：

根据工号更新员工在职/离职状态信息功能。

1）如该工号对应员工不存在，或已离职，禁止更新，返回false

2）该工号对应员工存在且在职，可以更新，返回true，如一员工由在职变为离职，其对应的工号数字部分释放，可以在创建员工档案或更新员工工号时使用。

**参数说明：**

员工完整工号，员工的新状态

**输出说明：**

true or false

## 查询员工信息 (10分)

**方法定义：**

/\*\*

\* 请实现根据员工姓名查询员工完整工号功能。

\* @param employeeName 员工姓名全拼写

\* @return 参见1.2章节系统约束

\*/

**public String** getEmployeeIDByName(String employeeName)

**功能说明**：

根据员工姓名，查找并返回员工完整工号。

* + 1. 如根据姓名，查找不到任何员工信息，返回“unknown”；
    2. 如根据姓名，查找到1名或2名以上在职员工，则返回数字标志部分最小的员工对应的完整员工工号；
    3. 如根据姓名，可以查找到员工，但是全部处于离职状态（1名或多名）， 返回”quit”；
    4. 如同时找到在职和离职员工，返回在职员工对应的完整员工工号；

**参数说明：**

员工姓名

**输出说明：**

员工的工号信息如：X2341