

C语言数据文件智能处理平台

产品解决方案

一、项目背景

在铁路车站信息管理等工业控制领域，大量关键数据以C语言源文件的形式存储和管理。这些文件包含结构体定义、数据变量等，是系统运行的核心配置。

当前面临的挑战：

痛点	影响
数据检查靠人工	效率低、易出错、规则难以统一
查找数据困难	需要逐行阅读代码，耗时耗力
文件比对不智能	普通比对工具无法识别“等价但写法不同”的情况
修改数据风险高	手动编辑C代码容易引入语法错误

二、解决方案概述

我们提供一套智能化的C语言数据文件处理平台，让业务人员无需编程知识，即可完成数据检查、查询、比对和修改工作。

核心理念

传统方式：业务人员 → 提需求 → 开发人员 → 写代码 → 结果
本平台： 业务人员 → 低代码配置 → 立即得到结果

三、核心功能

3.1 智能数据检查

功能说明：自动检查C语言文件中的数据是否符合业务规则，并以类似IDE的方式直观显示问题位置。

使用场景示例：

需求：检查“园区的猫”数据中，ID不能重复，出生日期必须晚于1990年

传统方式：需要开发人员编写检查程序，周期长、维护难

本平台方式：业务人员直接配置规则

规则1：园区的猫们 ID 不重复

规则2：园区的猫们 出生日期 > 1990

检查结果展示：



△ 警告 [第15行]：ID “CAT001” 重复出现，首次出现在第8行

X 错误 [第23行]：出生日期 “1985” 不符合规则（必须 > 1990）

✓ 第1-50行检查完成，发现 1 个错误，1 个警告

核心优势：

- ✓ 规则配置简单，类似写Excel公式
- ✓ 支持规则的版本管理和发布审批
- ✓ 检查结果一键定位到源文件位置
- ✓ 支持批量检查多个文件

3.2 数据快速查询

功能说明：用简单的查询语句，快速从C语言文件中找到需要的数据。

使用场景示例：

需求：查找所有狸花猫的信息

查询方式：

查询 园区的猫们 条件 品种 = 狸花

查询结果：

ID	名字	品种	出生日期	颜色
CAT002	小黑	狸花	2024	灰色
CAT005	花花	狸花	2023	花色

支持的查询能力：

- 条件筛选（等于、大于、小于、包含等）
- 多条件组合（AND、OR）
- 结果排序
- 统计汇总（数量、求和、平均值等）
- 结果导出（Excel、CSV）

3.3 智能文件比对

功能说明：比较两个C语言文件的差异，**自动过滤无意义差异**，只显示真正的数据变化。

传统比对工具的问题：

```
// 文件A                                // 文件B
猫们 = {小黄, 小黑, 小白};          猫们 = {小黑, 小白, 小黄};
数值 = 0x01;                          数值 = 0x1;
```

普通比对工具会报告“2处差异”，但实际上：

- 顺序不同 {小黄, 小黑, 小白} 和 {小黑, 小白, 小黄} **业务上等价**
- 0x01 和 0x1 对程序来说**完全相同**

本平台的智能比对：

比对级别	说明	可配置
文本比对	逐字符比较	-
语法比对	忽略空格、注释、数值格式差异	<input checked="" type="checkbox"/>
业务比对	忽略顺序差异、识别等价写法	<input checked="" type="checkbox"/>

比对结果展示：

比对报告

实质性变化：2 处

└─ [第12行] 新增: 小花 (奶牛, 2025, 黑白)

└─ [第8行] 修改: 小黄 品种 橘猫 → 金渐层

可忽略差异：3 处 (已自动过滤)

└─ 顺序调整: 2 处

└─ 格式差异: 1 处

3.4 数据安全修改

功能说明：通过简单指令修改数据，系统自动生成正确的C代码，避免手动编辑出错。

使用场景示例：

需求：将“小黄”的品种改为“金渐层”，新增一只猫“小花”

操作方式：

修改 小黄 品种 = 金渐层
插入 园区的猫们 (小花, 奶牛, 2025, 黑白)

系统自动完成:

- 验证操作的合法性
- 生成正确的C语言代码
- 显示修改前后对比
- 支持撤销操作

修改预览:

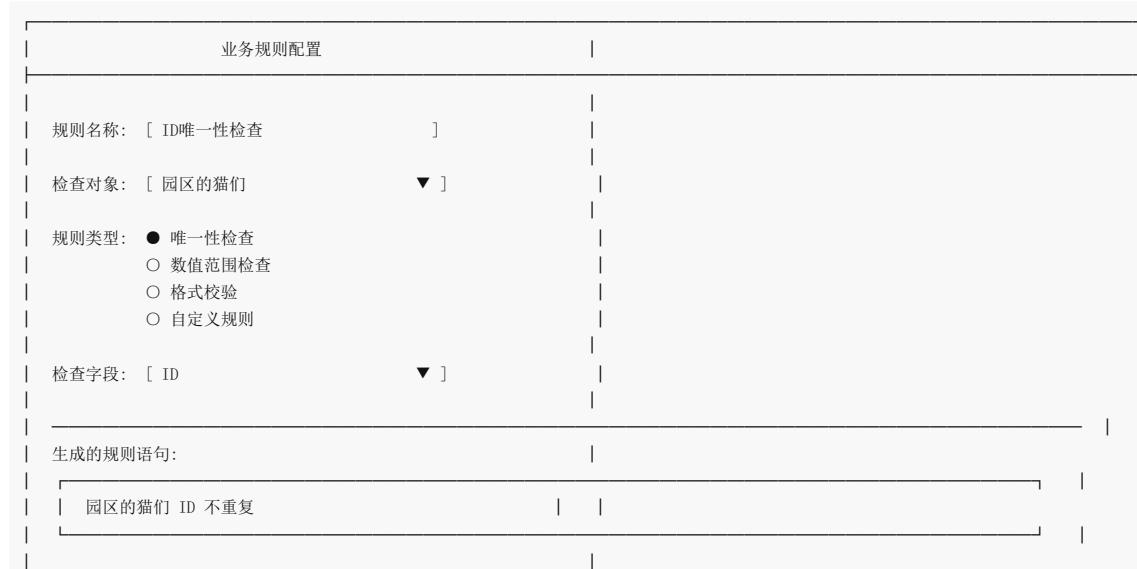
```
猫 园区的猫们[] = {
- {小黄, 橘猫, 2023, 黄色},
+ {小黄, 金渐层, 2023, 黄色},
  {小黑, 狸花, 2024, 灰色},
+ {小花, 奶牛, 2025, 黑白},
};
```

四、产品界面预览

4.1 主工作界面



4.2 规则配置界面



[测试规则] [保存] [取消]

五、产品优势

5.1 对比传统方式

维度	传统方式	本平台
数据检查	人工逐行审查或编写程序	配置规则，自动检查
检查效率	数小时至数天	秒级完成
规则维护	需要开发人员修改代码	业务人员自助配置
查询数据	阅读源码或写程序	简单查询语句
文件比对	普通diff工具，噪音多	智能过滤，只看重点
修改数据	手动编辑，易出错	安全指令，自动生成

5.2 核心价值

- 💡 降低成本：减少对开发人员的依赖，业务自助完成
- ⚡ 提升效率：检查效率提升 10倍+，查询秒级响应
- 🛡️ 降低风险：规则统一管理，避免人工疏漏
- 🌐 易于扩展：规则可复用，支持多项目统一管理

六、技术保障

6.1 技术架构

本平台采用业界主流的成熟技术构建，确保系统稳定可靠：

层级	技术选择	说明
前端界面	Vue 3	主流前端框架，响应迅速
代码编辑器	Monaco Editor	VS Code同款编辑器内核
后端服务	Spring Boot	企业级Java框架
语法解析	ANTLR4	业界标准解析工具
规则引擎	Drools	企业级规则引擎

七、实施方案

7.1 实施步骤

