|  |  |
| --- | --- |
| 卷 号 |  |
| 卷内编号 |  |
| 密 级 |  |

**系统详细设计文档**

**项目名称： 数据库管理系统**

**项目编号：**

**编写人员： 龙灏天**

**编写日期： 20181121**

**审批人员：**

**审批日期：**

目录

[1.前言 3](#_Toc390698046)

[**1.1.** **文档目的** 3](#_Toc390698047)

[**1.2.** **范围说明** 3](#_Toc390698048)

[**1.2.1.** **产品标识** 3](#_Toc390698049)

[**1.2.2.** **产品概述** 3](#_Toc390698050)

[**1.2.3.** **参考资料** 3](#_Toc390698051)

[**2.** **第0层系统设计** 3](#_Toc390698052)

[**2.1.** **系统用例** 3](#_Toc390698053)

[**2.2.** **系统功能** 4](#_Toc390698054)

[**2.3.** **层级调用关系** 4](#_Toc390698055)

[**2.4.** **需要定义的文件** 5](#_Toc390698056)

[3. 第1层系统设计 6](#_Toc390698057)

[**3.1数据库管理** 6](#_Toc390698058)

[**3.1.1.新建数据库** 6](#_Toc390698059)

[**3.1.2.修改数据库** 7](#_Toc390698060)

[**3.1.3删除数据库** 7](#_Toc390698061)

[**3.2表管理** 8](#_Toc390698062)

[**3.2.1.新建表** 8](#_Toc390698063)

[**3.2.2.删除表** 9](#_Toc390698064)

[**3.3基础记录管理** 10](#_Toc390698065)

[**3.3.1.增加字段** 10](#_Toc390698066)

[**3.3.2.修改字段** 11](#_Toc390698067)

[**3.3.3.删除字段** 12](#_Toc390698068)

[**3.4数据管理** 13](#_Toc390698069)

[**3.4.1.增加记录** 13](#_Toc390698070)

[**3.4.2.修改记录** 14](#_Toc390698071)

[4. 第2层系统设计 15](#_Toc390698072)

[4.1实体类设计 17](#_Toc390698073)

[4.1.1.数据库 17](#_Toc390698074)

[4.1.2.表 18](#_Toc390698075)

[4.1.3字段 20](#_Toc390698076)

[4.1.4记录 23](#_Toc390698077)

[4.2逻辑类设计 25](#_Toc390698078)

[4.2.1数据库 25](#_Toc390698079)

[4.2.2表 25](#_Toc390698080)

[4.2.3 字段 27](#_Toc390698081)

[4.2.4记录 28](#_Toc390698082)

[4.3业务类类设计 29](#_Toc390698083)

[4.3.1数据库 29](#_Toc390698084)

[4.3.2表 30](#_Toc390698085)

[4.3.3 字段 32](#_Toc390698086)

[4.3.4记录 33](#_Toc390698087)

[4.4工具类类设计 34](#_Toc390698088)

[4.4.1文件操作 34](#_Toc390698089)

[4.4.2工具 36](#_Toc390698090)

[4.5视图层类设计 39](#_Toc390698091)

[4.5.1 Tree 39](#_Toc390698092)

[4.5.2 Table 41](#_Toc390698093)

# 1.前言

* 1. **文档目的**

编写本文档的目的是让用户需求变得更加详实，在开发者之间达成对软件产品要求的共用理解，作为项目开发的参考文档，并作为软件测试的依据。

本文档的读者是：项目组长、实际开发人员

* 1. **范围说明**
     1. **产品标识**

数据库管理系统

* + 1. **产品概述**

数据库管理系统(Database Management System)是一种操纵和管理数据库的大型软件，用于建立、使用和维护数据库，简称DBMS。它对数据库进行统一的管理和控制，以保证数据库的安全性和完整性。用户通过DBMS访问数据库中的数据，数据库管理员也通过DBMS进行数据库的维护工作。它可使多个应用程序和用户用不同的方法在同时或不同时刻去建立，修改和询问数据库。大部分DBMS提供数据定义语言DDL（Data Definition Language）和数据操作语言DML（Data Manipulation Language），供用户定义数据库的模式结构与权限约束，实现对数据的追加、删除等操作。

* + 1. **参考资料**

需求规格说明书

1. **第0层系统设计**
   1. **系统用例**



* 1. **系统功能**



* 1. **层级调用关系**



* 1. **需要定义的文件**

1. 系统文件

数据库总系统

文件名：tibuilder.exe

文件结构：二进制PE文件，可执行。

1. 数据库描述文件

存储表的列表，包含表的基本信息。存储字段的信息

文件名：ti**.db**

文件结构：**二进制结构，存储列表信息。**

# 第1层系统设计

**3.1数据库管理**

**3.1.1.新建数据库**

(1).功能流程图



(2).功能说明

通过用户输入的数据库名新建数据库，若数据库名称重复或过长提示用户新建失败。

**3.1.2.修改数据库**

(1). 功能流程图



(2). 功能说明

将原数据库修改为用户要求的数据库名称，名称不合法则提示用户修改失败。

**3.1.3删除数据库**

(1). 功能流程图



(2). 功能说明

用户选择数据库进行删除操作。

**3.2表管理**

**3.2.1.新建表**

(1). 功能流程图



(2). 功能说明

通过用户输入的表明新建表，若表名不合法（表名重复或表名过长）则提示用户。

**3.2.2.删除表**

(1). 功能流程图



(2). 功能说明

用户选择删除表操作。

**3.3基础记录管理**

**3.3.1.增加字段**

(1). 功能流程图



(2). 功能说明

由用户输入的字段名等信息新建一个字段

**3.3.2.修改字段**

(1). 功能流程图



(2). 功能说明

根据用户输入修改字段信息

**3.3.3.删除字段**

(1). 功能流程图



(2). 功能说明

删除一个字段。

**3.4数据管理**

**3.4.1.增加记录**

(1). 功能流程图



(2). 功能说明

根据用户输入的记录信息新增一条记录。

**3.4.2.修改记录**

(1). 功能流程图



(2). 功能说明

修改一条记录的信息。

# 第2层系统设计

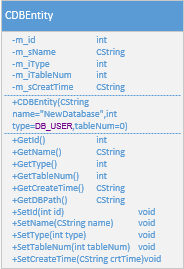
**宏定义常量**

|  |  |  |
| --- | --- | --- |
| **名称** | **值** | **说明** |
| STRING\_LENGTH | 1024 | 字符串长度 |
| INVALID | 0 | 主要用于id的不合法的 |
| YES | 1 |  |
| NO | 0 |  |
| DB\_SYSTEM | 1 | 系统数据库 |
| DB\_USER | 2 | 用户数据库 |
| DB\_INTEGER | 1 | 整型，对应int |
| DB\_BOOL | 2 | 布尔值类型，对应bool |
| DB\_DOUBLE | 3 | 双精度型，对应double |
| DB\_VARCHAR | 4 | 可变长字符类型，对应CString |
| DB\_DATETIME | 5 | 时间类型，对应SYSTEMTIME |
| DEFAULT\_FIELD\_LENGTH | 20 | 默认字段长度 |
| DEFAULT\_PAGE\_RECORDNUM | 20 | 默认每页显示条目 |
| DEFAULT\_SYS\_FILE | CString("data/dbms.sys") | 默认的系统文件名称 |
| DEFAULT\_ROOT | CString("data") | 默认数据跟目录 |
| INTEGRITY\_OK | 300 | 正常 |
| INTEGRITY\_OVER\_RANGE | 301 | 超出范围 |
| INTEGRITY\_TOO\_LONG | 302 | 字符过长 |
| INTEGRITY\_ERROR\_CHAR | 303 | 特殊字符 |
| INTEGRITY\_NULL\_VALUE | 304 | 字段空值 |
| INTEGRITY\_NOT\_UNIQUE | 305 | 字段不唯一 |
| INTEGRITY\_NOT\_PK | 306 | 不符合主键约束 |
| INTEGRITY\_ERROR\_TYPE | 350 | 错误类型 |
| DB\_EXIST | 307 | 数据库已存在 |
| DB\_NOT\_EXIST | 308 | 数据库不存在 |
| FIELD\_EXIST | 309 | 字段已存在 |
| FIELD\_NOT\_EXIST | 310 | 字段不存在 |
| RECORD\_NOT\_EXIST | 311 | 记录不存在 |
| TABLE\_EXIST | 312 | 表已存在 |
| TABLE\_NOT\_EXIST | 313 | 表不存在 |
| INITIATE\_ERROR | 314 | 初始化错误 |
| ADD\_ERROR | 315 | 添加错误 |
| MODIFY\_ERROR | 316 | 修改错误 |
| DELETE\_ERROR | 317 | 删除错误 |
| SAVE\_COUNTER\_ERROR | 318 | 保存计数器出错 |

## 4.1实体类设计

### 4.1.1.数据库

1.类图



2.方法说明

（1）构造方法

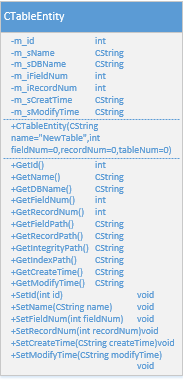
|  |  |  |
| --- | --- | --- |
| **方法名** | **功能** | **说明** |
| void CDBEntity(CString name=”NewDatabase”,int type=DB\_USER,tableNum=0) | 构造函数 |  |

（2）其他方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法名** | **输入值** | **返回值** | **说明** |
| int GetId() | 无 | 计数器的值 | 得到计数器的值 |
| CString GetName() | 无 | 数据库名称 | 得到数据库名称 |
| int GetType() | 无 | 数据库类型 | 得到数据库类型 |
| int GetTableNum() | 无 | 表的个数 | 得到表的个数 |
| CString GetCreateTime() | 无 | 创建时间 | 得到创建时间 |
| CString GetDBPath() | 无 | 数据库路径 | 得到数据库路径 |
| void SetId(int id) | id计数器的值 | 无 | 给计数器赋值 |
| void SetName(CString name) | name数据库名称 | 无 | 给数据库名称赋值 |
| void SetType(int type) | type数据库类型 | 无 | 给数据库类型赋值 |
| void SetTableNum(int tableNum) | tableNum表的个数 | 无 | 给表的个数赋值 |
| void SetCreatTime(CString crttime) | crttime创建时间 | 无 | 给创建时间赋值 |

### 4.1.2.表

1.类图



2.方法说明

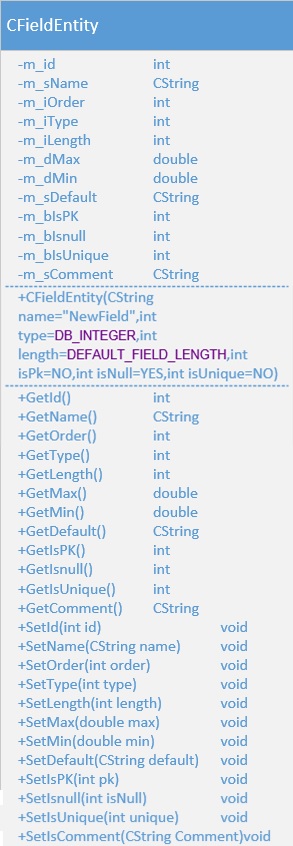
|  |  |  |
| --- | --- | --- |
| **方法名** | **功能** | **说明** |
| void CTableEntity(CString name="NewTable",int fieldNum=0,recordNum=0,tableNum=0) | 构造方法 |  |

（2）其他方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法名** | **输入值** | **返回值** | **说明** |
| int GetId() | 无 | 计数器的值 | 得到计数器的值 |
| CString GetName() | 无 | 表名 | 得到表名 |
| CString GetDBName() | 无 | 数据库名 | 得到数据库名 |
| int GetFieldNum() | 无 | 字段个数 | 得到字段个数 |
| int GetRecordNum() | 无 | 记录个数 | 得到记录个数 |
| CString GetFieldPath() | 无 | 字段路径 | 得到字段路径 |
| CString GetRecordPath() | 无 | 记录路径 | 得到记录路径 |
| CString GetIntegrityPath() | 无 | 完整性路径 | 得到完整性路径 |
| CString GetIndexPath() | 无 | 索引路径 | 得到索引路径 |
| CString GetCreateTime() | 无 | 创建时间 | 得到创建时间 |
| CString GetModifyTime() | 无 | 修改时间 | 得到修改时间 |
| void SetId(int id) | id计数器的值 | 无 | 给计数器的值 |
| void SetName(CString name) | name表名 | 无 | 给表名赋值 |
| void SetDBName(CString dbName) | dbName数据库名 | 无 | 给数据库名赋值 |
| void SetFieldNum(int fieldNum) | fieldNum字段个数 | 无 | 给字段个数赋值 |
| void SetRecordNum(int recordNum) | recordNum记录个数 | 无 | 给记录个数赋值 |
| void SetCreateTime(CString crtTime) | crtTime创建时间 | 无 | 给创建时间赋值 |
| void SetModifyTime(CString modifyTime) | modifyTime修改时间 | 无 | 给修改时间赋值 |

### 4.1.3字段

1.类图



2.方法说明

（1）构造方法

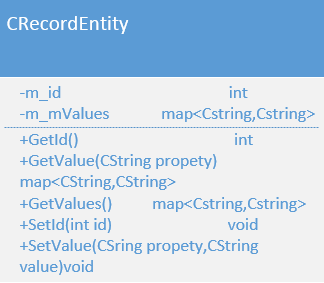
|  |  |  |
| --- | --- | --- |
| **方法名** | **功能** | **说明** |
| void  CFieldEntity(CString name="NewField",int type=DB\_INTEGER,int length=DEFAULT\_FIELD\_LENGTH,int isPk=NO,int isNull=YES,int isUnique=NO) | 构造方法 |  |

（2）其他方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法名** | **输入值** | **返回值** | **说明** |
| int GetId() | 无 | 计数器的值 | 得到计数器的值 |
| CString GetName() | 无 | 字段名称 | 得到字段名称 |
| int GetOrder() | 无 | 字段顺序 | 得到字段顺序 |
| int GetType() | 无 | 字段类型 | 得到字段类型 |
| int GetLength() | 无 | 字段长度 | 得到字段长度 |
| CString GetDBPath() | 无 | 数据库路径 | 得到数据库路径 |
| double GetMax() | 无 | 约束条件最大值 | 得到约束条件最大值 |
| double GetMin() | 无 | 约束条件最小值 | 得到约束条件最小值 |
| CString GetDefault() | 无 | 缺省值 | 得到缺省值 |
| int GetIsPK() | 无 | 宏定义结果 | 是否主键 |
| int GetIsNull() | 无 | 宏定义结果 | 是否为空 |
| int GetIsUnique() | 无 | 宏定义结果 | 是否唯一 |
| CString GetComment() | 无 | 注释 | 得到注释 |
| void SetId(int id) | id计数器的值 | 无 | 给计数器的值赋值 |
| void SetName(CString name) | name字段名称 | 无 | 给字段名称赋值 |
| void SetOrder(int order) | order字段顺序 | 无 | 给字段顺序赋值 |
| void SetType(int type) | type字段类型 | 无 | 给字段类型赋值 |
| void SetLength(int length) | length字段长度 | 无 | 给字段长度赋值 |
| void SetMax(double max) | max约束条件最大值 | 无 | 给约束条件最大值赋值 |
| void SetMin(double min) | min约束条件最小值 | 无 | 给约束条件最小值赋值 |
| void SetDefault(CString dft) | dft缺省值 | 无 | 给缺省值赋值 |
| void SetIsPK(int isPK) | isPK宏定义 | 无 | 确定是否主键 |
| void SetIsNull(int isNull) | isNull宏定义 | 无 | 确定是否为空 |
| void SetIsUnique(int isUnique) | isUnique宏定义 | 无 | 确定是否唯一 |
| void SetComment(CString comment) | comment注释 | 无 | 给注释赋值 |

### 4.1.4记录

1.类图



2.方法说明

（1）构造方法

|  |  |  |
| --- | --- | --- |
| **方法名** | **功能** | **说明** |
| +CRecordEntity(int id=INVALID) | 构造方法 |  |

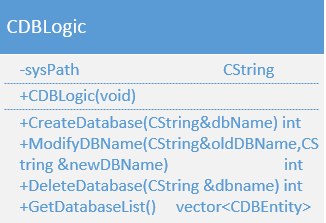
（2）其他方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法名** | **输入值** | **返回值** | **说明** |
| int GetId() | 无 | 计数器的值 | 得到计数器的值 |
| CString GetValue(CString propety) | 属性 | 相应属性的值 | 得到相应属性的值 |
| map<CString,CString> GetValues() | 无 | 所有值 | 得到所有值 |
| void SetId(int id) | id计数器的值 | 无 | 给计数器赋值 |
| void SetValue(CString propety,CString value) | Propety属性，  Value值 | 无 | 给相对应属性的值赋值 |

## 4.2逻辑类设计

### 4.2.1数据库

1.类图



2.方法说明

（1）构造方法

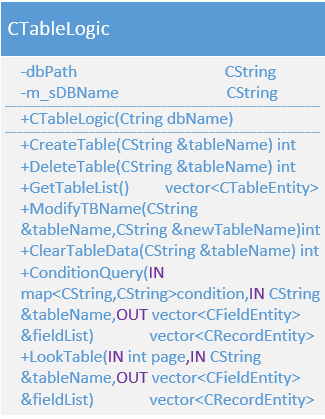
|  |  |  |
| --- | --- | --- |
| **方法名** | **功能** | **说明** |
| void CDBLogic(void) | 构造函数 |  |

（2）其他方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法名** | **输入值** | **返回值** | **说明** |
| int CreateDatabase(CString&dbName) | dbName  数据库名称 | 宏定义结果 | 创建数据库 |
| int ModifyDBName(CString&oldDBName,  CString &newDBName) | oldDBName数据库名称，  newDBName新数据库名称 | 宏定义结果 | 修改数据库名 |
| int DeleteDatabase(CString &dbname) | dbname数据库名称 | 宏定义结果 | 删除数据库 |
| Vector<CDBEntity> GetDatabaseList() |  | 数据库实体向量 | 得到数据库列表 |

### 4.2.2表

1.类图



2.方法说明

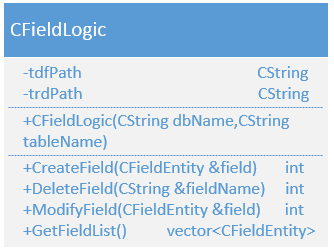
|  |  |  |
| --- | --- | --- |
| **方法名** | **功能** | **说明** |
| +CTableLogic(CString dbName) | 构造方法 |  |

（2）其他方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法名** | **输入值** | **返回值** | **说明** |
| int CreateTable(CString &tableName) | tableName表名称 | 宏定义结果 | 创建表 |
| int  DeleteTable(CString &tableName) | tableName表名称 | 宏定义结果 | 删除表 |
| Vector<CTableEntity> GetTableList() |  | 表实体向量 | 得到表列表 |
| int  ModifyTBName(CString &tableName,CString &newTableName) | tableName表名称，  newTableName新表名称 | 宏定义结果 | 修改表名 |
| int ClearTableData(CString &tableName) | tableName表名称 | 宏定义结果 | 清空表数据 |
| Vector<CRecordEntity>  ConditionQuery(IN map<CString,CString>condition,  IN CString &tableName,  OUT vector<CFieldEntity> &fieldList) | condition Map条件，  tableName表名称，  fieldList字段列表 | 记录实体向量 | 条件查询 |
| vector<CRecordEntity>  LookTable(IN int page,  IN CString%tableName,OUT vector<CFieldEntity>&fieldList) | page页，  tableName表名称，fieldList字段实体向量 | 记录实体向量 | 表查询 |

### 4.2.3 字段

1.类图



2.方法说明

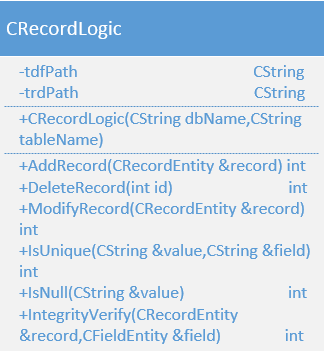
|  |  |  |
| --- | --- | --- |
| **方法名** | **功能** | **说明** |
| +CFieldLogic(CString dbName,CString tableName) | 构造方法 |  |

（2）其他方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法名** | **输入值** | **返回值** | **说明** |
| int CreateField(CFieldEntity &field) | field字段实体 | 宏定义结果 | 创建字段 |
| int DeleteField(CString &fieldName) | fieldName字段名称 | 宏定义结果 | 删除字段 |
| int ModifyField(CFieldEntity &field) | field字段实体 | 宏定义结果 | 修改字段 |
| vector<CFieldEntity>  GetFieldList() |  | 字段实体向量 | 得到字段列表 |

### 4.2.4记录

1.类图



2.方法说明

（1）构造方法

|  |  |  |
| --- | --- | --- |
| **方法名** | **功能** | **说明** |
| +CRecordLogic(CString dbName,CString tableName) | 构造方法 |  |

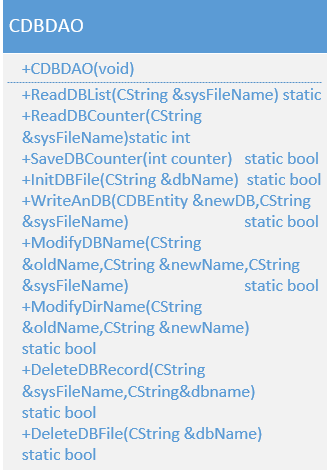
（2）其他方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法名** | **输入值** | **返回值** | **说明** |
| int AddRecord(CRecordEntity &record) | record记录实体 | 宏定义结果 | 增加记录 |
| int DeleteRecord(CRecordEntity &record) | record记录实体 | 宏定义结果 | 删除记录 |
| int ModifyRecord(CRecordEntity &record) | record记录实体 | 宏定义结果 | 修改记录 |
| int IsUnique(CString &value,CString &field) | value值,  field字段实体 | 宏定义结果 | 判断是否唯一 |
| int IsNull(CString &value) | value值 | 宏定义结果 | 判断是否为空 |
| int IntegrityVerify(CRecordEntity &record,CFieldEntity &field) | record记录实体，field实体 | 宏定义结果 | 完整性判断 |

## 4.3业务类类设计

### 4.3.1数据库

1.类图



2.方法说明

（1）构造方法

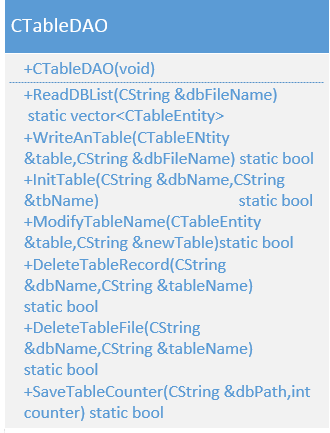
|  |  |  |
| --- | --- | --- |
| **方法名** | **功能** | **说明** |
| +CDBDAO(void) | 构造函数 |  |

（2）其他方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法名** | **输入值** | **返回值** | **说明** |
| static vector<CDBEntity> ReadDBList(CString &sysFileName); | sysFileName系统记录文件名称 | 数据库实体 | 数据库实体列表 |
| static int ReadDBCounter(CString &sysFileName); | sysFileName系统记录文件名称 | 宏定义结果 | 数据库计数器 |
| static bool WriteAnDB(CDBEntity &newDB,CString &sysFileName) | newDB  数据库实体，  sysFileName系统记录文件名 | True/false | 新增数据库到文件 |
| static bool ModifyDBName(CString &oldName,CString &newName,CString &sysFileName); | oldName旧数据库名，  newName新数据库名，  sysFileName系统记录文件名 | True/false | 把新数据库名写入文件 |
| static bool DeleteDBRecord(CString &sysFileName,CString &dbname) | sysFileName系统记录文件，  dbname数据库名 | True/false | 删除文件中数据库记录 |
| static bool DeleteDBFile(CString &dbName) | dbName数据库名 | True/false | 删除该数据库全部文件 |

### 4.3.2表

1.类图



2.方法说明

（1）构造方法

|  |  |  |
| --- | --- | --- |
| **方法名** | **功能** | **说明** |
| +CTableDAO(void); | 构造方法 |  |

（2）其他方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法名** | **输入值** | **返回值** | **说明** |
| static vector<CTableEntity> ReadTableList(CString &dbFileName); | dbFileName数据库文件名称 | 表实体 | 表实体列表 |
| static bool WriteAnTable(CTableEntity &table,CString &dbFileName) | table表实体，  dbFileName数据库文件名称 | True/false | 新建一张表到db文件 |
| static bool ModifyTableName(CTableEntity &table,CString &newTableName) | table表实体，  newTableName新表名 | True/false | 把新的表名写入数据库 |
| static  bool DeleteTableRecord(CString &dbName,CString &tableName) | dbName系统记录文件，  tableName表名 | True/false | 删除文件中数据库记录 |

### 4.3.3 字段

1.类图



2.方法说明

（1）构造方法

|  |  |  |
| --- | --- | --- |
| **方法名** | **功能** | **说明** |
| +CFieldLogic(CString dbName,CString tableName) | 构造方法 |  |

（2）其他方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法名** | **输入值** | **返回值** | **说明** |
| static vector<CFieldEntity> ReadFieldList(CString &tbFileName) | tbFileName字段文件名 | 字段实体 | 从文件读取字段列表 |
| static bool WriteAnField(CFieldEntity &newField,CString &tdfFileName) | newField新字段  tdfFileName指定字段文件名 | True/false | 文件增加一个字段记录 |
| static bool DeleteField(CString &fieldName,CString &tdfFileName) | fieldName文件名，  tdfFileName指定字段文件名 | True/false | 从文件删除一条字段信息 |
| static bool ModifyField(CFieldEntity &field,CString &tdfFileName) | Field字段实体  tdfFileName指定字段文件名 | True/false | 把修改后的字段写入文件 |
| static int GetFieldCounter(CString &fieldFileName) | fieldFileName字段文件名 | 宏定义结果 | 读取数据库的计数器 |
| static bool SaveFieldCounter(CString &fieldFileName,int counter) | fieldFileName字段文件名  counter计数器 | True/false | 保存计数器 |

### 4.3.4记录

1.类图



2.方法说明

（1）构造方法

|  |  |  |
| --- | --- | --- |
| **方法名** | **功能** | **说明** |
| +CRecordDAO(void) | 构造方法 |  |

（2）其他方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法名** | **输入值** | **返回值** | **说明** |
| static bool WriteAnRecord(CRecordEntity &record,CString &trdFileName) | Record记录实体，  trdFileName记录文件名 | True/false | 从文件新增一条记录,写入文件 |
| static vector<CRecordEntity> ReadRecordList(int page,int pageRcdNum,CString &trdFileName) | Page页号，  pageRcdNum每页数，  trdFileName记录文件名 | 记录实体 | 读取指定页的记录数 |
| static vector<CRecordEntity> ReadRecordList(CString &trdFileName) | trdFileName记录文件名 | 记录实体 | 获取全部记录 |
| static vector<CRecordEntity> ReadListWithCondition(CString &trdFileName,map<CString,CString> condition) | rdFileName记录文件名  condition条件 | 记录实体 | 按照查询条件读取指定页的记录数 |
| static bool ModifyRecord(CRecordEntity &record,CString &trdFileName) | Record记录实体，  trdFileName记录文件名 | True/false | 从文件修改指定的记录 |
| static bool DeleteRecord(int id,CString &trdFileName) | Id序号  trdFileName记录文件名 | True/false | 从文件删除一条记录 |
| static bool DeleteAllRecord(CString &trdFileName) | trdFileName记录文件名 | True/false | 从文件删除所有记录 |
| static bool DeleteFieldRecord(CString &trdFileName,int index) | trdFileName记录文件名  index索引 | True/false | 按照字段名称删除关于该字段的 |

## 4.4工具类类设计

### 4.4.1文件操作

1.类图



2.方法说明

（1）构造方法

|  |  |  |
| --- | --- | --- |
| **方法名** | **功能** | **说明** |
| Void CBinaryFile(void) | 构造函数 |  |

（2）其他方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法名** | 输入值 | **返回值** | **说明** |
| static vector<CString> ReadAll(CString &fileName) | fileName文件名 | 文件内的全部数据 | 分条读取全部文件的内容 |
| static vector<CString> Read(CString &fileName,int pageNo,int pageRcdNo) | fileName文件名  pageNo页数  pageRcdNo每页的数据条数 | 输入页的所有数据 | 读取指定页的文件的内容 |
| static CString ReadFirstLine(CString &fileName) | fileName文件名 | 返回文件的第一条数据 | 读取第一条文件的内容 |
| static bool Write(CString &fileName,vector<CString> &str) | fileName文件名  str数据 | 写入成功与否 | 将多条记录一次性写入文件 |
| static bool AddAnLine(CString &fileName,CString &str) | fileName文件名  str 数据 | 添加成功与否 | 在文件末尾加入一条记录 |
| static bool DeleteRecordByName(CString &fileName,CString &name) | fileName文件名  name 数据名 | 删除成功与否 | 根据名字删除数据 |
| static bool DeleteRecordById(CString &fileName,int id) | fileName文件名  id 数据ID | 删除成功与否 | 根据ID删除数据 |

### 4.4.2工具

1.类图



2.方法说明

（1）构造方法

|  |  |  |
| --- | --- | --- |
| **方法名** | **功能** | **说明** |
| Void CUtil(void) | **构造函数** |  |

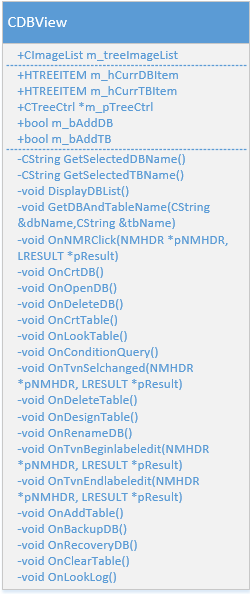
（2）其他方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法名** | **输入值** | **返回值** | **说明** |
| static vector<CString> StrSplit(CString str,CString split) | Str字符串  Split分隔符 | 返回所有的分割值 | 字符串分割函数 |
| static CString GetCurrTime() |  | 返回当前时间 | 得到当前时间 |
| static CString GetError(int code) | Code错误码 | 返回错误类型 | 通过错误码得到错误信息 |
| static CString GetIcon(int flag) | Flag标志 | 返回标识 | 根据标志得到是“√”还是“” |
| static CString GetDataType(int type) | Type类型代码 | 返回字面类型 | 得到数据类型代码对应的字面类型 |
| static CString CharArrToString(char \*charArr) | CharArr char\* | 返回char\*转换成的CString结果 | char\*转化成CString |
| static char \*StringToCharArr(CString cstring) | Cstring字符串 | 返回转换成的char\* | CString转化成char\* |
| static CString IntegerToString(int number) | Number 输入值 | 返回int转换成的字符串 | int转化成字符串 |
| static int StringToInteger(CString cstring) | Cstring字符串 | 返回字符串转化成的数字 | 字符串转化成int |
| static int StringToBool(CString cstring) | Cstring字符串 | 返回字符串转换成的布尔型 | 字符串转换成bool |
| static CString BoolToString(int b) | b数字 | 返回数字转换成的字符串 | bool转换为string |
| static double StringToDouble(CString cstring) | Cstring字符串 | 返回字符串转换成的double | 字符串转换成double |
| static CString DoubleToString(double doubleNum) | doubleNum数字 | 返回数字转换成的字符串 | double转换成字符串 |
| static bool CreateFolder(CString &folderName) | Cstring字符串 | 创建是否成功 | 创建文件夹 |
| static bool ModifyFolderName(CString &oldname,CString &newname) | Oldname文件夹的现有名字  Newname文件夹的新名字 | 修改是否成功 | 修改文件夹名称 |
| static bool ModifyFileName(CString &oldname,CString &newname) | Oldname文件的现有名字  Newname文件的新名字 | 修改是否成功 | 修改文件名 |
| static bool DeleteFolder(CString &folderName) | folderName文件夹名字 | 删除是否成功 | 删除文件夹 |
| static bool DeleteFileA(CString &filename) | Filename文件名 | 删除是否成功 | 删除文件 |
| static bool OnlyHaveNumber(CString &str) | Str字符串 | 是否只含有数字（是或否） | 字符串是否只含有数字 |
| static bool OnlyHaveNumberAndDot(CString &str) | Str字符串 | 是否只含有数字和"."（是或否） | 字符串是否只含有数字和"." |

## 4.5视图层类设计

### 4.5.1 Tree

1.类图

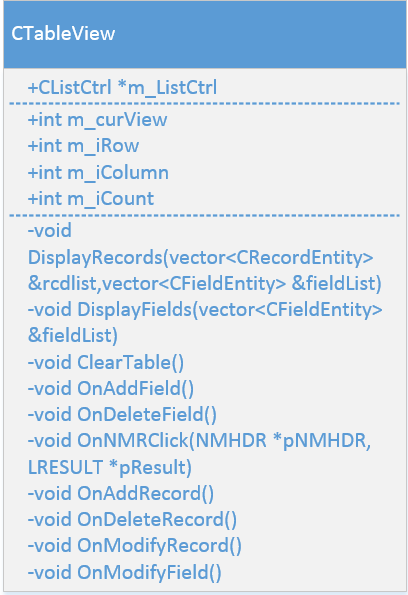


2.方法说明

|  |  |  |  |
| --- | --- | --- | --- |
| **方法名** | **输入值** | **返回值** | **说明** |
| CString GetSelectedDBName() | 无 | 数据库名 | 得到选中数据库名 |
| CString GetSelectedTBName() | 无 | 表名 | 得到选中的表名 |
| void DisplayDBList() | 无 | 无 | 显示数据库列表 |
| void GetDBAndTableName(CString &dbName,CString &tbName) | dbName数据库名，  tbName表名 | 无 | 得到表名和数据库名 |
| void OnCrtDB() | 无 | 无 | 创建数据库 |
| void OnOpenDB() | 无 | 无 | 打开数据库 |
| void OnDeleteDB() | 无 | 无 | 删除数据库 |
| void OnCrtTable() | 无 | 无 | 打开表 |
| void OnLookTable() | 无 | 无 | 查看表 |
| void OnConditionQuery() | 无 | 无 | 条件查询 |
| void OnDeleteTable() | 无 | 无 | 删除表 |
| void OnDesignTable() | 无 | 无 | 设计表 |
| void OnRenameDB() | 无 | 无 | 重命名表 |
| void OnAddTable() | 无 | 无 | 增加表 |
| void OnBackupDB() | 无 | 无 | 备份数据库 |
| void OnRecoveryDB() | 无 | 无 | 恢复数据库 |
| void OnClearTable() | 无 | 无 | 清空表 |
| void OnLookLog() | 无 | 无 | 查看日志 |

### 4.5.2 Table

1.类图



2.方法说明

|  |  |  |  |
| --- | --- | --- | --- |
| **方法名** | **输入值** | **返回值** | **说明** |
| void DisplayFields(vector<CFieldEntity> &fieldList) | 字段实体列表 | 无 | 显示字段 |
| void ClearTable() | 无 | 无 | 清空表 |
| void OnAddField() | 无 | 无 | 增加字段 |
| void OnDeleteField() | 无 | 无 | 删除字段 |
| void OnAddRecord() | 无 | 无 | 增加记录 |
| void OnDeleteRecord() | 无 | 无 | 删除记录 |
| void OnModifyRecord() | 无 | 无 | 修改记录 |
| void OnModifyField() | 无 | 无 | 修改字段 |