

简易日历设计文档

计算机系 计 55 班 张瑞喆，2015011371，E-mail: zhangrz15@mails.tsinghua.edu.cn

一、软件用途

本软件是一个多功能日历，主要用于日程维护和文件管理。

二、功能介绍

- a) 能显示一个月内的日期，并标注星期和周数。
- b) 支持单日编辑
 - i. 可编辑日程
 - 1. 对于日程信息，记录描述、开始时间、结束时间。
 - 2. 同一个日期可以有多条日程，将自动按照时间顺序展示在日期对应的格子里。
 - 3. 可删除日程，并支持一键删除一个日期内的所有日程。
 - ii. 可编辑颜色
 - 1. 日期颜色分为背景色、字体颜色两个部分，可以分别编辑。
- c) 可批量添加日程
 - i. 可以指定某一事件，每隔任意数量的年月日就重复一次，可指定重复次数。
- d) 实现拖拽功能
 - i. 可将外部文件拖入日历窗口，文件将记录在当前被选中的格子内。
 - ii. 文件进入格子内时，会被拷贝到指定的文件夹，便于管理。

- iii. 为防止误操作，可以选择关闭该功能。
- iv. 文件复制时有进度条，结束时提示复制成功。
- v. 可将内部文件拖出到其他文件夹

e) 支持导入导出

- i. 为方便用户存档，可将当前日程保存到文档。
- ii. 可读档，下次使用时可以直接从文档读取相关内容并显示到日历上。
- iii. 存档采用通用的 XML 文件格式，易于与其他软件交互。

f) 拖拽和透明度

- i. 可以任意拖拽日历或放大缩小窗口，日历将显示在指定的窗口内。
- ii. 可设置日历的透明度，透过日历可以看见背景信息。
- iii. 当透明的窗口被固定时，鼠标信息会穿透该窗口，传达给下一层的窗口。

g) 国际化

- i. 支持中文、英文两种语言，并能够在程序中切换。

h) 多用户

- i. 设有注册和登录功能，用户可以维护自己的日程。

结构与功能实现

一、" addschbatch.h"

基类: QDialog

功能: 实现批量添加日程。

函数:

信号:

`void batchadd(QDate, schedule, int, int, int)`; 发送批量添加文件的信息。

槽:

二、" calendar.h"

基类: QCalendarWidget

功能: 实现主界面的日历。

函数:

`void setColor(QColor& color)`; 设置画笔颜色。

`QColor getColor()`; 获取当前画笔颜色。

`void paintCell(QPainter * painter, const QRect & rect, const QDate & date) const`; 重载绘制格子的函数，所有对日期的绘制都调用这个函数完成。

`void dragEnterEvent(QDragEnterEvent *event)`; 拖入窗口事件的处理。

`void dropEvent(QDropEvent *event)`; 在窗口内释放鼠标左键的处理。

`void mousePressEvent(QMouseEvent *event)`; 在窗口内按下鼠标左键的处理（触发一个拖放）。

信号:

槽:

`void addNote(const QDate &)`; 打开一个对话框，并编辑、修改删除日程及格子颜色，由双击格子触发。

`void get_day_edit_changed(QDate, dateinfo)`; 接收格子编辑信息，并进行存储。

`void acdropt()`; 打开文件拖放。

```
void acdropf();禁止文件拖放。  
void fixon();固定窗口。  
void fixoff();解除窗口固定。
```

三、" celledit.h"

基类: `QDialog`

功能: 实现编辑单个格子时弹出的窗口。

函数:

信号:

`void changed(QDate, dateinfo)`;发送修改信号, 在点击“ok”时触发, 将修改应用到数据库。

`void des_changed(QString)`;日程文本信息修改。

`void st_changed(QTime)`;开始时间修改。

`void et_changed(QTime)`;结束时间修改。

槽:

`void cbkcolor()`;点击修改背景颜色按钮时触发, 弹出颜色编辑对话框, 并根据该对话框操作修改背景颜色。

`void ctcolor()`;点击修改文本颜色按钮时触发, 弹出颜色编辑对话框, 并根据该对话框操作修改文本颜色。

`void merge()`;发送修改信号 `changed(QDate, dateinfo)`; , 在点击“ok”时触发, 将修改应用到数据库。

`void selc_ch(int)`;被选中的项目切换, 更新下方栏目, 以便修改。

`void ss()`;更新所有日程信息, 刷新本对话框内信息。

`void del_all()`;删除本日所有信息。

`void changeEvent(QEvent* e)`;用于接收语言修改信号。

四、“ croundprocessebar.h”

基类: `QProgressBar`

功能: 绘制传入文件时的进度条。

函数:

`void setScanValue(QString strValue)`;设置当前进度。

信号:

槽:

五、“ dateinfo.h”

基类: `QDate`

功能: 用于存储单日信息的文件。

函数:

```
QVector<schedule> get_sch_list(); 获取事件列表。  
QColor get_text_color() const; 获取当前文本颜色。  
QColor get_back_ground() const; 获取当前背景颜色。  
void clear_sch(); 清空事件列表。  
void add_sch(schedule sch); 添加事件。  
void del_sch(schedule sch); 删除事件。  
void change_text_color(QColor new_color); 修改文本颜色。  
void change_back_ground(QColor new_color); 修改背景颜色。  
void sort(); 对事件按开始时间排序。  
void del_all(); 删除全部事件。
```

信号:

槽:

六、" func.h"

基类: `QWidget`

功能: 常用的通用接口, 本软件中仅有复制文件。

函数:

信号:

槽:

```
void copyFile(const QString&, const QString&); 复制文件, 并带进  
度条显示。
```

七、" login.h"

基类: `QDialog`

功能: 用户登录窗口。

函数:

信号:

`void loi(QString, QString)`; 在点击“ok”时发送，将输入的用户名和密码发送给核对器核对。

槽：

`void on_buttonBox_accepted()`; 点击“ok”按钮时发送信号使用。

八、“mainwindow.h”

基类： `QMainWindow`

功能： 绘制传入文件时的进度条。

函数：

`void in_port(QString urls)`; 读取存档的日程文件。

`void ex_port(QString urls)`; 存储当前日程到文件。

信号：

`void changed(QDate, dateinfo)`; 修改某一天的日程信息。

槽：

`void on_actionAdd_schedule_triggered()`; 按下批量添加日程按钮时触发，弹出相应对话框。

`void batchadd(QDate, schedule, int, int, int)`; 批量添加日程。

`void set_low()`; 将透明度设置为低透明度。

`void set_high()`; 将透明度设置为高透明度。

`void set_no()`; 将透明度设置为不透明。

`void choose_file_1()`; 弹出对话框，让用户选取要读取的文件。

`void choose_file_2()`; 弹出对话框，让用户选取要存档的文件。

`void new_u(QString usr, QString pwd)`; 添加新用户。

`void chk_u(QString usr, QString pwd)`; 核对用户名密码并确认是否登录成功。

九、“resign.h”

基类： `QDialog`

功能： 注册用户的对话框。

函数：

信号：

`void res(QString, QString)`; 发送用户注册时填写的信息。

槽：

`void on_buttonBox_accepted();`点击“ok”时触发，用于调用信号发送注册信息。

十、“schedule.h”

基类：

功能：存储单个日程信息，作为 dateinfo 下的数据格式之一。

函数：

`QString describe();`返回日程文本信息。

`QTime start_time();`返回日程开始时间。

`QTime end_time();`返回日程结束时间。

`void set_describe(QString new_describe);`设置日程文本信息。

`void set_start_time(QTime new_time);`设置日程开始时间

`void set_end_time(QTime new_time);`设置日程结束时间

信号：

槽：