

Event类:

需要派生出:

课程类:

类信息:

- 课程名称
- 课程编号(id对外不可见)
- 某一节课的编号
- 上课时间(20240425 34)
- 上课地点
- 课程的详细信息

类方法:

- 新建课程
- 删除课程
- 修改课程
- 查询课程

新建课程

需要两种新建课程的方法:

1. 直接提供信息新建课程
2. 提供课程信息文件新建课程

建议课程信息文件只支持csv格式(减小工作量)

(需要的课程信息有: 课程名称, 课程编号(id对外不可见), 某一节课的编号, 上课时间(需要详细区分单双周), 上课地点, 课程的详细信息)

因此, 建议直接提供信息新建课程的方法为调用 `lectures_creat(string name, string id, string time, string location, string info)`函数, 注意一次调用需要随之创建很多课程, 建议写一个`lecture_creat()`子函数.

提供课程信息文件新建课程的方法为调用 `lectures_creat_from_file(string file)`函数, 该函数需要读取文件, 解析文件, 然后调用 `lectures_creat()`子函数.

删除课程:

删除课程有两种方式:

1. 删除某一节课
2. 删除本课程的所有的课

对于删除某一节课, 前端提供了课程名称,id, 上课时间, 上课地点, 课程的详细信息, 需要在数据库中找到对应的课, 然后删除.

对于删除本课程的所有的课, 前端提供了课程名称,id, 上课地点, 课程的详细信息, 需要在数据库中找到对应的课程, 然后删除.

修改课程:

修改课程有两种方式: 1.修改某一节课 2.修改本课程的所有的课

大概同上

需要一个修改函数

查询课程

查询课程的目的在于前端需要获取所有的课程时间名称地点, 然后能够在日历上显示出来.

需要一个查询函数, 该函数需要返回所有的课程信息, 课程名称, 上课时间, 上课地点, 课程的详细信息.

todo类:

类信息:

1. todo名称
2. 某一天
3. todo编号(可以把这个id跟上面的时间id结合起来, 形成一个唯一的id, 但是不需要对外提供)
4. todo的详细信息
5. todo的完成状态(完成/未完成)
6. 优先级

需要派生出:

1. 新建todo
2. 删除todo
3. 修改todo
4. 查询todo
5. 完成todo

新建todo

新建todo有两种方式: 1.直接提供信息新建todo

因此, 建议直接提供信息新建todo的方法为调用 `todo_creat(string name, string id, string info)`函数,

删除todo:

删除todo前端会查询该todo的编号, 然后在后端数据中找到这个todo进而删除

修改todo:

修改todo前端会查询该todo的编号, 然后在后端数据中找到这个todo进而修改,

仅需要修改todo的名称, 详细信息, 其他的都不需要修改.(调用`todo_modify(string newname, string newinfo)`函数即可)

查询todo

查询todo的目的在于前端需要获取所有的todo信息, 然后能够在日历上显示出来.

需要一个查询函数, 该函数需要返回所有的todo信息, todo名称, todo编号, todo的详细信息, todo的完成状态.

完成todo

完成todo前端会查询该todo的编号, 然后在后端数据中找到这个todo进而修改完成状态为完成, 需要一个函数完成这个操作.

该函数需要返回一个bool值, 表示是否完成成功. (调用todo_finish()函数即可)