小学期项目文档

## 开发环境

推荐DevC++/小熊猫C++ MinGW-w64 GCC 13.1 Debug。

## 数据模型

数据表：**顾客个人信息表、管理员个人信息表、场地信息表、预定信息表。**

枚举：运动类型，地理区域。

命名规范：**场馆Venue，场地Site**

## 业务拆分

**数据持久化方案未定，以下出现的数组也可能是链表节点！！！**

将用户交互逻辑与业务逻辑分离。交互编写咱们放到后面再说。

整体：顾客查询一人，顾客预定一人，管理员查询/统计一人，用户或管理员登录/注册/个人信息修改/订单取消一人，数据存取中台一人。

关于返回数组见：https://www.runoob.com/cprogramming/c-return-arrays-from-function.html

### 总入口 - 登录注册

建议拆分管理员登录入口与普通用户登录入口。

用户：

* 注册模块：注册为新用户。用户输入用户ID、姓名、性别、年龄、电话、密码、邮箱、地理区域，业务逻辑中堆用户输入的基本信息进行校验（如用户ID不合规定，姓名超长，年龄大于130岁，电话不是11位数，邮箱不符合规范等），校验通过则保存信息，不通过则要求用户修改。

1. 将各部分校验封装成函数，参数为对应条目的字符串，返回值为该条目是否合规。推荐函数命名：isValidName，isValidUserID，isValidGender等等。

2. 将校验完成后的注册过程封装成函数，默认所有传入参数均合法。参数为以上各项用户输入（字符串），返回值为是否成功注册。推荐函数命名：doRegister。

* 登录模块：用户进行登录。用户输入用户ID与密码登录系统，需先对用户输入信息作基本校验（如用户ID和密码不得为空，用户ID需要符合格式等），若校验通过则取用现有用户数据比对是否存在此用户，用户不存在时提醒用户不存在，存在则比对密码，密码错误则提醒密码错误，若ID及密码均正确则登录成功。

1. 校验部分可复用注册部分的校验函数。

2. 将校验完成后的登录验证过程封装成函数。参数为用户ID和密码（字符串），返回值定义：若存在此用户且密码正确返回0，若存在用户但密码不正确返回1，若用户不存在返回2。推荐命名：doUserLogin

管理员：

* 登录模块：管理员进行登录。用户输入用户ID与密码登录系统，需先对用户输入信息作基本校验（如用户ID和密码不得为空，用户ID需要符合格式等），若校验通过则取用现有用户数据比对是否存在此用户，用户不存在时提醒用户不存在，存在则比对密码，密码错误则提醒密码错误，若ID及密码均正确则登录成功。

1. 将校验完成后的管理员登录验证过程封装成函数。参数为用户ID和密码（字符串），返回值定义：若存在此管理员用户且密码正确返回0，若存在用户但密码不正确返回1，若用户不存在返回2。推荐命名：doAdminLogin

### 用户方面

#### 场地查询

查询模块：

* 用户输入场地名，返回该场地的场馆、预定等信息
* 用户输入场馆名，返回该场馆下的所有场地
* 用户输入运动类型，返回符合该运动类型的所有场地
* 用户选择一个区域，返回该区域下的所有场地
* 在以上查询完成后，允许用户根据有无空余场地进行过滤，根据租金和预定量排序
* 可对用户查询到的场地直接预定

1. 将查询场地信息封装成函数，参数为场地ID，返回该场地所属的场馆和预定等信息，可用两个函数实现，也可令函数返回数组取用。参考函数命名：querySiteInfo，或querySiteVenue、querySiteAppoint等。

2. 将根据场馆查询场地、根据运动类型、根据区域查询场地均封装成函数，参数为场馆，返回该场馆下的所有场地（数组）。参考函数命名：queryVenueSite，querySportSite等等。

3. 将过滤步骤封装成函数。参数为结构体数组或场地名称数组，返回过滤后的结构体数组或场地名称数组，可按实际开发难度自由选择。参考函数命名filterHasVacant

4. 将排序封装成函数。参数为结构体数组或场地名称数组，返回过滤后的结构体数组或场地名称数组，可按实际开发难度自由选择。参考函数命名sortByRentMoney，sortByAppointment等。

5. 热门场地预定，待所有模块开发完毕后另行开发。

#### 场地预定

预定模块：

* 每次预定可预定多个场地；
* 预定时间以整点开始，整点结束；
* 来自不同场馆的场地自动拆分成多个预定订单；
* 租借的场地数目不能超出场馆的可用场地数目；
* 租借的场地总费用不能超出账户余额；
* 要能处理时段冲突；
* 有些运动不适合儿童或者老年人，不允许预约；
* 历史上爽约超过三次的顾客不允许预定。

1. 封装一系列函数判断用户输入是否合法。传入用户输入数据（场地ID、），返回是否合法。参考函数命名isValidSiteID、isValidStartTime、isValidEndTime等等。

2. 封装函数判断用户是否可以预约某场地。传入用户ID、场地ID、预约开始时间（int）、预约结束时间（int），若可预约返回0，若用户爽约次数超过三次返回1，用户年龄与场地不匹配返回2，用户预约时段与自身预约有冲突返回3，用户预约时段与他人有冲突返回4，用户余额不足返回5。

3. 封装函数写入预约信息，默认传入参数合法。传入用户ID、场地ID、预约开始时间、预约结束时间；需要操作场地信息、订单信息、预约信息三个表。返回是否写入成功。

#### 个人信息

* 用户可以修改自己的电话、密码、邮箱、地址等个人信息。

1. 使用上面写好的函数校验用户输入合不合法。

2. 封装函数修改用户信息，参数为用户电话、密码、邮箱、地址，返回是否修改成功。

* 用户可在预约生效24h前取消预定信息。

1. 编写函数返回当前日期，精确到小时。

2. 判断用户是否可以取消预约。传入预定条目ID、用户ID，若可取消此预约返回0，若时间超出限制返回1，其他情况返回2。

* 用户可查询自己的订单。

1. 封装函数查询用户订单，默认参数信息合法。传入用户ID，返回用户所有订单记录数组。

### 管理员方面

* 管理员需要查看场馆预定情况

1. 复用原有函数，根据场馆名查询其下所有场地

2. 封装函数查询场地预定情况，传入管理员用户ID，返回预定ID数组。

* 管理员需要查看统计信息

1. 按预定量排序场地的功能，复用用户系统中编写的函数。

2. 封装函数统计场地热门预定时间。传入场地ID，返回场地各时间段按预定数的排序结果。

3. 封装函数计算**场地**营业额。传入场地ID，返回该场地的总营业额。

4. 用户年龄层次分布、男女热门运动等按上述方法设计函数实现。

5. 封装函数，按营业额排序场馆内所有场地，传入管理员ID，返回排序后的场地列表。

6. 按时间段排序场馆内场地。传入时间段（int，按一小时计算，如传入9则查询9~10点的预约记录），返回排序后的场馆数组。

* 管理员可查询、添加场地。

1. 查询场地。上面已经实现过这个函数了。

2. 添加场地。传入场地ID名称等等信息，返回是否添加成功。

* 管理员可编辑自身信息

1. 与用户的机制一致，但注意编辑的数据表不同。