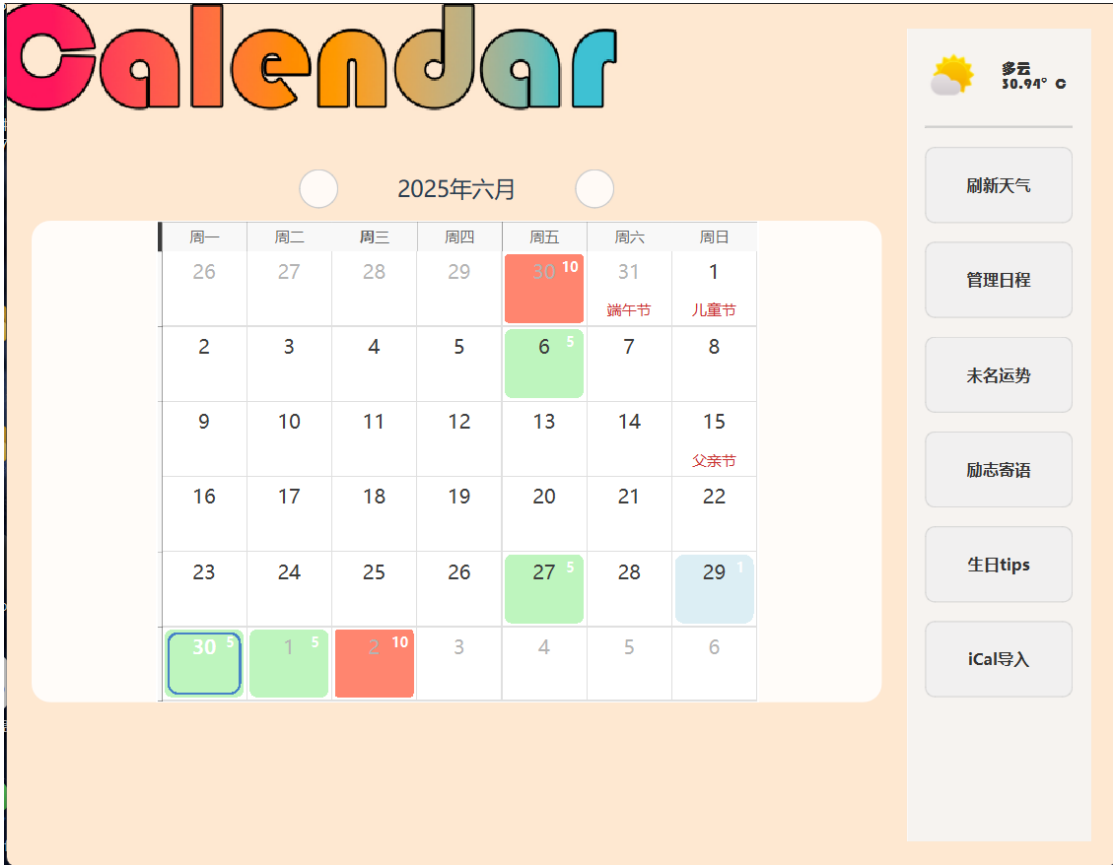


# 47 组程设大作业开发报告

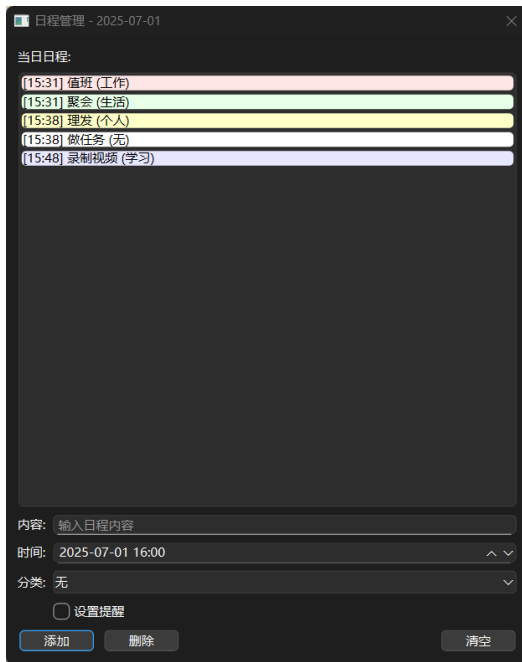
## 一、项目概述

这是一个基于 Qt 框架开发的跨平台桌面应用，整合了日历系统、天气查询、日程规划、运势生成、励志寄语等功能的综合性工具。为北京大学师生设计，具有独特的校园元素和学术风格。



## 二、程序功能介绍

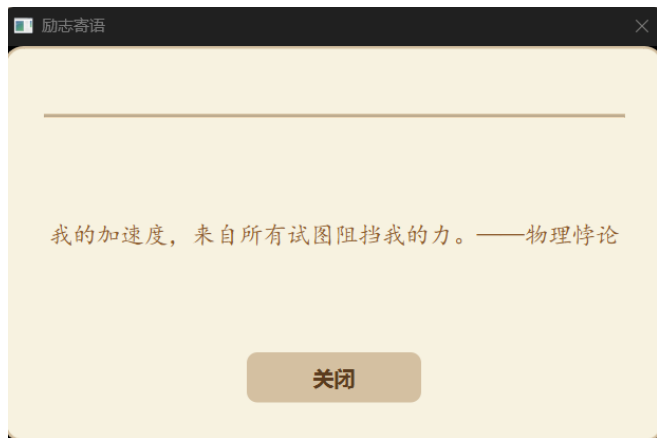
- 1、日历系统：排版上采用经典的日历视图模式，通过按键实现月份的转换。该日历能够实现日期与星期的正确对应，以及各种节假日的查询。自制的界面 UI 美观和谐，风格温馨。
- 2、天气查询：通过 OpenWeatherMap API 获取天气数据，并通过“刷新天气”按钮获取实时天气情况，天气图标与字体美观可爱，制作精美。
- 3、日程管理：在选中具体日期后，通过“日程管理”按钮可以进行日程的编辑，对日程进行添加，删除或清空。对于编辑的日程，可以选择其具体时间，分类情况，是否提醒。不同的分类会用不同的底色展示，选择提醒后会加粗显



示。日程编辑好后，在日历中会有日程热力图显示，根据日程数量将该日期涂上不同底色，并在右上角显示当日日程的具体数目，使日程安排更加一目了然。

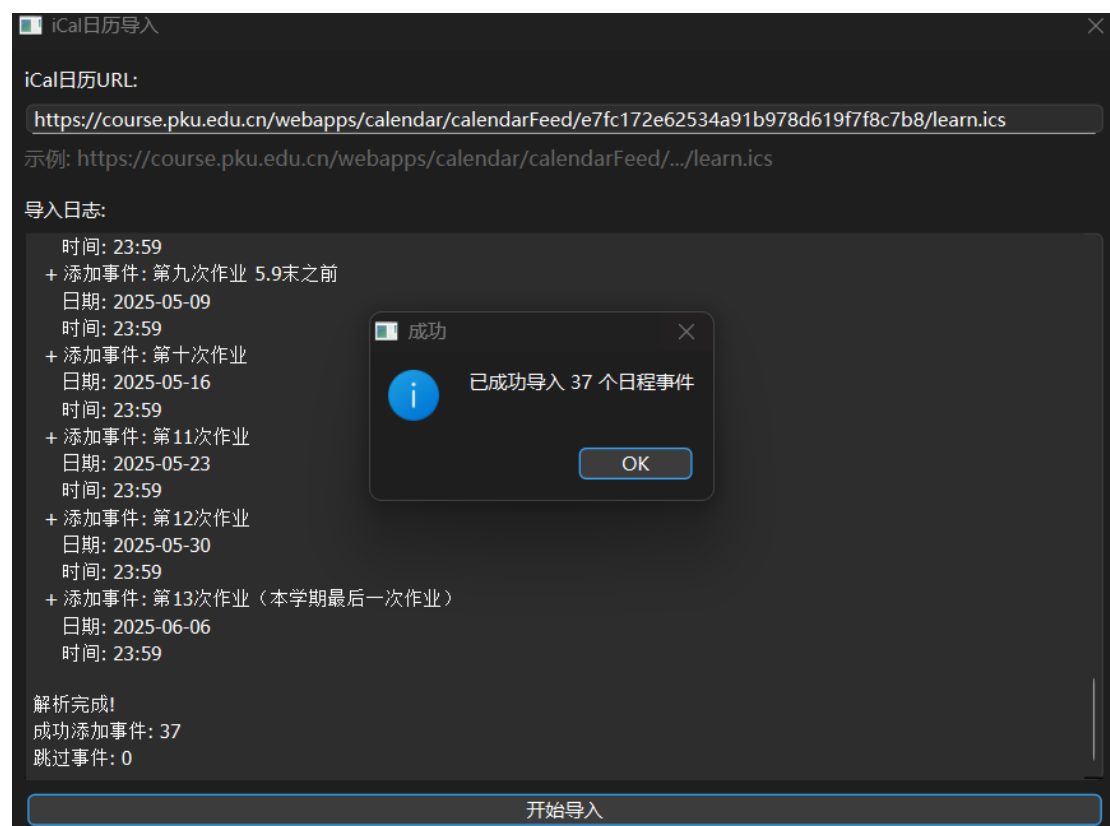
4、未名运势：该功能专为北大师生设计，具有独特的校园元素。用户通过“未名运势”按钮，使用 DeepSeek API 生成北京大学专属运势，解锁元气满满的一天。

5、励志寄语：我们搜寻了许多小众但又励志的话，整合成了程序内置的励志寄语语料库。用户通过“励志寄语”按钮，将会随机解锁励志寄语中的一条，来激励自己。



6、生日管理：生活中，我们常常想要在别人生日时献上我们真挚的问候。在这一功能中，我们可以记录下“TA”的生日，并为“TA”进行需要的备注设置。系统将会自动为“TA”的生日倒计时，不用再担心会忘记重要的人的生日啦！

7、iCal 导入：你是否因为忘记教学网上作业的 ddl 而尴尬地逾期提交。在该功能中，用户可以输入 iCal 日历 URL，程序将下载并解析 iCal 数据，自动提取事件并添加到日程中。这样就不用再担心忘记作业的 ddl 了。



### 三、项目各模块与类设计细节

#### (一)、日历显示模块

##### 1. CalendarDelegate (日历单元格绘制)

- 职责：负责日历单元格的视觉渲染
- 核心功能：热力图效果：根据日程数量显示蓝-黄-红渐变背景；日期标注：公历（顶部）、农历/节日（底部）；特殊标识：当前日期蓝色边框、日程数量角标

```
// 热力图颜色计算
QColor heatMapColor(float value, float maxValue) const {
    float ratio = qBound(0.0f, value/maxValue, 1.0f);
    if (ratio < 0.33f) return QColor(173,216,230,150); // 浅蓝
    else if (ratio < 0.66f) return QColor(144,238,144,180); // 浅绿
    else return QColor(255,99,71,210); // 番茄红
}

// 当日高亮效果
if (isToday) {
    painter->setPen(QPen(QColor(70,130,200),2));
    painter->drawRoundedRect(contentRect,8,8);
}
```

## 2. CalendarModel (日历数据模型)

- 职责：提供日历的底层数据逻辑
- 数据结构：

```
enum CustomRoles {  
    DateRole = Qt::UserRole + 1,    // 公历日期  
    LunarRole,                      // 农历信息  
    HolidayRole,                    // 节日信息  
    ScheduleCountRole               // 日程数量  
};
```

- 核心算法：

```
// 日期计算逻辑  
QDate dateFromIndex(const QModelIndex &index) const {  
    QDate firstDay = m_currentDate.addDays(1 - m_currentDate.day());  
    int weekDay = firstDay.dayOfWeek();  
    int offset = index.row()*7 + index.column() - weekDay + 1;  
    return firstDay.addDays(offset);  
}
```

## 3. CalendarView (日历视图)

- 职责：整合模型和委托的显示组件
- 特性：6×7 网格布局；自适应大小调整；日期选择信号传递

```
connect(m_calendarView, &QTableView::clicked, [this](const QModelIndex &index){  
    m_selectedDate = index.data(CalendarModel::DateRole).toDate();  
});
```

## 4. HolidayManager (节假日管理)

- 职责：从 JSON 文件加载节假日数据；根据日期获取节假日名称

### (二)、天气模块

#### 1. 安全机制

- API 密钥加密：

```
// 异或加密存储  
for(int i=0; i<toEncrypt.size(); ++i) {  
    toEncrypt[i] = toEncrypt[i] ^ ENCRYPT_KEY.at(i % ENCRYPT_KEY.size()).toLatin1();  
}  
settings.setValue("apiKey", toEncrypt.toBase64());
```

#### 2. 数据获取

```
void requestWeather(const QString &city) {
    QUrl url("http://api.openweathermap.org/data/2.5/weather");
    QUrlQuery query;
    query.addQueryItem("q", city);
    query.addQueryItem("appid", m_apiKey); // 加密存储的API密钥
    url.setQuery(query);
    m_weatherManager->get(QNetworkRequest(url));
}
```

### 3. 数据处理

```
if (root.contains("weather")) {
    QString desc = root["weather"].toArray()[0].toObject()["description"].toString();
    double temp = root["main"].toObject()["temp"].toDouble();
    // 匹配天气图标
    if (desc == "阴") iconPath = ":/images/yin.png";
    else if (desc == "晴") iconPath = ":/images/sunny.png";
}
```

## (三)、日程管理模块

### 1. ScheduleItem (日程条目)

- 数据结构:

```
struct ScheduleItem {
    enum Category { Work, Life, Study, Personal };
    QString title;
    QDateTime time;
    bool reminder;
    Category category;

    // 分类颜色映射
    static QColor categoryColor(Category cat) {
        static const QMap<Category, QColor> colors = {
            {Work, QColor(255,230,230)}, // 浅红
            {Life, QColor(230,255,230)} // 浅绿
        };
        return colors.value(cat, Qt::white);
    }
};
```

### 2. ScheduleDialog (日程管理对话框)

- 功能亮点: 分类颜色标识; 时间校验; 支持批量操作

```
// 添加日程校验
if (item.time() < QDateTime::currentDateTime()) {
    QMessageBox::warning(this, "警告", "不能添加过去时间的日程");
    return;
}
```

### 3. 数据持久化

- 文件格式: .txt 文件
- 存储位置: 文档目录/WeatherCalendarSchedules.txt

#### (四)、特色功能模块

##### 1. 未名运势 (Fortune)

- 技术实现:

```
void showFortune() {
    // 构建DeepSeek API请求
    QString prompt = "作为北京大学专属运势分析专家，请生成运势报告...";
    QJsonObject requestBody;
    requestBody["model"] = "deepseek-chat";
    requestBody["messages"] = QJsonArray({
        QJsonObject{{"role", "system"}, {"content", "运势专家"}},
        QJsonObject{{"role", "user"}, {"content", prompt}}
    });

    // 处理gzip压缩响应
    QByteArray uncompressed = gzipDecompress(reply->readAll());
}
```

- 输出格式:

【未名运势】博雅塔顶见朝阳  
 ✨ 幸运颜色: 北大红  
 🍀 幸运数字: 7  
 ✅ 宜: 复习高数、图书馆自习  
 ❌ 忌: 拖延作业

##### 2. 励志寄语

- 设计特点: 23 条哲理语句库; 随机展示机制; 精美对话框设计

```
void showInspiration() {
    int index = QRandomGenerator::global()->bounded(m_inspirationQuotes.size());
    InspirationDialog dialog(m_inspirationQuotes[index], this);
    dialog.exec();
}
```

##### 3. iCal 导入

- 技术亮点: 支持公开 URL 导入; VEVENT 解析器

```
// iCal事件解析
QRegularExpression eventRegex("BEGIN:VEVENT(.+?)END:VEVENT",
    QRegularExpression::DotMatchesEverythingOption);
```

##### 4. 生日管理

- 数据结构:

```
struct BirthdayEntry {
    QString name;
    QDate birthday;
    QString note;

    // 距离计算
    int daysToToday() const {
        QDate nextBirthday(today.year(), birthday.month(), birthday.day());
        return today.daysTo(nextBirthday);
    }
};
```

- 持久化：二进制文件存储（生日数据.dat）

## 四、小组成员分工情况

组长：郭梓昊：日历功能实现，天气查询、未名运势、励志寄语、生日管理、iCal 导入等拓展功能实现。

组员：韦 易：前端设计，界面设计，UI 美化，自制图标。

夏中豪：日程管理存储功能实现，日程热力图拓展功能实现。

## 五、项目总结与反思

本项目成功设计并实现了一款面向北京大学师生的综合性桌面应用，我们可以叫它“未名生活日历助手”。项目目的在于集合日常所需的时间管理、精神激励及校园特色功能，打造一个便捷丰富的校园生活工具。经过团队成员的共同努力，各项核心功能均能够稳定运行。

本项目有一些亮点与成功之处：功能整合度较高，成功将日历、天气、日程管理、运势、寄语、生日提醒、iCal 导入等独立的功能模块有机整合。同时还有鲜明的校园特色，“未名运势”功能巧妙结合 DeepSeek API，生成了具有北大元素的专属运势；“励志寄语”库也加入了许多具有学术氛围的语句。这些设计显著提升了项目的丰富程度，也是小组成员在多次讨论中创新与智慧的结晶。

本项目也存在一些不足与未来可以改进的方向。数据持久化机制方面，当前日程和生日数据分别采用纯文本(.txt)和二进制(.dat)文件存储，在数据结构复杂度提升和跨平台兼容性上存在局限。扩展功能上，日程提醒目前仅通过界面加粗显示，缺乏系统级的提醒（如桌面通知、声音提醒）。“励志寄语”内容库的丰富度和更新机制有待加强。UI 优化上：缺少设置选项供用户自定义部分功能（如热力图颜色方案、默认视图等）。

反思：本次程设 QT 大作业项目开发是一次成功的开发实践。我们小组在制作中不仅学习到了 Qt 框架开发跨平台应用的诸多技能，更将 API 调用、面向对象编程、UI 设计等所学知识学以致用。协作过程锻炼了团队成员的技术能力、解决问题的能力 and 团队沟通能力。期待在后续版本中持续完善，带来更卓越的体验。