**系统总体分析与设计**

1. **系统总体功能描述**

**模块图：**

1. 用户管理模块

功能：处理用户相关的所有操作，包括注册、登录、个人信息修改等。

子模块：用户注册

用户登录

用户个人信息修改

用户账户余额管理

2. 管理员管理模块

功能：处理管理员相关的所有操作，包括注册、登录、个人信息修改等。

子模块：管理员注册

管理员登录

管理员个人信息修改

管理员操作界面（查看影厅、添加排片、查看排片、订单管理等）

3. 电影管理模块

功能：处理电影和场次的所有操作，包括电影的添加、删除、修改以及排片管理等。

子模块：电影信息录入

排片管理（添加、删除、修改场次）

电影和场次信息查询

4. 订单管理模块

功能：处理用户订单的所有操作，包括订单的创建、查询、修改和删除等。

子模块：订单创建

订单查询

订单状态更新（如取消订单）

订单数据持久化（保存到文件）

5. 影厅管理模块

功能：处理影厅的所有操作，包括影厅信息的录入、查询和修改等。

子模块：影厅信息录入

影厅信息查询

影厅座位管理（座位分配、释放等）

6. 公共工具模块

功能：提供程序中可能会重复使用的工具函数或类，如时间转换、数据验证等。

子模块：时间转换函数

数据验证函数（如邮箱、ID、电话验证）

文件读写函数

总体功能描述

**总体功能描述：**

实现一个综合性的电影票购买与管理系统。系统包含用户管理、管理员管理、电影管理、订单管理和影厅管理等多个功能模块，旨在为用户提供便捷的电影票购买体验，同时为管理员提供全面的后台管理功能。

具体来说，该程序的总体功能包括但不限于以下几点：

1. 用户管理：

用户注册：允许新用户创建账户，并输入必要的个人信息（如用户名、密码、邮箱、电话号码等）。

用户登录：验证用户的登录凭据，允许已注册用户登录系统。

个人信息修改：允许用户更新其个人信息，如密码、邮箱、电话号码等。

账户余额管理：记录用户的账户余额，支持充值和支付操作。

1. 管理员管理：

管理员注册：允许新的管理员创建账户，并输入必要的个人信息和管理员权限密码。

管理员登录：验证管理员的登录凭据，允许已注册管理员登录后台管理系统。

个人信息修改：允许管理员更新其个人信息。

后台操作界面：提供管理员进行电影管理、影厅管理、订单管理等操作的界面。

1. 电影管理：

电影信息录入：允许管理员添加新的电影信息，包括电影名称、类型、时长、票价等。

排片管理：允许管理员为特定影厅和时间段安排电影放映场次，并设置座位数、放映时间等。

电影信息查询：提供电影信息的查询功能，支持按电影名称、类型等条件进行筛选。

1. 订单管理：

订单创建：允许用户选择电影场次、座位并创建订单，同时扣除相应的账户余额。

订单查询：提供用户查询自己订单历史的功能。

订单状态更新：支持订单状态的修改，如取消订单、完成订单等。

订单数据持久化：将订单数据保存到文件中，确保数据的持久性和可恢复性。

1. 影厅管理：

影厅信息录入：允许管理员添加新的影厅信息，包括影厅名称、座位数等。

影厅信息查询：提供影厅信息的查询功能，支持按影厅名称等条件进行筛选。

座位管理：在放映场次中管理影厅的座位分配情况，包括座位的占用和释放。

1. **接口定义规范**

该系统使用了单一职责原则，接口保持单一职责，即一个接口应该只负责一项职责。这有助于降低接口的复杂性，提高系统的可维护性和可扩展性。在该程序中，不同的功能模块（如用户管理、电影管理、订单管理等）各自定义了对应的接口，每个接口负责该模块的具体业务逻辑。

1. **工程组架构设计**

该系统使用了分模块设计，模块化设计是现代软件开发中常用的一种架构方法，它将一个大系统分解为若干个相互独立但又相互联系的模块，每个模块完成一个特定的功能。在该程序中，可以明显看到多个功能模块的存在，如用户管理模块、管理员管理模块、电影管理模块、订单管理模块和影厅管理模块等。这些模块各自独立，通过接口进行交互，从而实现了系统的整体功能。

1. **系统的数据结构描述**

该程序作为一个综合性的电影票购买与管理系统，需要处理多种类型的数据。这些数据在系统中以不同的形式存储，并在程序运行过程中使用各种数据结构来管理和操作这些数据。以下是对该程序需要处理的数据、数据存储形式及使用的数据结构的详细描述：

**需要处理的数据类型：**

用户数据：包括用户的基本信息（如用户名、密码、邮箱、电话号码、性别等）、账户余额、订单历史等。

管理员数据：包括管理员的基本信息（如管理员ID、密码、邮箱、电话号码、姓名等）以及管理员的操作记录等。

电影数据：包括电影的基本信息（如电影名称、类型、时长、票价等）、放映场次信息（如放映时间、影厅、座位数等）。

订单数据：包括订单的详细信息（如订单号、用户ID、电影场次、座位号、订单金额、订单状态等）。

影厅数据：包括影厅的基本信息（如影厅名称、座位数等）以及座位分配情况。

**数据存储形式：**

数据库：系统中最常用的数据存储形式是数据库。数据库可以高效地存储和管理结构化数据，如用户信息、电影信息、订单信息等。

内存：在程序运行过程中，系统还会将部分数据加载到内存中以提高数据访问速度。内存中的数据通常是临时的，程序结束后数据会丢失。

**程序运行过程中使用的数据结构描述：**

数组和列表：用于存储具有相同类型元素的集合。例如，用户列表、电影列表、订单列表等都可以使用数组或列表来表示。

结构体和类：在面向对象编程中，结构体和类是非常重要的数据结构。它们允许将多个相关的数据项组合成一个单一的实体（对象），并提供了操作这些数据项的方法。在该程序中，用户、管理员、电影、订单等都可以定义为结构体或类来表示其属性和行为。

1. **系统的开发环境及运行环境描述**

**开发环境：**

1. 硬件环境：

笔记本电脑

1. 软件环境：

操作系统：Windows

集成开发环境（IDE）：Visual Studio Codes

编程语言：C语言

**运行环境：**

Windows系统