# 引言

## 编写目的与基线

*【--说明编写详细设计方案的主要目的。*

*说明书编制的目的是说明一个软件系统各个层次中的每个程序（每个模块或子程序）和数据库系统的设计考虑，为程序员编码提供依据。*

*如果一个软件系统比较简单，层次很少，本文件可以不单独编写，和概要设计说明书中不重复部分合并编写。*

*方案重点是模块的执行流程和数据库系统详细设计的描述。--】*

***该文档针对XX设计的XX模块的XX版本进行编写的***

## 参考资料

*【--列出详细设计报告引用的文献或资料，资料的作者、标题、出版单位和出版日期等信息，必要时说明如何得到这些资料。--】*

## 术语定义及说明

*【--列出本文档中用到的可能会引起混淆的专门术语、定义和缩写词的原文。--】*

# 需求及设计决策

## 输入项

### 输入项的需求

### 输入项的决策

## 输出项

### 输出项的需求

### 输出项的决策

## 约束条件

安全性、保密性、私密性的需求及选择的方法

# 配置项详细设计

## 配置项体系结构

*【--对系统组成、逻辑结构及层次进行确认，对应用系统、支撑系统及各自实现的功能进行确认，细化集成设计及系统工作流程，特别要注意因软件的引进造成的系统本身结构和公司其他系统的结构变化。包括：*

* *系统组成、逻辑结构及层次确认*
* *应用系统结构确认*
* *支撑系统结构确认*
* *系统集成确认*

*系统工作流程确认--】*

### 单元划分

### 层级结构

## 全局数据结构

### 常量

### 变量

### 数据结构

## 接口设计

### 接口标识与接口图

### XX接口

#### 接口的优先级别

#### 接口类型

#### 元素及集合特征

#### 通讯方法

#### 协议的特征

#### 兼容性

### XX接口

## 单元详细设计

*【与3.1.1一一对应】*

### XX单元

#### 限制条件以及非常规特性

#### 软件逻辑

#### 启动时内部起的作用

#### 控制交予其他配置项的条件

#### 对每个输入的响应及响应时间，包括数据转换、重命名、数据传送操作

#### 操作序列和动态控制序列

#### 异常与错误

### XX单元

### XX单元

### XX单元

# 需求可追溯性

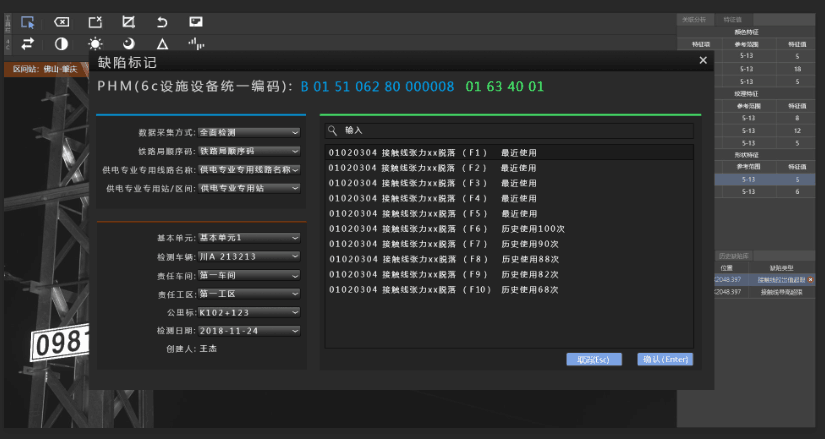
# 注解

*【--本章应包含有助于理解本文档的一般信息（例如原理本章应包含为理解本文档需要的术语和定义，所有缩略语和它们在文档中的含义的字母序列表--】*

# 外包要求

## 样本模型及缺陷管理组件

### 组件示意图



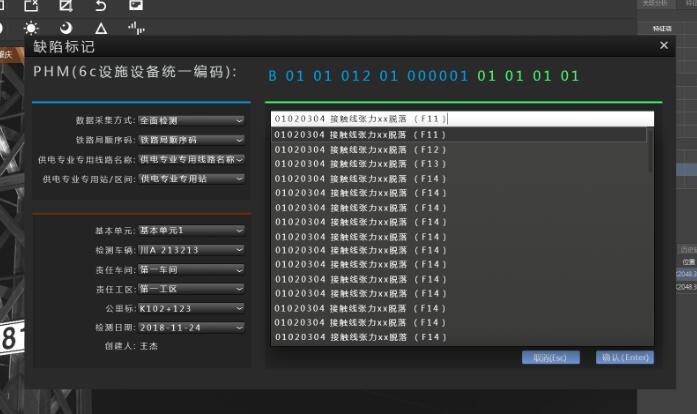


图 1 标记原型

### 组件功能描述

1. 组件初始化前设置缺陷标记、样本标记中要展示的项(加载各下拉框数据)
2. 加载最近使用与历史使用次数最多缺陷或者样本
3. 组件初始化时可设置默认光标位置。
4. 组件初始化后，图1中6C设施设备统一编码表的数据源只设置一次；图2中第一行数据源只设置一次，提供reset方法清空所有设置。
5. 样本与缺陷检索时支持编码检索、名称检索与拼音检索，检索后的结果面板覆盖最近使用面板
6. 最近使用面板与搜索结果中的项都要支持快捷键选中
7. 下图中所有项：

* 支持类型设置text、combox、lable三种
* text、combox支持ReadOnly设置
* 支持tab键切换
* 支持快捷键定位
* 下次弹出默认显示最后一次编辑信息
* 所有下拉列表支持快速检索功能，检索结果为单一结果时，以该结果填充单元格，并自动跳转到下一单元格（不可跳至ReadOnly单元格，手动选ReadOnly除外）
* 各项可以设置过滤项，过滤项为最多2项，过滤项之间的关系为and，如设置设备参数项目的过滤字段1的过滤参数为设备设施种类，那么，那么设备参数项目应显示 过滤字段1值等于设备设施种类.value的项
* 列表最后一项不可修改

1. 提供保存事件，将弹框信息作为参数传出，提供各选择项模型，与设备统一编码值（最终组合值，缺陷是24位编码，样本为23位）

### 组件接口方法描述

/// <summary>

/// 缺陷标记右键菜单对象

/// </summary>

public class DefectMenuEntity

{

/// <summary>

/// PHM编码（基本由界面属性组合而成）

/// </summary>

public string PHMCodes { set; get; }

/// <summary>

/// 缺陷标记信息

/// </summary>

public TyeDefectImgmarkinfoEntity DefectMark { get; set; }

/// <summary>

/// 数据采集方式

/// </summary>

public List<TyeEncodeCategoryconfigEntity> DataAcquisitionMode { get; set; }

/// <summary>

/// 铁路局顺序码

/// <summary>

public List<TyeEncodeCategoryconfigEntity> RailwaySsequence { get; set; }

/// <summary>

/// 供电专业专用线路名称

/// <summary>

public List<TyeBaseLineEntity> DedicatedLine { get; set; }

/// <summary>

/// 供电专业专用站\区间

/// <summary>

public List<TyeBaseSiteEntity> DedicatedStation { get; set; }

/// <summary>

/// 基本单元

/// <summary>

public List<TyeEncodeCategoryconfigEntity> BasicUnit { get; set; }

/// <summary>

/// 检测车辆

/// <summary>

public string DetectionVehicles { get; set; }

/// <summary>

/// 责任车间

/// <summary>

public List<TyeEncodeCategoryconfigEntity> ResponsibilityWorkshop { get; set; }

/// <summary>

/// 责任工区

/// <summary>

public List<TyeEncodeCategoryconfigEntity> ResponsibilityWorkArea { get; set; }

/// <summary>

/// 公里标

/// <summary>

public string KmLog { get; set; }

/// <summary>

/// 检测日期

/// <summary>

public DateTime? DetectDate { get; set; }

/// <summary>

/// 缺陷或者标定代码

/// <summary>

public ObservableCollection<TyeEncodeDeviceEntity> DefectOrMarkCodes { get; set; }

/// <summary>

/// 当前用户

/// </summary>

public TyeAdminUserEntity tyeAdminUserEntity { set; get; }

/// <summary>

/// 最近使用缺陷、标定；历史使用缺陷、样本标定次数top number

/// </summary>

public List<DefectCommonUsed> CommonHistoricalDefectsOrMark { set; get; }

### }

### ///事件确认数据后返回PHM编码

### public Action<string> ConfirmData;