# 零度网盘概要设计说明书

1. 任务

1.1 目标

设计一个高效、安全、易用的云存储服务，即零度网盘，以满足用户文件存储和共享的需求。

1.2 环境

操作系统： Windows10/11

开发语言：C++

数据库：MySQL

1.3 需求

用户可以注册、登录、上传、下载和分享文件。

实现文件夹管理和文件搜索功能。

提供安全的数据传输和存储机制。

实现用户权限管理，包括公开分享和私密文件。

1.4 局限

不包括个性化推荐功能。

可能存在上传下载速度较慢的现象。

2. 总体设计

2.1 处理流程

用户注册和登录

文件上传和下载

文件管理（创建、删除、移动文件和文件夹）

文件分享和权限管理

2.2 总体结构与模块

用户管理模块

文件管理模块

权限管理模块

数据存储模块

网络通信模块

2.3 功能与模块的关系

用户管理模块负责处理用户注册和登录。

文件管理模块实现文件的上传、下载和管理。

权限管理模块用于处理文件的分享和权限控制。

数据存储模块负责安全存储用户文件和相关信息。

网络通信模块处理与客户端和其他服务器的通信。

3. 接口设计

3.1 外部用户接口

GUI界面：用户通过视窗访问零度网盘。

3.2 软硬件接口

数据库接口：与MySQL数据库交互。

网络接口：处理与客户端的通信。

3.3 内部模块间接口

用户管理模块与文件管理模块通过用户身份信息进行关联。

文件管理模块与权限管理模块协同实现文件分享和权限控制。

4. 数据结构

4.1 逻辑结构

用户信息表

文件信息表

文件夹信息表

权限信息表

4.2 物理结构

用户文件存储目录结构

数据库存储结构

4.3 程序结构的关系

程序通过数据库查询和更新操作实现对逻辑结构的管理。

5. 模块设计

5.1 用户管理模块

功能： 处理用户注册和登录。

输入： 用户注册信息、登录凭证。

输出： 用户登录状态、用户信息。

5.2 文件管理模块

功能： 处理文件上传、下载和管理。

输入： 文件、文件夹操作请求。

输出： 文件列表、上传下载状态。

5.3 权限管理模块

功能： 处理文件分享和权限控制。

输入： 分享请求、权限设置。

输出： 分享链接、权限状态。

6. 运行设计

6.1 运行模块组合

用户管理模块和文件管理模块同时运行，协同处理用户的文件操作请求。

6.2 控制

控制通过调度器实现，确保模块按照正确的顺序和时间运行。

6.3 时间

实时处理用户请求，保障系统响应时间。

7. 出错设计

7.1 出错信息

提供详细的错误信息，包括用户操作失败的原因和解决建议。

7.2 处错处理

实现日志记录功能，方便开发人员追踪和解决问题。

8. 其他设计

8.1维护

定期数据库备份和系统更新，确保系统的稳定性和安全性。

# 结构化软件详细设计说明书结构

1引言

1.1编写目的

　　说明编写这份详细设计说明书的目的，指出预期的读者。

1.2背景

　　说明：

　　a．待开发的软件系统的名称：零度网盘；

b．本项目的任务提出者：庞帅

开发者：庞帅、程志琦、陈高上、覃灏

用户：主要面向学生、教师等个人用户

实现该软件的计算中心或计算机网络；校园网。

1.3定义

文件上传： 将本地文件传输到零度网盘服务器的过程，用户可通过网盘管理这些文件。

文件下载： 从零度网盘服务器获取文件到本地设备的过程，用户可在本地使用这些文件。

文件分享： 允许用户将文件分享给其他用户或生成外链，使其他人可以查看或下载文件。

权限管理： 控制文件或文件夹访问权限的功能，包括读取、写入、删除等操作的授权。

版本控制： 保留文件的不同版本，使用户可以恢复到先前的文件状态。

用户界面： 用户与零度网盘交互的图形化界面，包括网站、移动应用等。

数据传输协议： 定义数据传输格式和规则的协议，例如HTTP、HTTPS等。

安全保密： 对用户隐私和数据安全的保护措施，包括加密、身份验证等。

1.4参考资料

　　a．《软件工程原理》课程需求说明书模板

b.《概要设计与详细设计模板》

2程序系统的结构

一、用户基本功能实现

1.1 用户注册功能实现

1.2 用户登录功能实现

1.3 用户退出功能实现

二、操作主界面UI设计

2.1 好友子界面UI设计

2.2 图书子界面UI设计

2.3 合并子界面UI到操作主界面UI

三、用户登陆跳转功能实现

四、好友操作功能实现

4.1 查看在线好友

4.2 搜索好友

4.3 添加好友

4.4 刷新在线好友

4.5 删除好友

4.6 私聊

4.7 群聊

五、文件操作功能实现

5.1 图书界面设计

5.2 文件夹操作

5.2.1 创建文件夹

5.2.2 查看所有文件

5.2.3 删除文件夹

5.2.4 重命名文件

5.2.5 进入文件夹

5.2.6 返回上一级

5.3 常规文件操作

5.3.1 上传文件

5.3.2 删除文件

5.3.3 下载文件

5.3.4 分享文件

5.3.5 移动文件

3 用户注册功能

3.1 程序描述：

用户注册功能程序的设计旨在实现用户在系统中进行注册的功能。其主要目的是为了管理系统用户，并确保系统的安全性和可追溯性。该程序的特点包括：

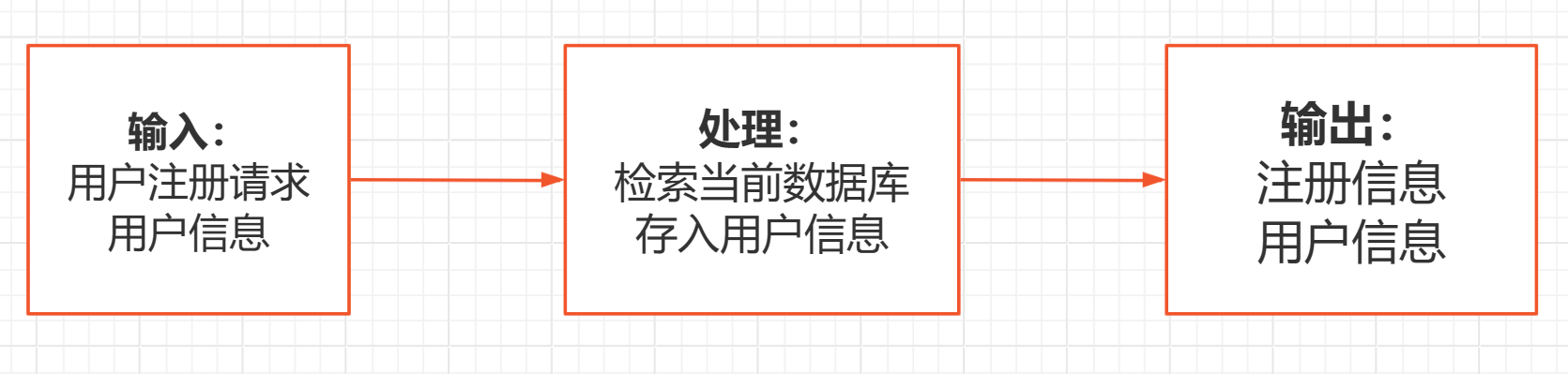
非常驻内存：用户注册功能在需要时调用，不常驻内存。

子程序：作为系统的一个子程序，用于处理特定的用户注册请求。

可重入性：设计考虑支持多用户同时注册，具有可重入性。

顺序处理：用户注册按顺序进行，确保每个注册请求都得到妥善处理。

3.2 功能：

3.3 输入项：

用户名（Name）：字符串，格式为文本，有效范围取决于系统要求，通过表单输入。

密码（Password）：字符串，格式为文本，有效范围取决于系统要求，通过表单输入。

3.4 输出项：

注册状态（Status）：字符串，表示注册成功或失败，格式为文本。

用户信息（UserInfo）：包括用户名等信息。

3.5 算法：

用户注册功能的算法可简述如下：

接收用户注册请求。

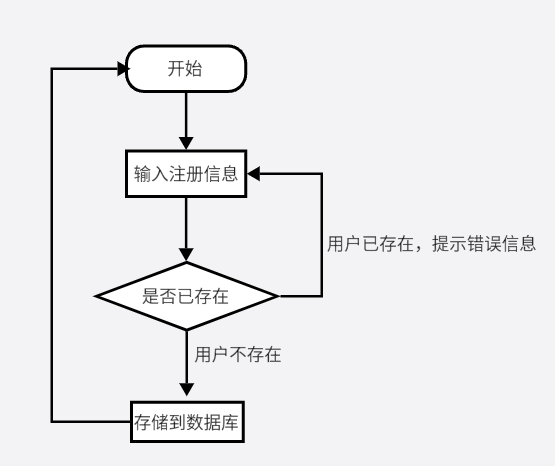
验证输入数据的有效性和合法性。

生成唯一的用户ID。

将用户信息存储到系统数据库中。

返回注册状态和用户信息。

3.6 流程逻辑：



3.7 限制条件：

系统数据库存储容量。

用户名的唯一性限制。

3.8 尚未解决的问题：

如何处理密码的安全存储和传输。

是否需要添加验证码等额外的安全措施。

4 用户登录功能

4.1 程序描述

本程序旨在实现用户登录功能，其设计目的是为了确保系统只有经过身份验证的用户才能访问特定资源或执行特定操作。该程序的特点包括：

是否常驻内存： 非常驻内存，仅在用户登录时调用。

是否子程序： 是子程序，作为系统的一个模块被调用。

是否可重入： 不可重入，每次只能处理一个用户登录请求。

覆盖要求： 无覆盖要求，专注于用户登录功能。

处理方式： 顺序处理，依次执行用户身份验证步骤。

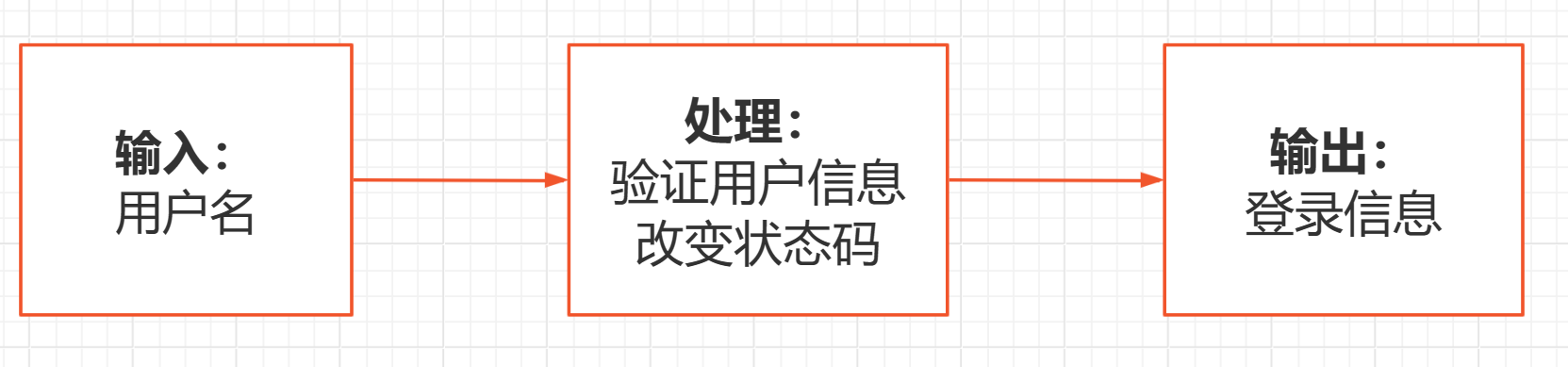
4.2 功能

用户登录功能程序应包含以下功能：

输入： 用户名和密码。

处理： 对输入的用户名和密码进行验证。

输出： 验证成功则允许用户访问系统资源，否则提示登录失败。

4.3 输入项

用户名： 字符串，格式为文本

密码： 字符串，格式为文本

输入方式： 通过用户界面输入。

4.4 输出项

访问状态： 字符串，格式为文本，表示用户登录成功或失败。

输出形式： 在用户界面上显示消息。

4.5 算法

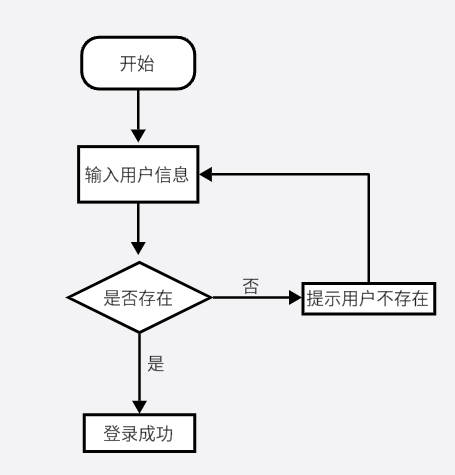
用户登录功能的简化算法如下：

获取用户输入的用户名和密码。

查询存储的用户信息，验证输入的用户名和密码是否匹配。

如果匹配成功，则允许用户访问系统资源；否则，提示登录失败。

4.6 流程逻辑



4.7 限制条件

用户登录程序受数据库查询性能的限制。

网络连接稳定性可能影响用户登录过程。

用户名和密码的有效性检查受输入限制。

4.8 尚未解决的问题

如何处理用户密码的安全存储和传输。

是否需要实现多因素身份验证。

如何处理用户在登录过程中的错误尝试次数限制

5 用户退出功能

5.1 程序描述

该程序旨在设计一个用户退出功能，以提供系统用户有序、安全地退出系统。主要目的在于管理用户会话和释放相关资源，确保系统的安全性和资源有效利用。

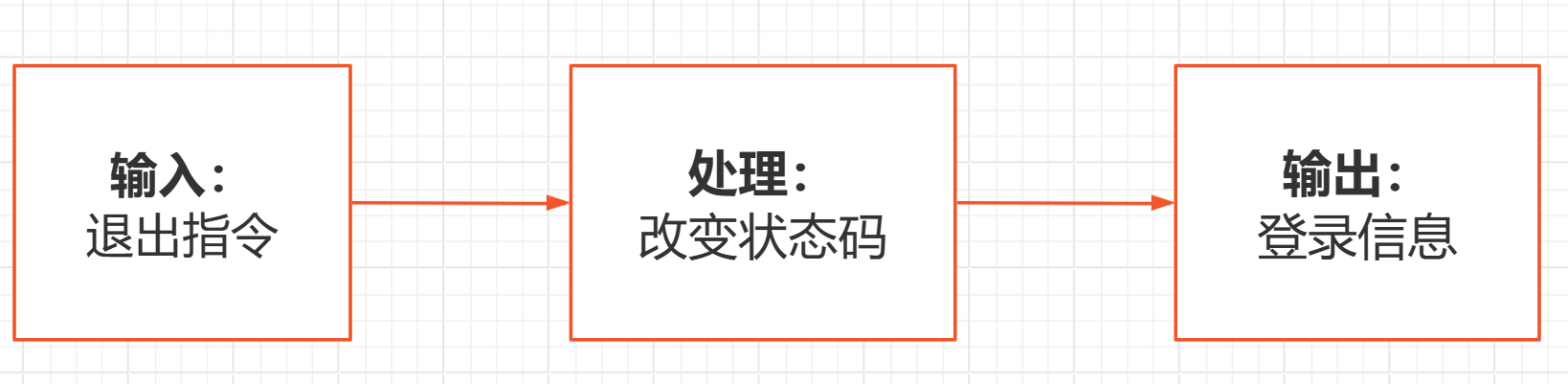
非常驻内存： 该程序不是常驻内存，而是在需要退出时被调用执行。

子程序： 是一个独立的子程序，可通过其他程序或系统界面调用。

不可重入： 为确保退出过程的一致性，该程序不支持被中断和重入。

顺序处理： 退出操作按照一定顺序执行，以确保数据完整性。

5.2 功能

5.3 输入项

退出指令

5.4 输出项

退出状态码

5.5 算法

接收用户信息作为输入。

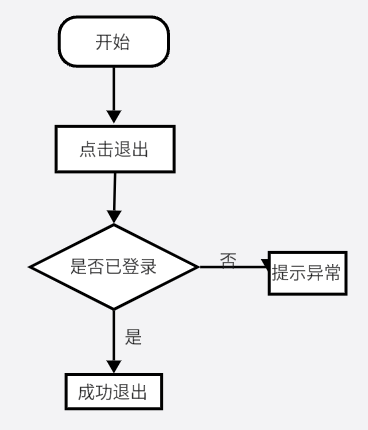
验证用户信息的有效性。

根据系统规定执行退出处理步骤。

生成退出处理信息。

输出退出处理信息。

5.6 流程逻辑



5.7 限制条件

用户信息必须是有效的，否则退出处理将失败。

退出过程中可能会有一定的时间延迟，具体取决于系统资源释放和其他清理操作的复杂性。

5.8 尚未解决的问题

未解决用户信息被恶意篡改的安全性问题，需要进一步加强数据传输和验证机制。

未解决在退出处理过程中可能发生的异常情况的处理方式，需要考虑异常处理机制

6 创建文件夹

6.1 程序描述

该程序的设计目的是创建文件夹，为用户提供一个方便快捷的工具，以满足文件组织和管理的需求。程序的主要特点包括非常驻内存、顺序处理、可重入、无覆盖要求。

非常驻内存： 程序在执行完成后不会一直驻留在内存中，只有在需要创建文件夹时才会加载。

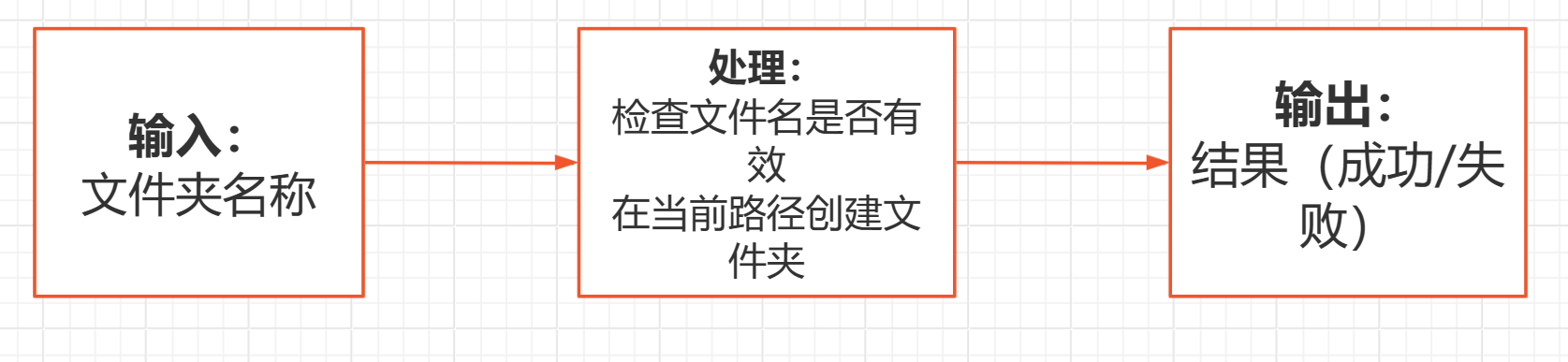
顺序处理： 操作是按照用户输入的顺序逐步执行。

可重入： 程序能够安全地同时被多个任务或用户调用，不会产生冲突。

无覆盖要求：不会覆盖已存在的同名文件夹。

6.2 功能

通过IPO图描述程序功能：



6.3 输入项

文件夹名称：

名称： 文件夹名称

数据类型： 字符串

6.4 输出项

操作结果：

名称： 结果

数据类型： 布尔值 (成功/失败)

6.5 算法

程序的算法采用以下步骤：

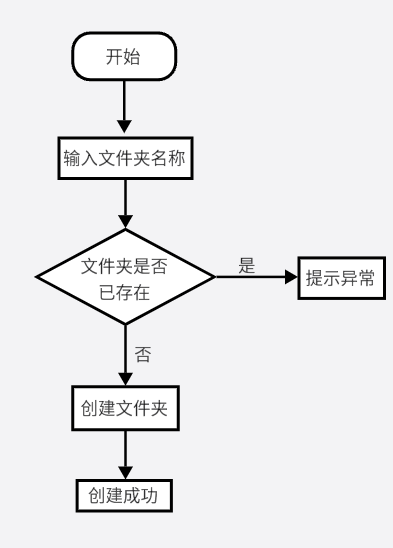
获取用户输入的文件夹名称。

检查是否已存在同名文件夹，若存在则返回创建失败。

创建新文件夹。

返回创建成功。

6.6 流程逻辑



6.7 限制条件

用户需要具有创建文件夹的权限。

文件夹名称不能包含非法字符。

文件系统空间不足可能导致创建失败。

6.8 尚未解决的问题

尚未解决如何处理特殊字符（如斜杠、冒号等）。

尚未解决如何处理文件夹创建失败的情况，是否提供详细的错误信息。

7 查看所有文件

7.1 程序描述

本程序旨在设计一个能够查看当前文件夹中所有文件的应用程序。其目的在于提供用户一种方便快捷的方式，通过执行程序来获取当前工作目录下的文件列表。该程序非常驻内存，是常驻的子程序。程序设计的特点包括：非常驻内存，顺序处理，可重入，无覆盖要求。

7.2 功能

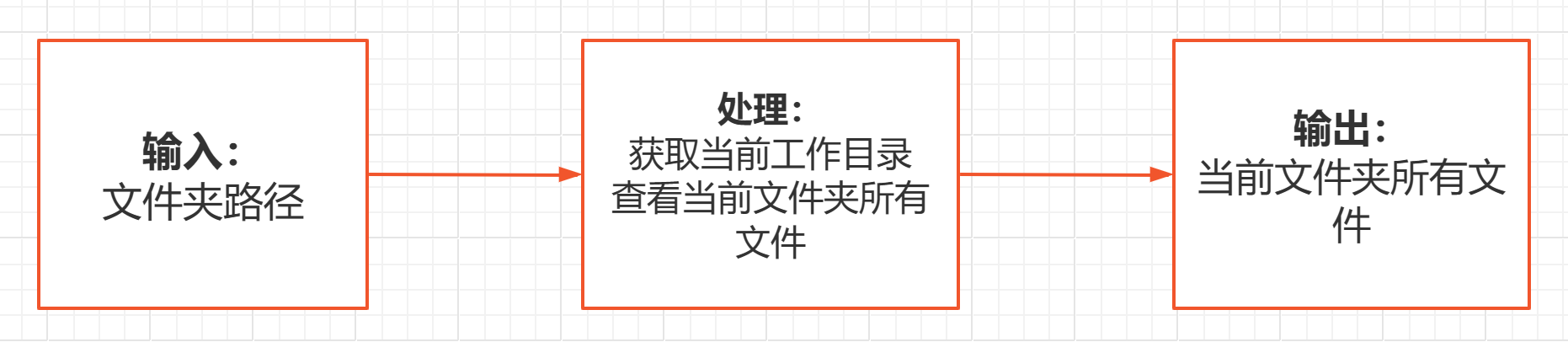
该程序的功能包括：

输入：无需用户输入，自动获取当前工作目录。

处理：遍历当前目录下的文件列表。

输出：显示文件列表。

IPO图如下：

7.3 输入项

无用户输入，自动获取当前工作目录。

7.4 输出项

文件列表的名称、数据类型为字符串，格式为文本形式。

输出形式为在命令行或图形界面中显示文件列表。

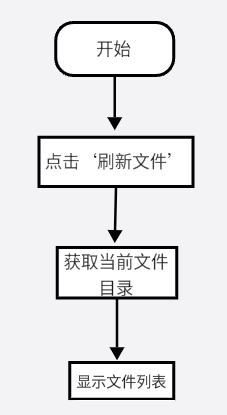
7.5 算法

获取当前工作目录。

遍历当前目录下的文件列表。

将文件列表输出到屏幕。

7.6 流程逻辑



7.7 限制条件

本程序受操作系统权限的限制，可能无法访问某些受限制的目录。

7.8 尚未解决的问题

是否考虑文件的属性信息。

对大量文件的性能优化。

8 删除文件夹

8.1程序描述：

本程序的设计目的是提供一个用于删除文件夹的实用工具。其主要目标是通过一个简单而有效的方式，允许用户删除指定的文件夹及其内容。该程序的特点包括非常驻内存、是子程序、可重入、无覆盖要求，采用顺序处理。

8.2功能：

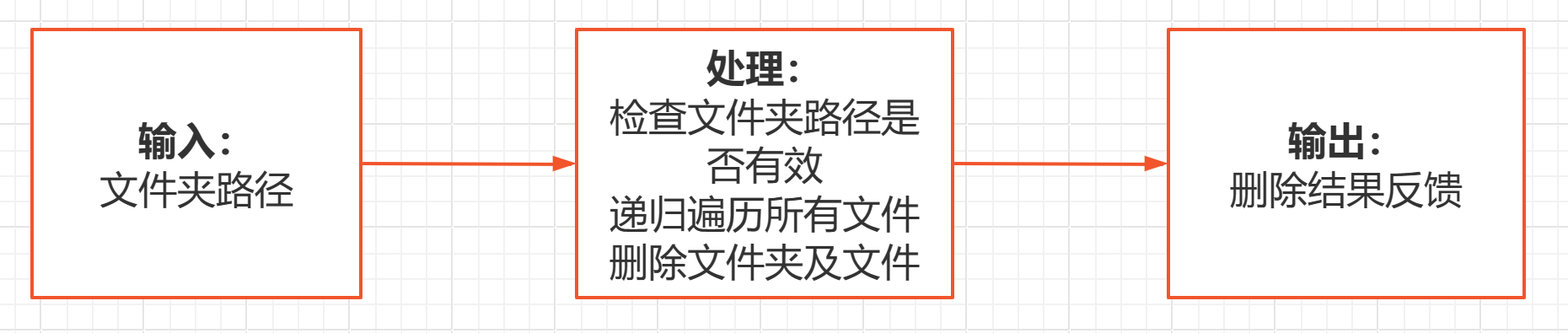
程序应具有以下功能：

输入：用户提供要删除的文件夹路径。

处理：程序删除指定文件夹及其所有内容。

输出：删除操作的结果反馈，例如成功或失败的消息。

IPO图：

8.3输入项：

名称：文件夹路径

数据类型和格式：字符串

数据值的有效范围：有效的文件夹路径

8.4输出项：

名称：删除结果反馈

数据类型和格式：字符串

数据值的有效范围：成功/失败消息

8.5算法：

删除文件夹的算法如下：

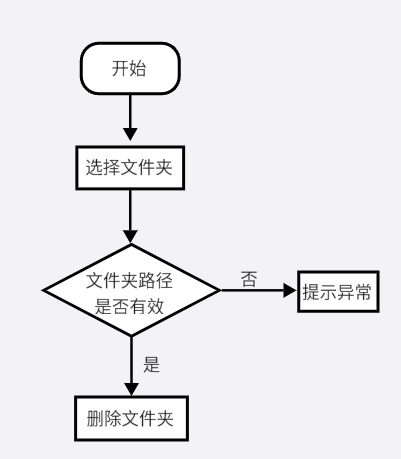
检查文件夹路径的有效性。

递归地遍历文件夹内容。

删除所有文件和子文件夹。

删除空文件夹。

8.6流程逻辑：



8.7限制条件：

用户必须具有足够的权限来删除指定文件夹。

文件夹路径必须存在，否则删除操作将失败。

8.8尚未解决的问题：

如何处理在删除过程中出现的错误，例如无法删除某个文件的情况。

是否需要提供备份或撤销功能，以防用户错误删除文件夹。

9 重命名文件

9.1程序描述：

“重命名文件”程序的设计旨在提供用户一个简便的工具，用于批量重命名文件或文件夹。该程序的主要目的是提高用户在管理文件时的效率和整理文件命名结构。该程序是非常驻内存的，是一个子程序使用。它是可重入的，允许用户多次调用并执行文件重命名操作。该程序采用顺序处理方式，逐一对用户指定的文件或文件夹进行重命名操作。

9.2功能：

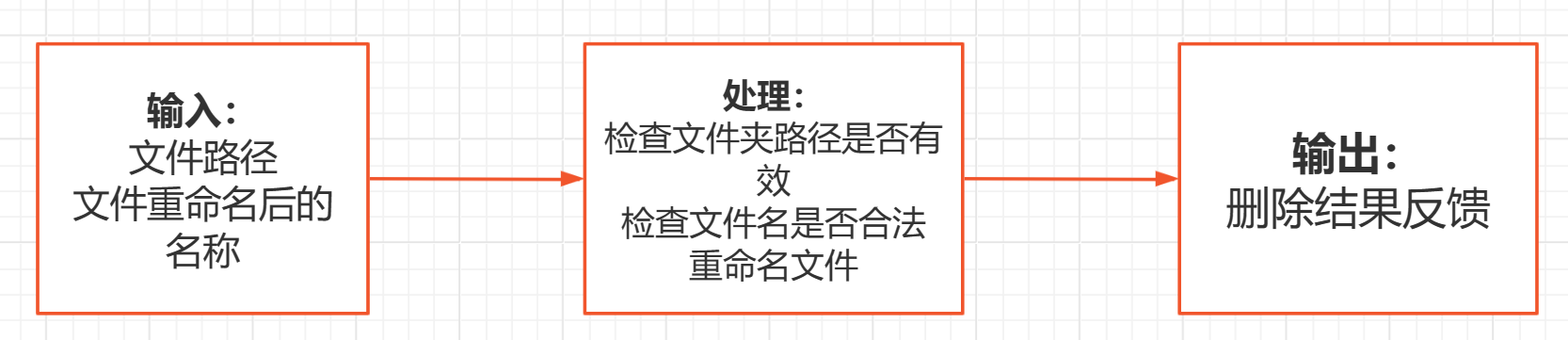
该程序的主要功能包括：

输入：接受用户指定的文件或文件夹路径。

处理：根据用户指令，对文件进行重命名。

输出：显示重命名操作的结果，包括成功和失败的信息。

IPO图：

9.3输入项：

文件路径：字符串，表示用户指定的文件或文件夹路径。

命名指令：字符串，表示重命名后的文件名

9.4输出项：

操作结果：字符串，显示重命名操作的结果，包括成功和失败的信息。

9.5算法：

程序采用以下算法进行文件重命名：

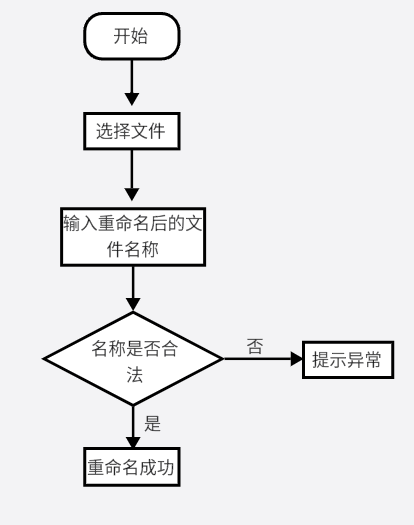
获取用户输入的文件路径和命名。

遍历指定路径下的文件或文件夹。

根据用户设定的命名规则，生成新的文件名。

执行文件重命名操作，并记录操作结果。

9.6流程逻辑：



9.7限制条件：

仅支持对文件进行重命名，不涉及文件内容修改。

用户权限受限于文件系统，需要确保有足够的权限进行文件重命名操作。

10 进入文件夹

10.1 程序描述:

该程序的目的是设计一个“进入文件夹”程序，其主要功能是让用户能够轻松地进入指定的文件夹。程序的特点包括非常驻内存、是子程序、可重入、无覆盖要求，采用顺序处理。

10.2 功能:

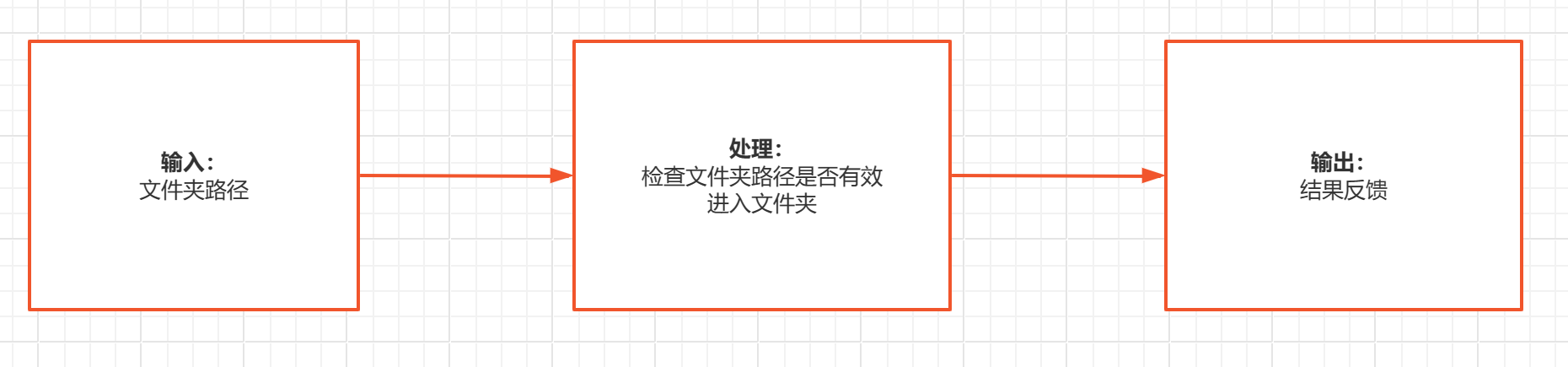
程序应具有以下功能：

输入: 接收用户输入的目标文件夹路径。

处理: 定位并进入用户指定的文件夹。

输出: 反馈用户是否成功进入文件夹。

IPO图：

10.3 输入项:

名称: 文件夹路径

数据类型和格式: 字符串

数据值的有效范围: 任何合法文件夹路径

10.4 输出项:

名称: 操作结果

数据类型和格式: 字符串

数据值的有效范围: 成功/失败消息

10.5 算法:

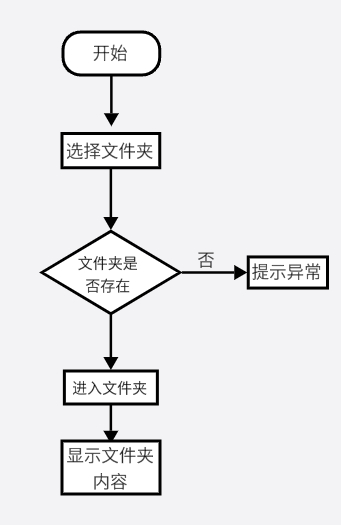
接收用户输入的文件夹路径。

检查文件夹是否存在。

若存在，进入文件夹；若不存在，输出错误消息。

反馈成功/失败消息。

10.6 流程逻辑:



10.7 限制条件:

程序需要有足够的权限来进入指定的文件夹。

用户提供的文件夹名称应符合操作系统的合法文件夹命名规范。

11 返回上一级

11.1 程序描述：

本程序的设计目的是创建一个“返回上一级文件夹”的功能，旨在提供用户在文件系统中导航时快速返回到当前文件夹的上一级。该程序非常驻内存，是一个简单的单次执行程序。它是子程序，是不可重入的，顺序处理用户输入。

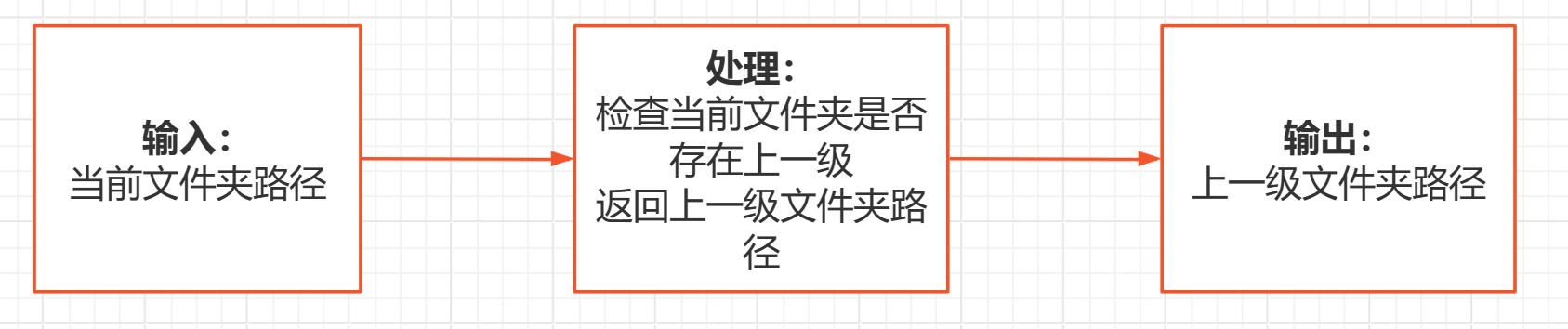
11.2 功能：

该程序的功能是在用户输入的当前文件夹路径下，返回上一级文件夹的路径。可以采用IPO图表示：

输入(Input): 当前文件夹路径

处理(Process): 返回上一级文件夹路径

输出(Output): 上一级文件夹路径



11.3 输入项：

名称：当前文件夹路径

数据类型和格式：字符串

11.4 输出项：

名称：上一级文件夹路径

数据类型和格式：字符串

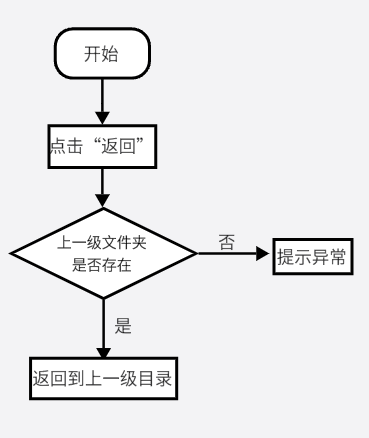
11.5 算法：

接收用户输入的当前路径。

解析路径，识别上一级文件夹。

返回上一级文件夹的路径。

11.6 流程逻辑：



11.7 限制条件：

输入的路径必须是有效的文件夹路径。

不考虑特殊字符或路径格式的处理。

12 上传文件

12.1 程序描述

该程序是一个用于上传文件的应用程序，主要设计的目的是允许用户将本地文件上传到服务器。这个程序的特点包括：

非常驻内存： 程序在需要上传文件时启动，完成任务后退出，不常驻内存。

是子程序： 是其他程序的子程序。

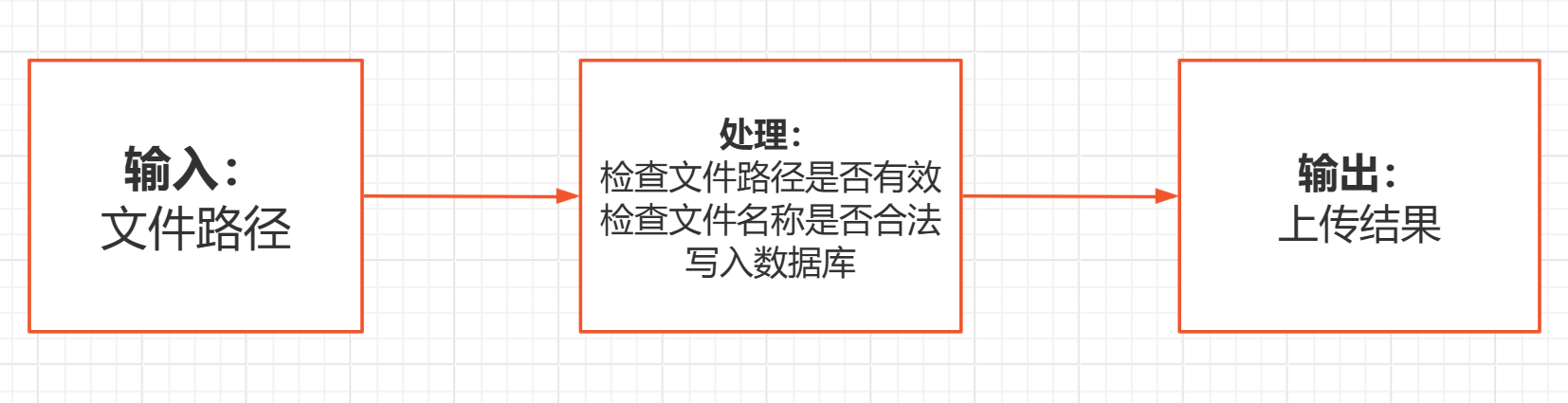
可重入性： 允许多个用户同时上传文件，具有可重入性。

无覆盖要求： 不覆盖已存在的文件，保留原文件。

顺序处理： 用户上传文件的请求将按照其提交的顺序进行处理。

12.2 功能

IPO图：

12.3 输入项

文件路径：

名称：文件路径

数据类型和格式：字符串

数据值的有效范围：任意有效的本地文件路径

文件：

名称：文件

数据类型和格式：不限

12.4 输出项

上传结果：

名称：上传结果

数据类型和格式：字符串

数据值的有效范围：成功、失败等

12.5 算法

上传文件程序的算法如下：

接收用户输入的文件路径。

检查文件路径的合法性，确保文件存在且可读。

连接到服务器。

将文件发送到服务器。

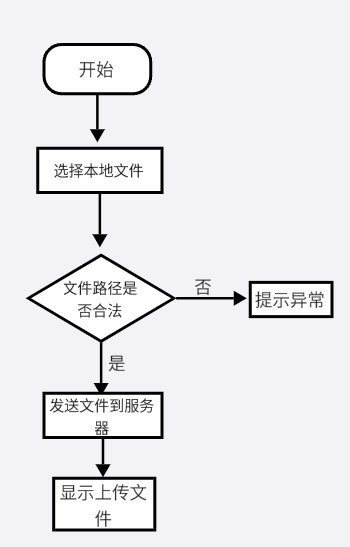
等待服务器响应，获取上传结果。

关闭与服务器的连接。

输出上传结果。

12.6 流程逻辑

以下是上传文件程序的流程图：



12.7 限制条件

用户必须有合法的访问权限来读取指定的文件。

上传文件大小受服务器限制，超出限制将导致上传失败。

13 删除文件

13.1 程序描述

目的与意义

该程序的设计目的是实现一个文件删除工具，用于删除指定路径下的文件。其主要意义在于提供用户一种方便、安全的方式来管理文件，确保不再需要的文件可以被彻底删除，释放存储空间。

特点

非常驻内存： 本程序非常驻内存，只在运行时占用系统资源。

子程序： 是其他程序的子程序。

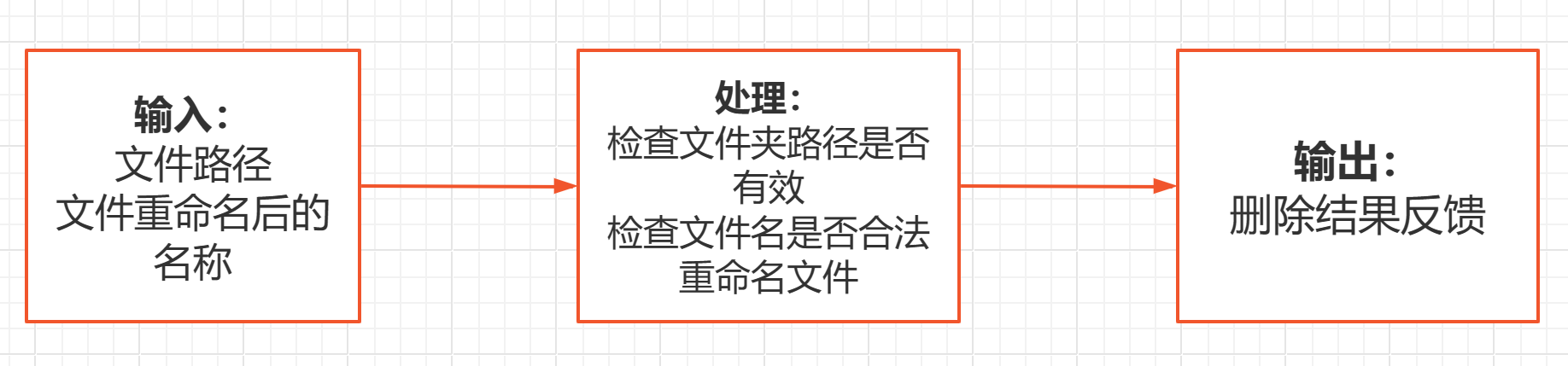
可重入性： 具备可重入性，允许多次执行而不产生冲突。

无覆盖要求： 不对文件进行覆盖操作，仅删除指定文件。

顺序处理： 以顺序方式处理文件删除请求。

13.2 功能

IPO图

13.3 输入项

删除路径

名称：文件路径

数据类型： 字符串

有效范围： 有效的文件路径

13.4 输出项

删除结果

名称： 删除结果

数据类型： 布尔值（成功/失败）

13.5 算法

删除文件算法

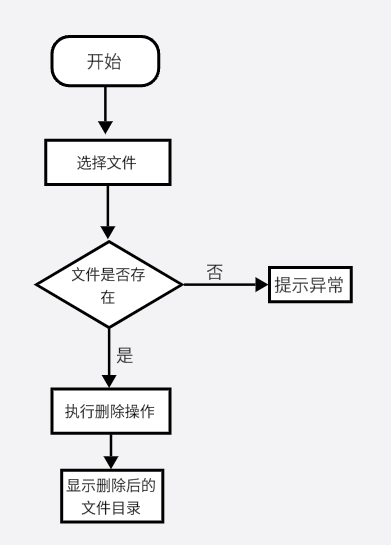
接收用户输入的文件路径。

检查文件路径的合法性。

如果文件存在，执行删除操作；否则，输出删除失败。

输出删除结果。

13.6 流程逻辑



13.7 限制条件

程序需要有足够的权限来删除指定文件。

14 下载文件

14.1 程序描述

目的意义：

该程序的目的是设计一个“下载文件”程序，使用户能够通过输入指定的文件链接或路径，将文件下载到本地设备。主要方便用户获取在线资源，提高文件传输效率。

特点：

非常驻内存： 程序在执行下载任务后可以释放占用的内存。

子程序： 该程序是其他程序的子程序。

可重人的： 允许用户在同一时间下载多个文件，支持并发下载。

无覆盖要求： 下载的文件不会覆盖本地已存在的同名文件。

顺序处理： 用户的下载请求按照其提交顺序依次处理。

14.2 功能

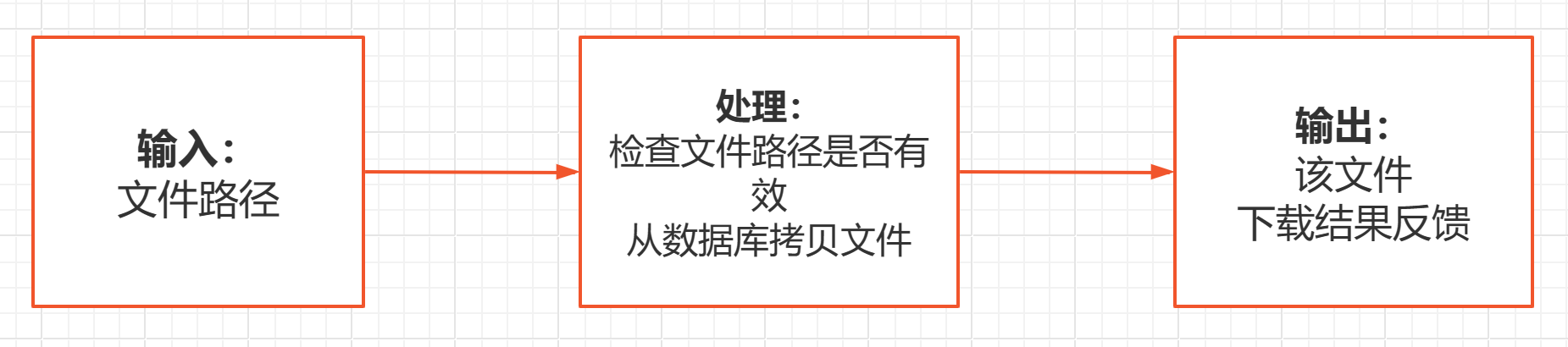
程序具有以下功能：

输入：用户提供文件的链接或路径。

处理：根据用户输入下载文件。

输出：下载完成的文件保存在本地。

IPO图：



14.3 输入项

文件链接或路径：

名称： 文件链接或路径

数据类型和格式： 字符串

数据值的有效范围： 有效文件路径

14.4 输出项

文件：

名称：文件

数据类型：不限

下载完成提示：

名称： 下载状态

数据类型： 字符串

数据值的有效范围： 成功/失败等状态信息

14.5 算法

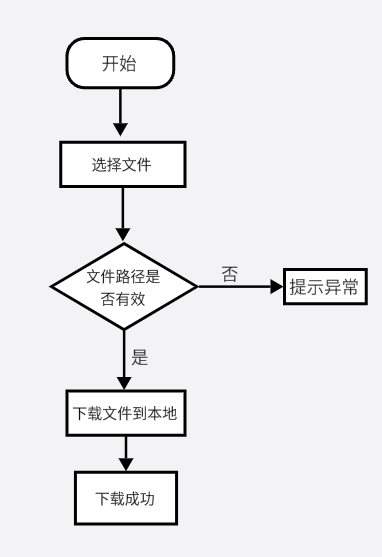
获取用户输入的文件链接或路径。

检查链接的有效性或路径的存在性。

如果有效，下载文件到本地。

输出下载状态信息。

14.6 流程逻辑



14.7 限制条件

下载速度受网络条件限制。

下载的文件大小受本地存储空间限制。

15 分享文件

15.1 程序描述:

本程序的目的是设计一个“分享文件”程序，旨在提供用户便捷、安全、高效地分享文件的功能。该程序的特点包括非常驻内存、顺序处理、可重入、无覆盖要求。用户可以通过该程序选择文件并分享给其他用户，确保数据的安全传输和妥善管理。

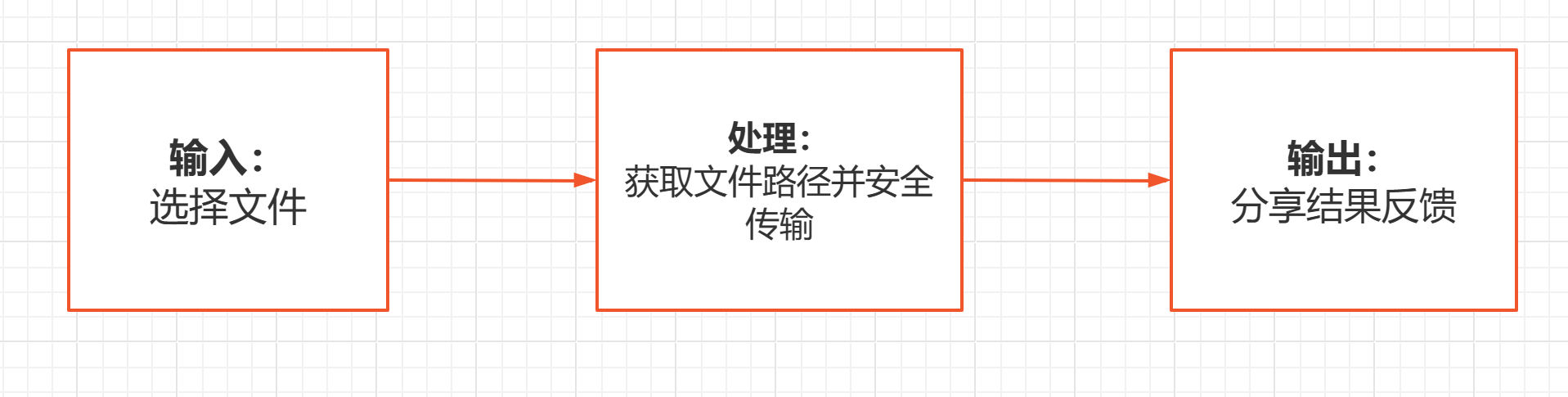
15.2 功能:

输入：用户选择要分享的文件。

处理：对文件进行安全传输和管理。

输出：分享结果反馈

IPO图：

15.3 输入项:

名称：文件

数据类型和格式：字符串

数据值的有效范围：有效的本地文件路径

15.4 输出项:

操作结果：字符串，显示分享操作的结果，包括成功和失败的信息

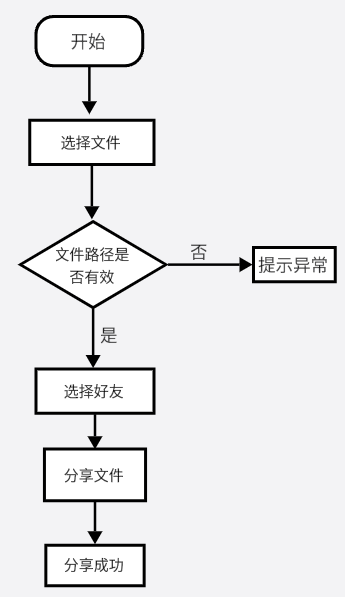
15.5 算法:

用户通过GUI选择要分享的文件。

程序获取文件路径并进行安全传输。

分享成功后生成通知消息。

15.6 流程逻辑:



15.7 限制条件:

文件大小限制：受操作系统和网络限制，大文件可能导致传输失败。

网络连接：需要稳定的网络连接以确保文件安全传输。

安全性：程序需采用加密技术确保文件传输的安全性。

15.8 尚未解决的问题:

大规模文件传输：处理大文件可能导致性能问题，需要优化。

16 移动文件

16.1程序描述

移动文件程序的设计旨在提供一种方便快捷的方式，使用户能够在计算机系统中轻松地移动文件。该程序的设计目的在于简化文件管理过程，提高用户体验。该程序非常驻内存，属于子程序，支持可重入，采用顺序处理方式。

16.2功能

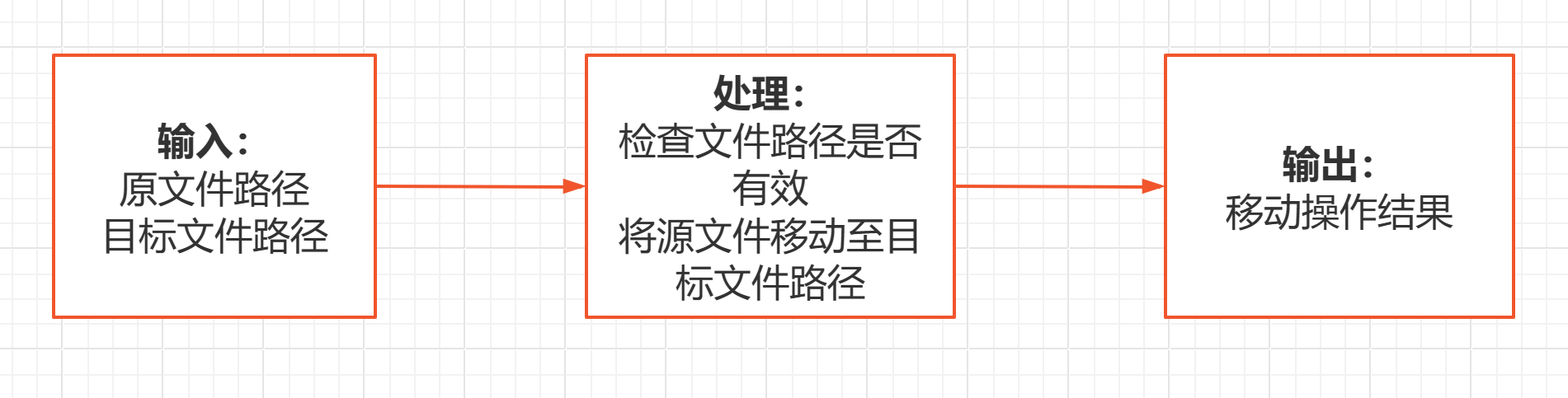
该程序的主要功能包括：

输入：接收用户指定的源文件路径和目标文件路径。

处理：根据用户输入移动源文件至目标文件路径。

输出：向用户显示移动文件的结果，例如成功或失败的消息。

IPO图形式：

16.3输入项

源文件路径：

名称：源路径

数据类型：字符串

有效范围：有效的文件路径

目标文件路径：

名称：目标文件路径

数据类型：字符串

有效范围：有效的文件路径

16.4输出项

移动结果：

名称：移动结果

数据类型：布尔型

数据格式：成功/失败消息

16.5算法

移动文件程序采用以下算法：

1. 从用户输入中获取源文件路径和目标文件路径

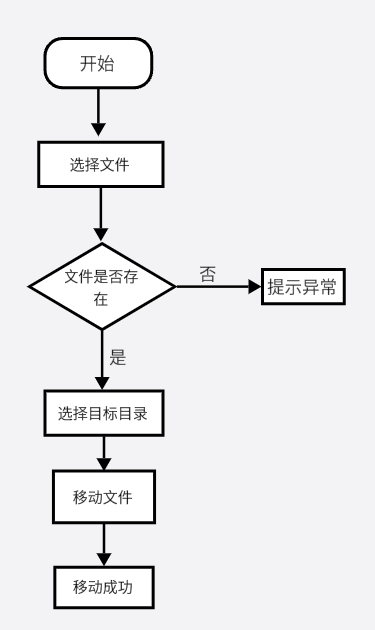
2. 检查源文件是否存在，目标文件路径是否有效。

16. 如果源文件存在且目标路径有效，则将源文件移动至目标路径。

4. 输出移动结果消息，指示移动是否成功。

16.6流程逻辑

流程图如下：



16.7限制条件

该程序的运行受到以下限制条件：

操作系统权限：需要足够的权限来读取源文件并在目标路径写入文件。

硬盘空间：目标路径需要有足够的空间来容纳源文件。