**系统设计**

# 引言

文件备份系统设计说明书是在指导老师和小组成员以及相关人员的帮助下完成的。在撰写本文档前，笔者全面了解了设计文档的要求，通过小组研讨确认了软件系统的设计架构。

通过阅读本说明书，可以全面了解本软件实现需求功能的方式，软件开发的详细流程以

及系统整体设计架构。

## 编写目的

本软件系统概要设计报告是基于数据备份软件需求规格说明书编写的。系统设计说明书在满足功能需求的前提下，对被开发软件的系统进行了合理设计，对其具体功能实现方法、软件开发流程进行了适当规定，为软件的开发提供了基本模板和具体范式。读者通过阅读本书可以对开发流程、 软件架构有所了解。

## 项目风险

首要风险承担者及其各自在本阶段所需要承担的主要风险包括：

* 任务提出者：系统设计与预期设计不一致。
* 软件开发者：功能实现与系统设计不一致。
* 产品使用者：使用需求与软件设计不一致。

## 预期读者和阅读建议

可能的读者及其阅读建议包括：

* 用户：适用于具体命令行部分。
* 开发人员：适用于全文档。
* 项目经理：适用于全文档。
* 测试人员：适用于全文档。
* 文档编写入员：适用于全文档。

## 参考资料

软件产品需求分析报告时所用到的参考文献及资料包括：

* 本项目的需求分析说明书。
* 本项目参考的课程 PDF。
* 《南京市交通局信息化数据库建设规范》

# 词汇表

LZ78算法：基于字典的压缩方法

Sqlite：轻型数据库

# 设计概述

本节描述了现有开发条件和需要实现的目标，说明了进行概要设计时应该遵循的设计原则和必须采用的设计方法。

## 限制和约束

该项目主要受到如下约束

● 技术条件；本科开发水平

● 人数：3

● 开发环境(包括：工具和平台)：Linux

● 时间限制：4 周

● 开发语言限制：C++ 14

● 其他限制：不使用外部依赖库

在以上条件下，项目应完成一个具有备份还原、打包解包、网络备份功能的文件备份软件。

## 设计原则和设计要求

本软件系统进行概要设计的原则，包括以下几方面的内容：

● 命名规则；文件命名具有可读性，如 Nettools、Sever等

● 模块独立性原则；模块可独立运行、独立测试功能

● 数据库设计规则；第三范式

● 必须的安全措施：拒绝非法数据的写入，如空用户名、空密码

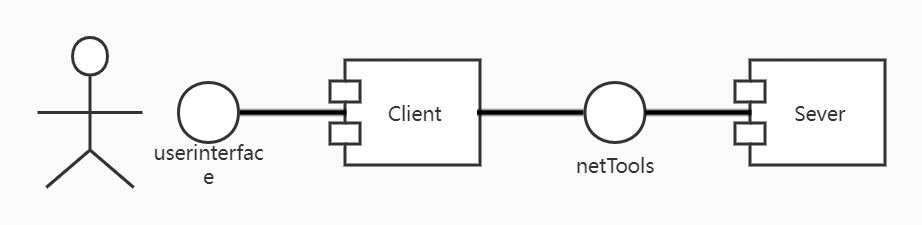
● 系统易操作性要求；用户操作界面简洁

# 总体设计

## 系统结构设计

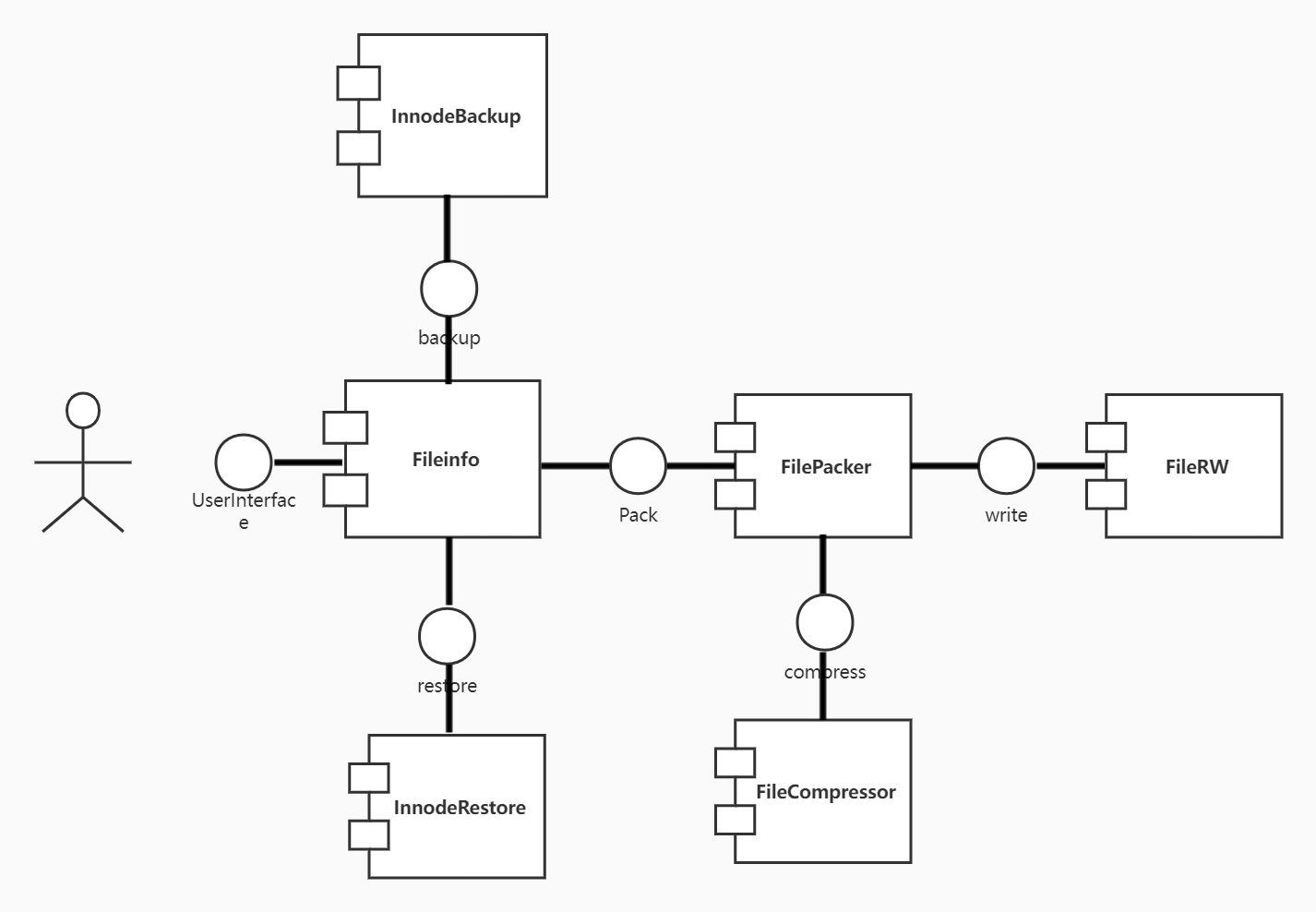
此处提供了高层系统结构的描述，使用构件图显示了本系统主要的组件及组件间的交互。

### 顶层系统结构

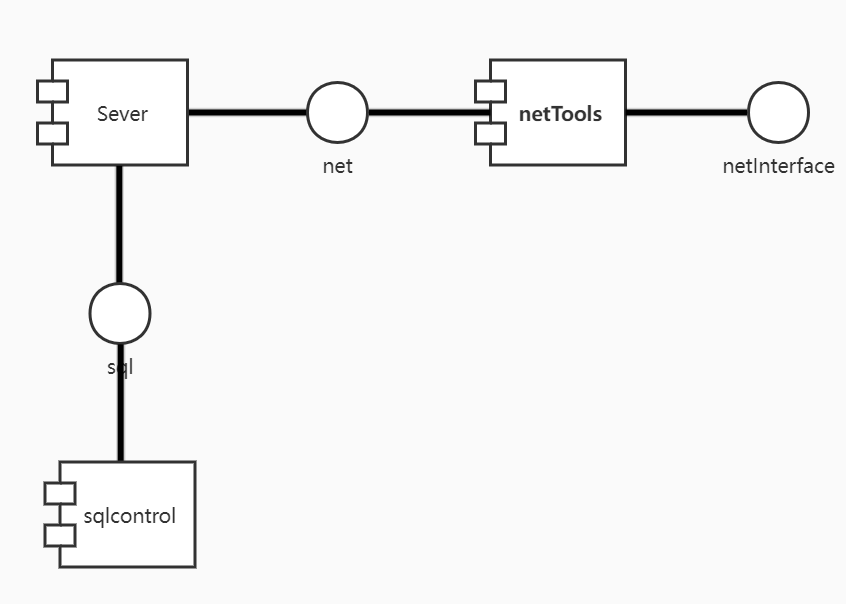


本软件提供了网络备份功能，因而系统的顶层系统结构由客户端和服务器两部分构成。

### 客户端结构

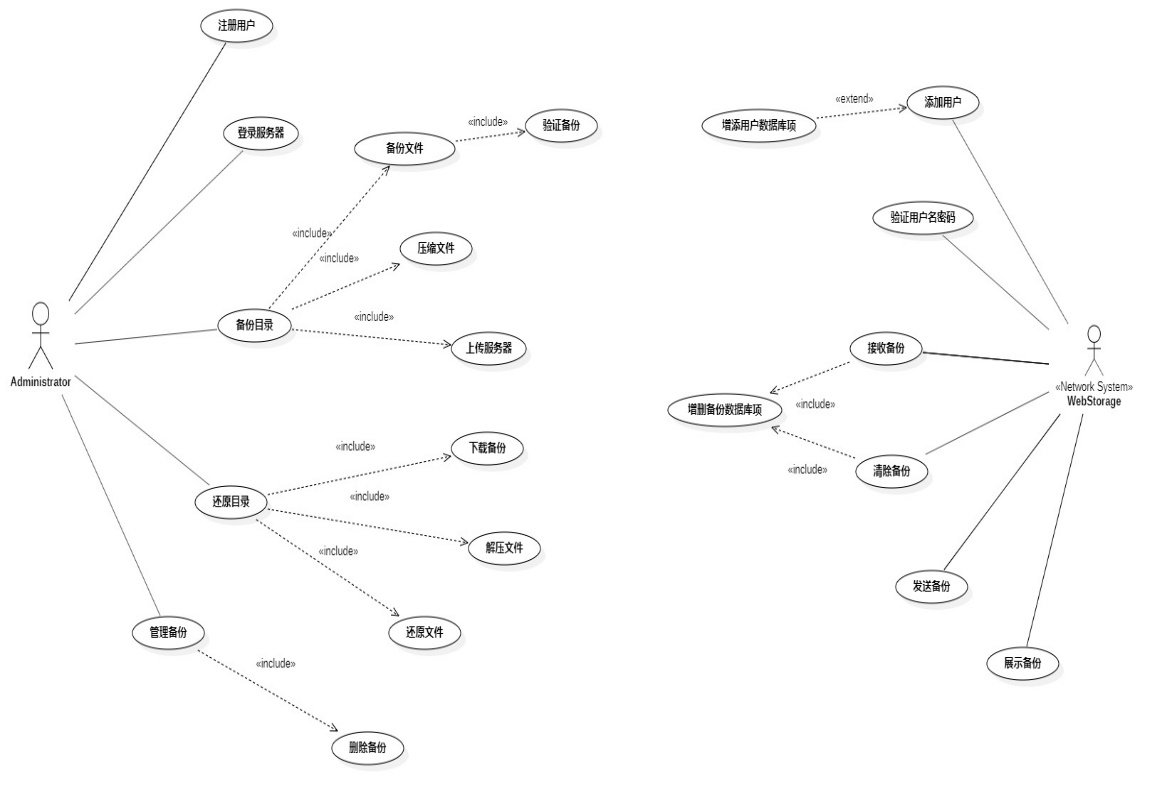


### 服务器结构



### 数据库设计

数据库逻辑设计：



数据库分布：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 数据库  编号 | 数据库  管理系统  名称 | 数据库  管理系统  版本号 | 数据库  英文名称 | 数据库  中文名称 | 数据库  安装  物理位置 |
| 1 | sqlcontrol | Sqlite3 | sqlcontrol | 数据库控制 | D:\sqlite |

基表1设置：

|  |  |  |
| --- | --- | --- |
| User | | |
| PK | ID | Varchar |
|  | UserName | Varchar |
|  | Password | Varchar |

基表2设置：

|  |  |  |
| --- | --- | --- |
| History | | |
| PK | ID | Varchar |
|  | Time | Date |
|  | Path | Varchar |
|  | Size | Varchar |
| FK | fk\_user | Varchar |
|  | OldPath | Varchar |

## 系统界面

系统界面为命令行界面。

进入界面后会有以下提示。

|  |
| --- |
| Welcome to use Backup Tools!"  Choose the function you want to use:  1. Sign up  2. Log in  3. Backup  4. Restore  5. Maintenance backup  6. Exit" |

根据主界面提示，用户可以输入对应数字来继续操作。

如输入数字1，会有以下提示。

|  |
| --- |
| Sign up:  Please enter your username:"  Please enter your password:  username:  username length:  password:  password length: |

若成功注册，则会有相应提示：“Signup successfully!Will go back to main menu in 3 seconds.”；若注册失败则会提示：“Signup failed!Will go back to main menu in 3 seconds.”

由于系统界面较简单且清晰，因此不做更详细的介绍，根据提示做操作即可。

## 约束和假定

### 4.3.1 客户约束：

依赖库约束：项目不能使用第三方依赖库。

时间约束：4 周

### 4.3.2 其他约束语言约束：

统一使用 C++（C12）进行软件开发。由于 QT 支持对 C++的编译，为了使项目

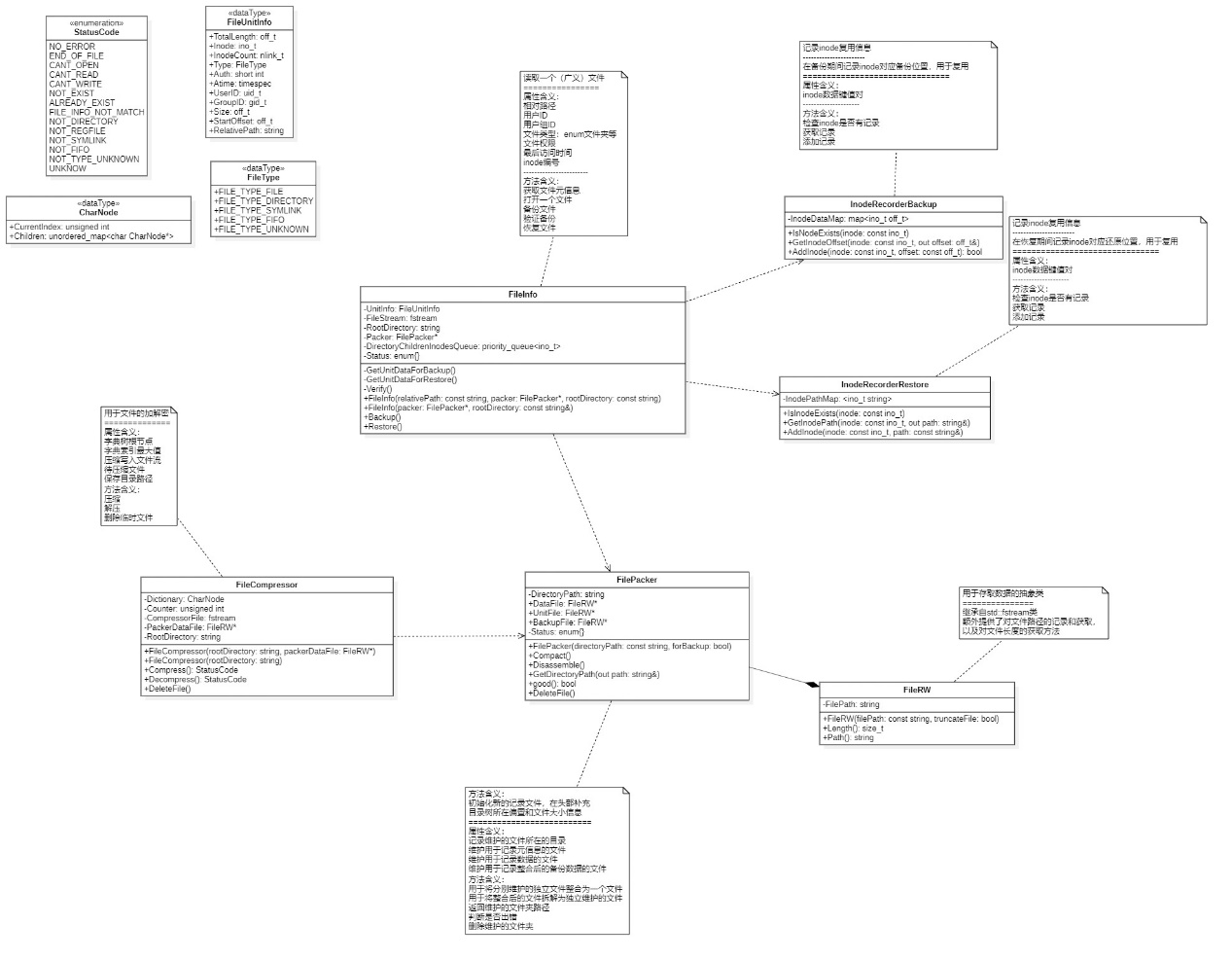
代码实现、管理较为方便，均采用 C++作为开发语言。

数据库约束：选用 Sqlite 作为项目数据库。Sqlite 为 本系统的数据库，其具有体量小、使用方便的特点，非常适合用于小型项目。

# 静态建模（对象模型）

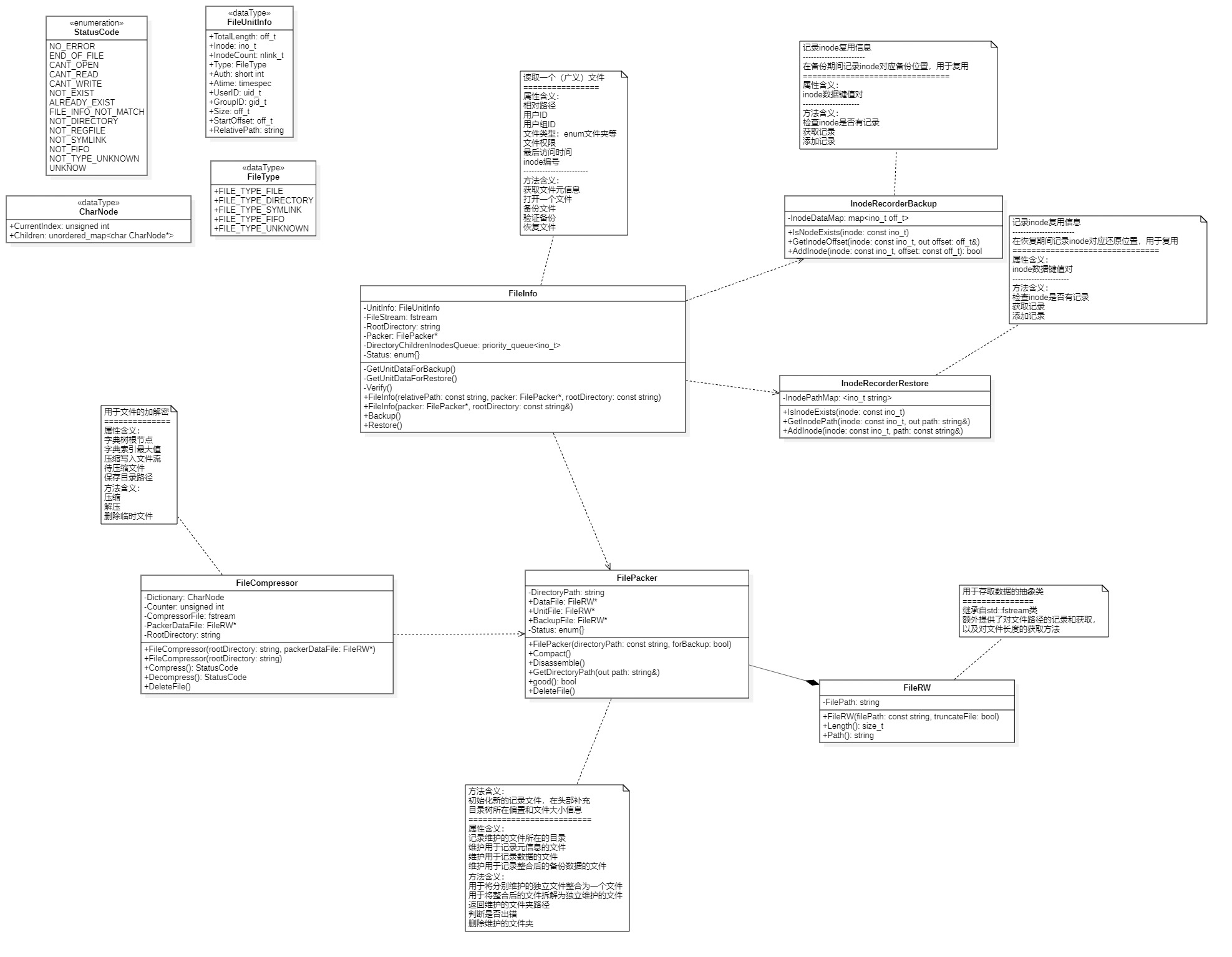
## 系统对象模型

# 类（对象）描述



## 系统中的类（对象）

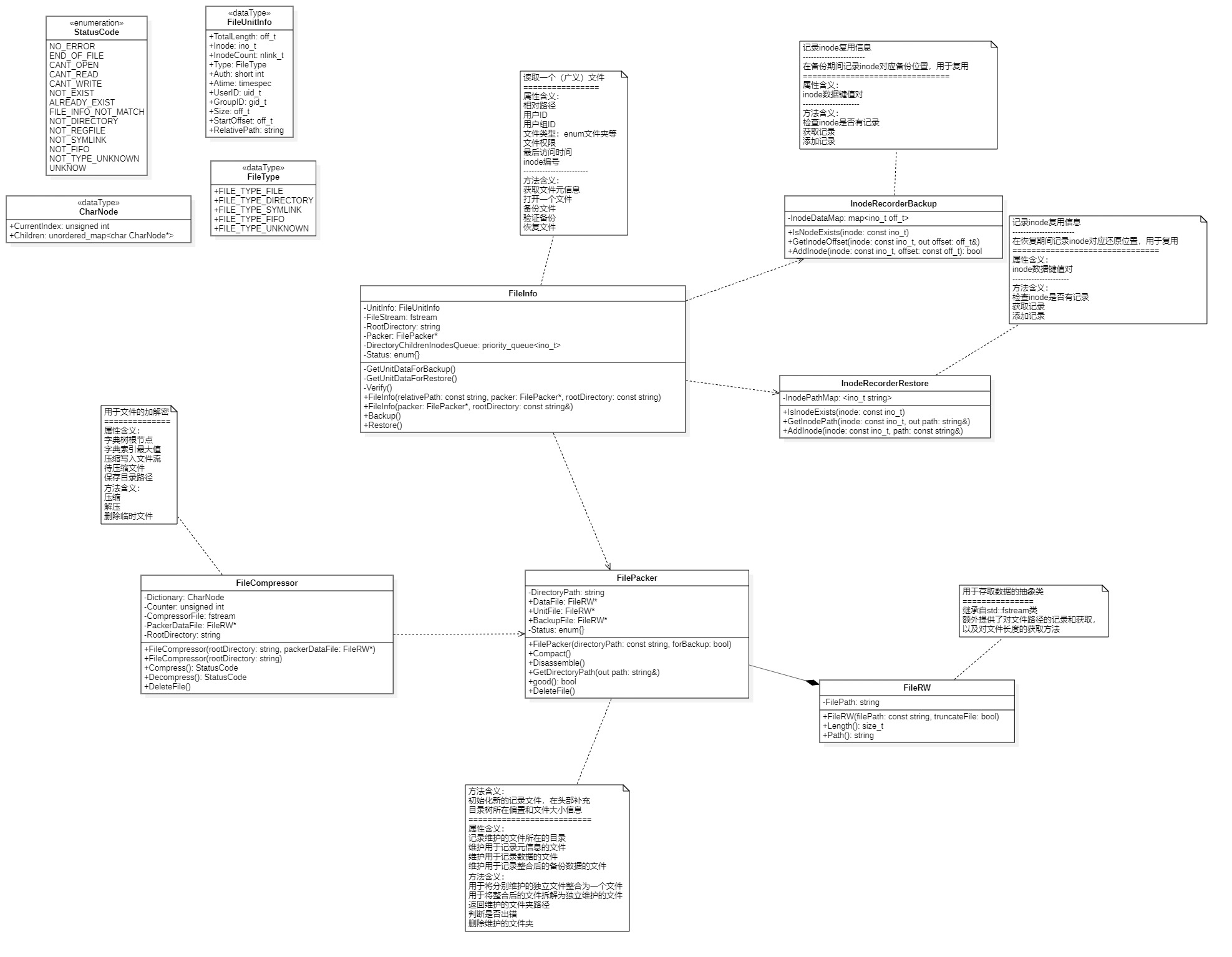
### FileInfo对象：



对象的说明如下：读取一个（广义文件）

|  |  |  |
| --- | --- | --- |
| 属性含义 | 文件类型 | 方法含义 |
| 相对路径 | Enum文件夹等 | 获取文件元信息 |
| 用户ID | 文件权限 | 打开一个文件 |
| 用户组ID | 最后访问时间 | 备份文件 |
|  | Inode编号 | 验证备份 |
|  |  | 恢复文件 |

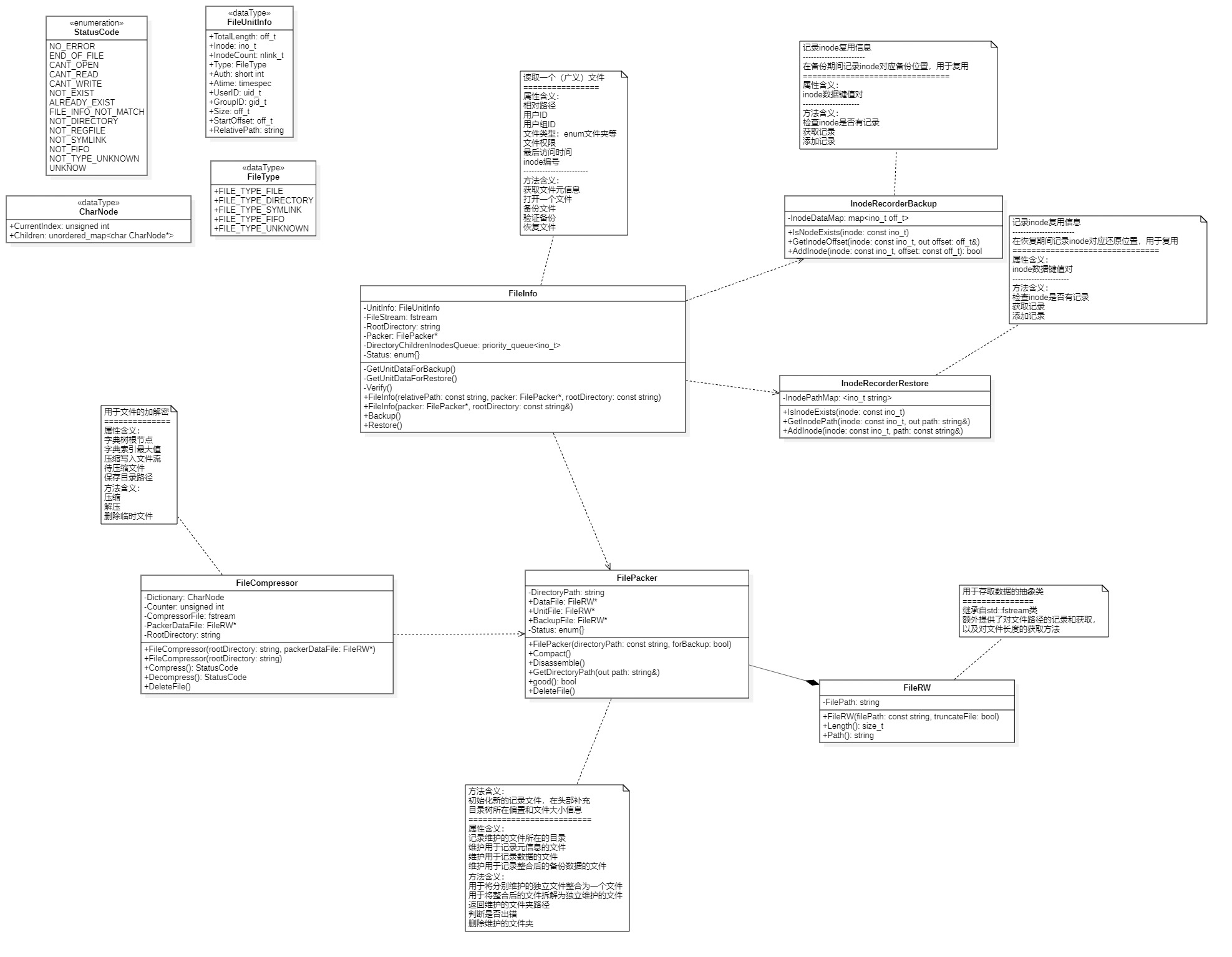
### FliePacker对象：



对象的说明如下：打包一个文件

|  |  |
| --- | --- |
| 属性含义 | 方法含义 |
| 记录维护的文件所在的目录 | 初始化新的记录文件，在头部补充目录树所在偏置和文件大小信息 |
| 维护用于记录元信息的文件 | 用于将分别维护的独立文件整合为一个文件 |
| 维护用于记录数据的文件 | 用于将整合后的文件拆解为独立维护的文件 |
| 维护用于记录整合后的备份数据的文件 | 返回维护的文件夹路径 |
|  | 判断是否出错 |
|  | 删除维护的文件夹 |

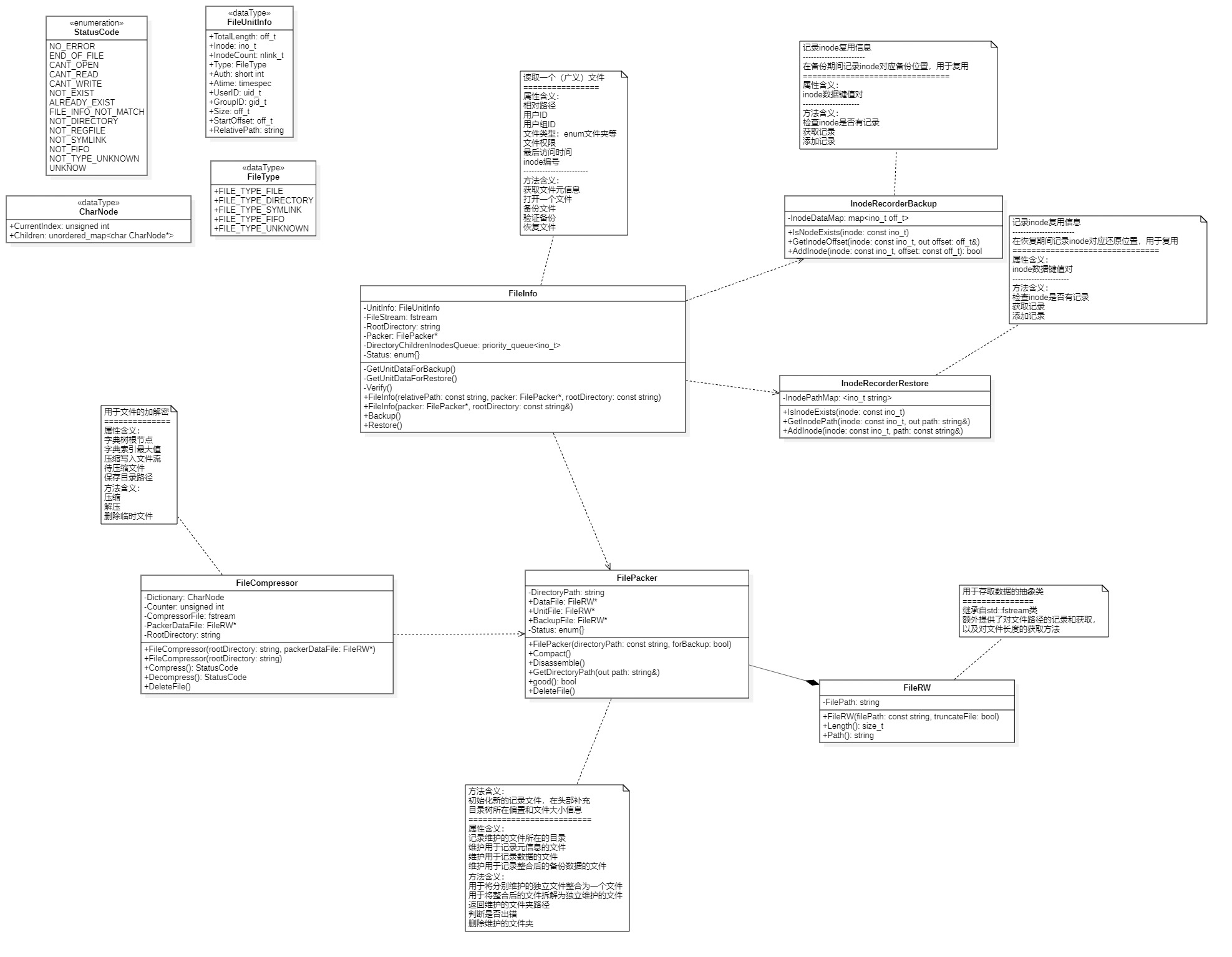
### FileCompress对象：



对象的说明如下：用于文件的加解压

|  |  |
| --- | --- |
| 属性含义 | 方法含义 |
| 字典树根节点 | 压缩 |
| 字典索引最大值 | 解压 |
| 压缩写入文件流 | 删除临时文件 |
| 待压缩文件 |  |
| 保存目录路径 |  |

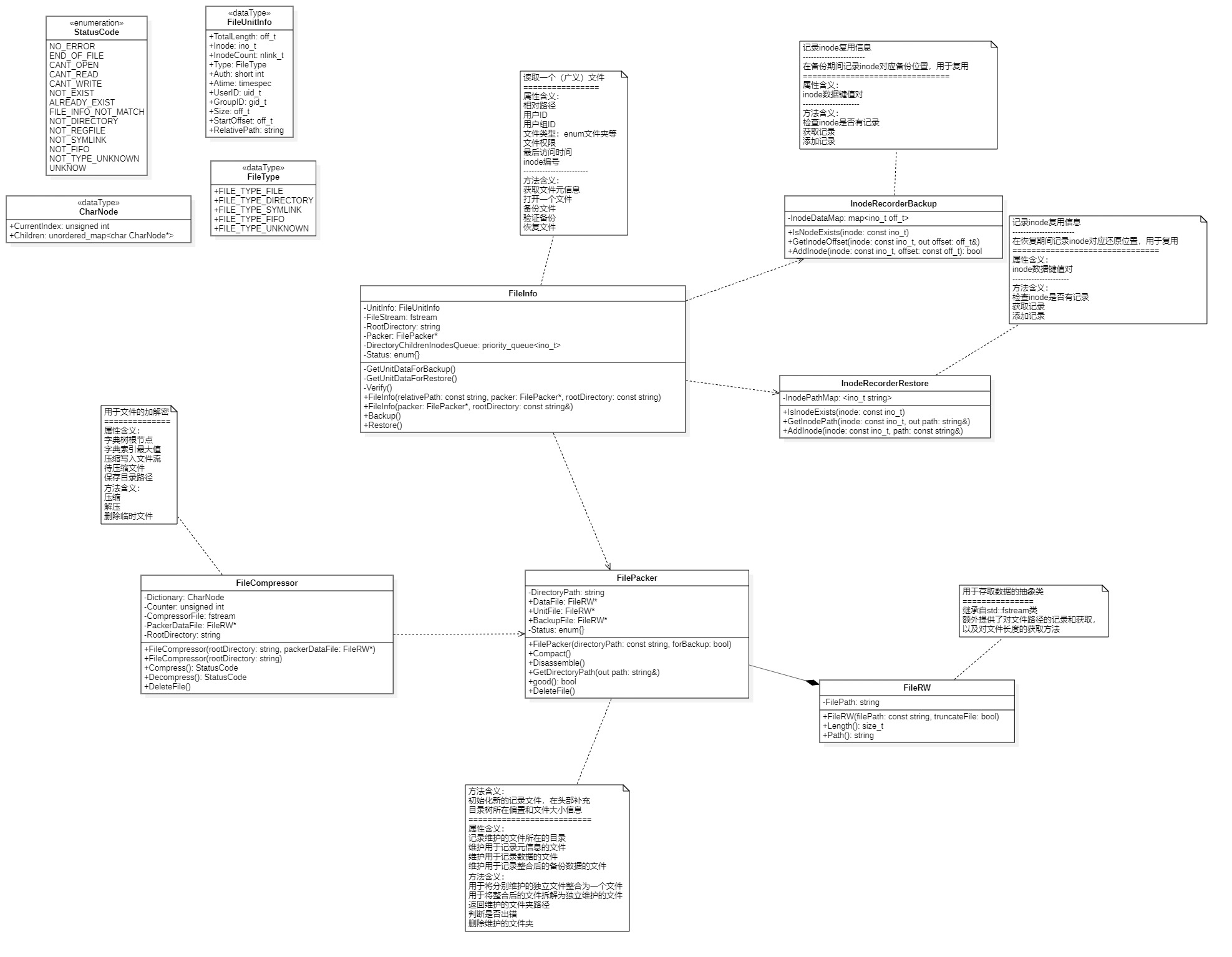
### InodeRecorderBackcup对象：



对象的说明如下：在备份期间记录inode对应备份位置，用于复用。

|  |  |
| --- | --- |
| 属性含义 | 方法含义 |
| Inode数据键值对 | 检查inode是否有记录 |
|  | 获取记录 |
|  | 添加记录 |

### InodeRecorderRestore对象：



对象的说明如下：在恢复期间记录inode对应备份位置，用于复用。

|  |  |
| --- | --- |
| 属性含义 | 方法含义 |
| Inode数据键值对 | 检查inode是否有记录 |
|  | 获取记录 |
|  | 添加记录 |

# 动态建模

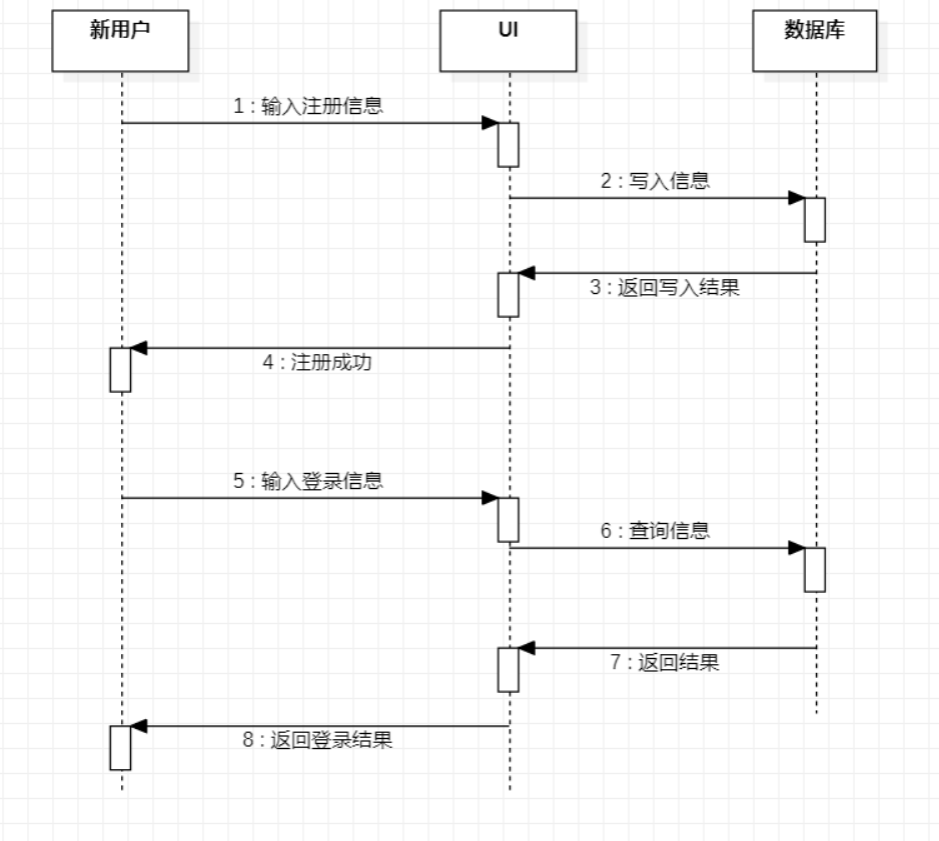
这部分的作用是描述系统如何响应各种事件，确定了不同的场景（Scenario）。

## 场景（Scenarios）

### 场景：新用户注册、登陆场景

描述：软件新用户进行登录、注册的场景

顺序图：



详细说明：

1. 用户输入注册信息

2、将信息写入数据库，对非法数据进行拒绝

3、返回写入结果

4、若成功写入则注册成功，若失败则返回 1

5、输入登录信息

6、在数据库中查询、比对信息

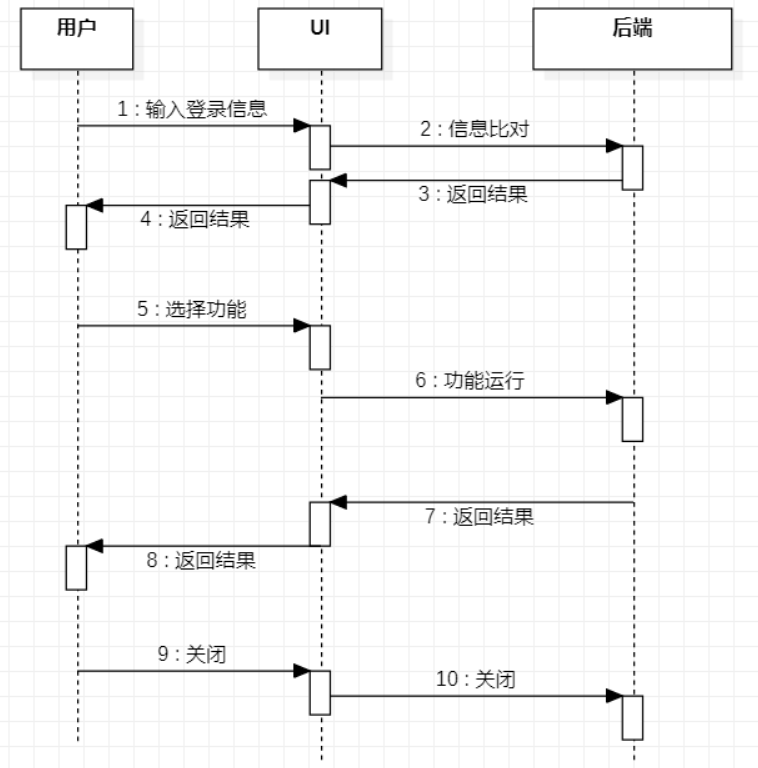
7、返回查询、比对结果

8、若成功则登录成功，若失败则返回

### 场景2：用户登录，使用软件

描述：软件已注册用户登录并使用软件

顺序图：



详细说明：

1、用户输入登录信息

2、后端进行数据比对

3、返回结果

4、用户登录成功则进入下一步，失败则返回 1

5、用户选择软件功能，备份、还原、校验

6、后端进行功能运行

7、返回运行结果

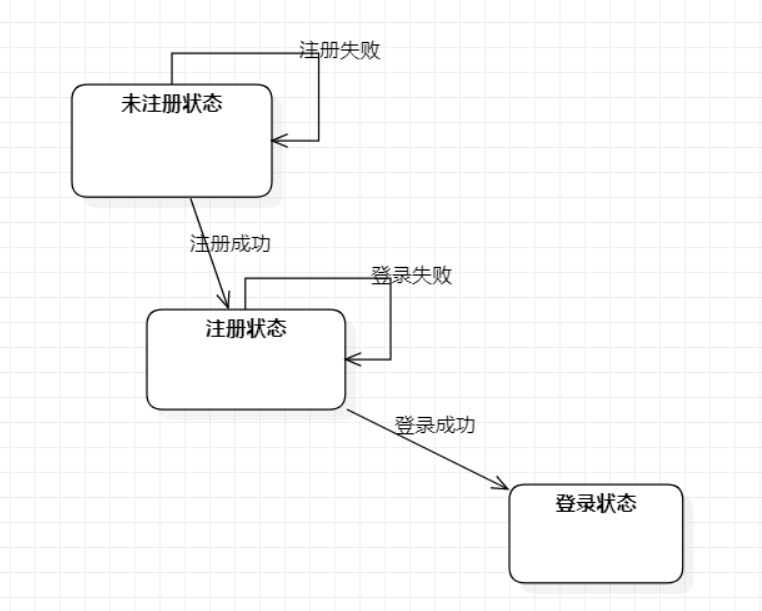
8、用户查看结果

9、用户关闭软件

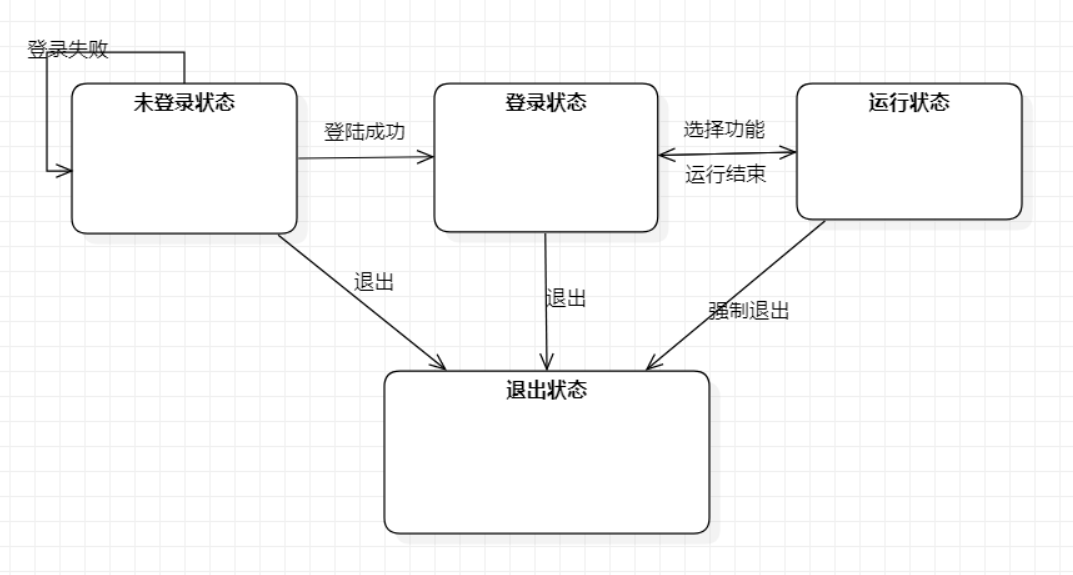
10、服务器关闭该端口

## 状态图

### 状态图1：



### 状态图2：



# 支撑环境

## 数据库管理系统

本次项目采用 QT 内置的 SQLite 数据库作为项目数据库

数据库管理系统、以及安装配置情况如下：

● 产品名称：SQLite

● 发行公司：SQLite Consortium 开源项目

● 版本号：Version 3.36.0

● 语言或代码集：C++ 14

● 安装位置：G++

● 配置参数：

|  |
| --- |
| #include<sqlite3.h>  int sqlcontrol::createTable(){  excuteSql("create table user(ID INTEGER PRIMARY KEY AUTOINCREMENT,USERNAME TEXT NOT NULL UNIQUE,PASSWORD CHAR(50))");  excuteSql("create table history(ID INTEGER PRIMARY KEY AUTOINCREMENT,TIME TEXT NOT NULL,USER INT,PATH TEXT NOT NULL, OLDPATH TEXT NOT NULL, SIZE INT NOT NULL, CONSTRAINT fk\_user FOREIGN KEY (USER) REFERENCES user(ID))");  return 1;  } |

## 开发工具、中间件以及数据库接口

**编译环境 ：**

产品名称：G++

发型厂商：GNU

版本号：G++10.3

语言：C++ 14

**数据库接口：**

数据库版本：Sqlite3

接口协议：Tcp

接口端口号：8888

接口配置：#include<sqlite3.h>

**虚拟机运行平台：**

产品名称：VMware Workstation Pro

发行厂商：vmware

版本号：Workstation 16 Pro

**虚拟机镜像：**

产品名称：Ubuntu

发行厂商：Cannoical

版本号：Ubuntu 20.04.3 LTS

## 硬件环境

● 机型；虚拟机 Ubuntu 20.04.3 LTS

● 主频；2.2Ghz

● 内存容量；2GB

● 磁盘容量；30GB

● 特殊部件；VMware Workstation

● 操作系统；Linux