

## 《C/C++学习指南》 实战篇

### 003 学生信息管理(基于SQLite实现)

作者 邵发 官网 <http://afanihao.cn>

## 《实战篇》的教学目标

写应用程序就如同堆积木，本篇的重点是展示堆积木的思路... (代码规模并不重要)



- **原材料** 语法，线程，SOCKET，SQL，加密解密，标准C库，STL库。。。
- **积木** 模块：类/函数
- **堆积木** 应用程序：将各模块联合起来

## 本节课的背景知识

《语法篇》：文件操作，STL

《Qt篇》：树控件，对话框，自定义Widget

《SQLite篇》：数据库基本操作，BLOB读写

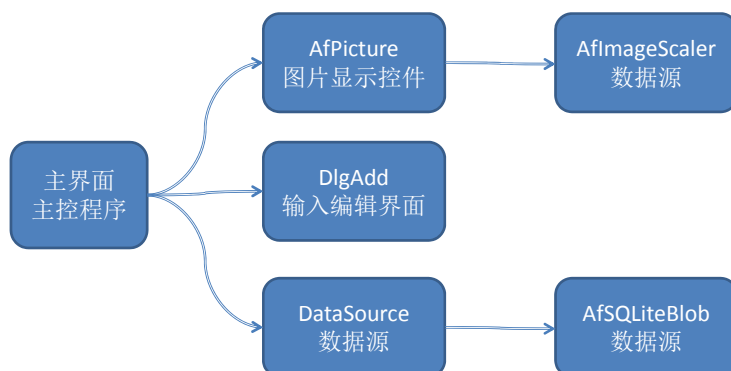
建议先看Demo002，功能类似

## 需求 (要完成的目标功能)

实现一个“学生信息管理”软件

- (1) 增加
- (2) 删除
- (3) 列表显示
- (4) 查询（按名称匹配）
- (5) 数据存盘（永久化存储）
- (6) 设置与显示头像 (一寸照片, JPG)

## 设计 架构



## 设计架构

demo003: 主控界面

DataSource: 负责数据的存取，本例中使用SQLite实现

AfSQLiteBlob: 可重用类。负责将一个文件存取BLOB:

AfPicture: 可重用类。负责显示一个图片。

AfImageScaler:可重用类。图片按比例计算器。

## 可重用

优秀代码的三条标准：

可读，可扩展，可重用

可重用，是指一个类/函数/模块，可以适用于其他项目（无需改动）

可重用是一个很高的标准。

思考：你写的类别人能直接用吗？

## 可重用

如何达到可重用的标准？

1. 良好的封装

2. 单向调用，被动模式

除非必要，均使用被动模式

3. 使用标准的，尽量不要用平台相关的

例如，我们应该使用`int`, `int*`，不要使用`INT`, `PINT`。。。使用`fopen`不使用`fopen_s`。。。

4. 去除不必要的头文件依赖

## 可重用

思考：为什么AflImageScaler内部要定义Rect类？为什么不直接使用QRect类？

## 技术点

技术细节不是本篇的重点。

### 1. 图片的比例缩放

当图片的比例，与外框的比例不一致时，要自己计算。

### 2. BLOB的存取

### 3. 右键菜单的响应

### 4. 如何判断BLOB列是否为NULL

## 小结

重点：理解“可重用性”

我们所使用的API、类库都是可重用的。思考和比较，为什么他们写的类库我们可以自由使用，而我们写的类别人却难以使用。。。