

《暑期 Python 课》

大作业实验报告

组 号____40____

组员 1 张柯 19376075

组员 2 黄星阳 20373356

2022 年 8 月

一、实验任务

题目 1

内容：

随着大学生活日益丰富，家庭作业和活动变得更加交织在一起。因此，需要设计一个应用程序，帮助规划自己的日常日程，合理记录任务，更好地规划的时间。

要求：

1. 需要使用 Tkinter 和 PyQt5 等 GUI 库来设计 GUI。
2. 提供日历、任务状态区分等功能清晰简洁的工作系统，历史数据回顾、任务调度等。
3. 界面要简洁明了，美观大方。其他功能可根据您的需求添加能力，但最终用户应该能够轻松地导航系统。

二、已完成任务

1. 基本实验要求

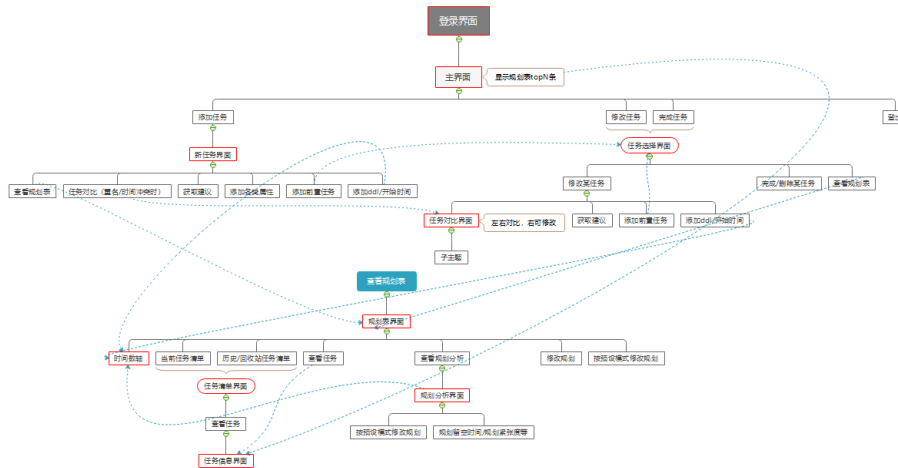
1. 任务添加系统。用户应该能够创建新任务。任务应包括标题、内容
2. 描述、期限、重要性等。同时可以允许修改和删除。
3. 显示每日任务。显示每天需要完成的任务。
4. 确认任务完成。当一个任务完成后，用户可以通过某种方法删除该任务，
5. 例如勾选复选框。
6. 日历系统。用户可以通过查看日历查看一个月中每个日期的任务计划。
7. 任务状态区分。区分未开始、进行中、已完成和过期的任务状态。
8. 任务调度。通过设置任务列表和组织空闲时间来自动安排任务。
9. 用户登录系统...

2. 可选要求

1. 待办事项系统可以按时间过滤和显示任务。
2. 任务类别划分。设置任务类别，如运动、学习等。
3. 数据恢复。系统可以根据历史任务完成数据进行数据分析。
4. 支持日常任务。日常任务只需要设置一次，就会自动出现在

- 每天指定时间内的任务列表。
- 加权任务调度。通过制作任务列表和设置任务的优先权重，可以制定任务计划
- 自动安排，同时也分配空闲时间休息

三、总体设计方案



总体的方案设计流程图如上，获取 page 信息后，转化为单一 id 的结构体，进行信息的处理和分析流程

四、创新之处

1. 说明你采用了什么独特的方法或者课堂介绍方法之外的方法来解决某个问题；增加了哪些功能；介绍你设计的亮点在于什么地方。
2. 使用了大量通过自定义信号使用信号槽方式而非重写事件的方式完成许多逻辑要求，如窗口拖拽，QWidget 的点击触发等等。
3. 使用了 MVC 模型，如任务类别的下拉选单，作为 QComboBox 使用了 QListWidget 的 model 和 view
4. 任务列表的自定义排序
5. 大量自定义了重写了原生的类，以实现逻辑等的封装
6. 采用 ui 与逻辑分离的方法，ui 调整作用于一个单独的文件。
7. 设计了一种调度算法，在完成全部任务的大前提下，优先保证高优先级任务率先完成。同时可以有两种优先顺序，完成相应分析
8. 设计了若干帮助用户体验的部件，方便进行大小调整，环境设置和动画

设置等等

9. 创新的使用了加密 xml 文件，进行用户信息的存储和读取
10. 大量使用了 CSS 样式表进行界面美化
11. 创新的使用添加了背景音乐，可以提升体验
12. 创新的使用了鼠标点云图，可以提高代入感

五、实验总结

1. 自定义窗口的标题栏。原生的标题栏难以自定义且不符合要求，必须重写，然而重写需要完成窗口拖拽、最大化、最小化等一系列逻辑的实现，尤其是窗口拖拽。在实验中，选择对标题栏自定义信号，采用信号槽机制解决此问题。
2. 日历界面的栅格。首先，日历界面中的栅格必须既能作为容器，容纳部分组件，能实现显示日期和任务功能。其次，栅格必须具有按钮的部分特性，能记录并发射选择状态，此外，日历界面中的栅格的选择也必须具有唯一性，即选中一个默认取消其他栅格的选择，同时，日历界面栅格的选择还需与各种日期编辑方式（如日期编辑、上下月/周等）协同……在逻辑之外，还需考虑到编辑内容的便利性，由于切换日期频繁，内容编辑应当具有效率高、集成程度高的特点。本处选择完全重写，通过对栅格容器和栅格本身两大类进行重写。栅格容器继承自 `QWidget` 类，而栅格继承自 `QFrame` 类。在其中通过事件等的重写和自定义信号和槽完成信息的交互。利用父控件可以定向查找子控件的特性建立映射关系，是的栅格容器能统一管理其中的栅格，栅格又能管理其中的 `QLabel` 和 `QListWidget`。
3. 任务类别的管理。任务类别选择使用 `QComboBox` 进行选取，然而，如果不进行改进的话，首先任务类别的删除难以实现，其次 `QComboBox` 的下拉选单过于简单。本处选择重写 `QComboBox`，并将其模型和视图委托给 `QListWidget` 完成，同时自定义 `QListWidget` 中的项，最终使得下拉选单中即可完成任务类别的删除。
4. 任务列表的排序功能。原生的排序功能一旦打开，将作用于每一个表头，然而实际上这与需求不符，部分表头不应作为排序的选择，本处重写了表头切换排序事件，选择性的禁用了排序。

5. 标题栏无法正确应用样式表。设置标题栏的样式表时发现标题栏实际上无法应用样式。最终发现继承 `QWidget` 的子类无法正确使用父类的样式，最终通过重写 `paintEvent` 事件完成样式的设置

六、课程学习总结

（一）、课程收获

1. 了解学习了 Python 的基本语法和特点。Python 的语法与 C/C++ 以及 Java 有着很大的不同，语法更加的灵活多变，同时弱类型的特性也让 Python 虽然使用方便，但是难以调试。由于弱类型的特性，在任务调度算法中，我可以返回调度失败这一布尔值，又或者返回整个完整的调度结果，这些都是 C 语言所做不到的。

当然弱类型也使得 `pycharm` 的联想功能不如 `idea` 那般强大，不过总体来说，以我的体验，还是值得的牺牲（如果不考虑后续可读性的话）。

2. 了解学习了图形化界面编程，无论是 Python 内置的库，还是丰富的第三方库，同时 Python 跨平台的特性均让 Python 适合编写一个图形化软件。

此外，还接触了新的 `qt` 环境，由于此时正在 `pyside2` 和 `pyside6` 的换代期间，许多想要的功能不得不在已有文档，或者更好的表现之间进行抉择。

（二）、难点分析

1. 语法的变化一定程度上影响了学习效率，尤其是 Python 对于换行和缩进有着严格的要求。

2. 图形化界面 `ui` 与逻辑的结合较为困难。`Qt designer` 是一个完完全全的面向对象设计思路，各个部件结构层次分明。这本来对于我们这些 OO 毕业生非常友好，只是由于 `Qt designer` 糟糕的手感和低熟练度，以及 `qt` 格式的学习成本，使得有时效率低下。

3. 英文文本阅读，一来是对于作业字里行间信息的理解，二来是对于国内成熟文档较少的 `qt` 领域，需要大量的阅读生肉文档。

（三）、教师授课评价（不计入分数，请尽量给出真实评价或者建议）

1. 难度适中，适合作为暑期课程
2. 实例丰富，尤其是图形化界面一节中与学生一起进行实际操作

（四）、助教评价（不计入分数，请尽量给出真实评价或者建议）

1. 助教非常耐心细致

2. 助教在出题是也难易适中，数据也较为友好，错误的点往往自己都能找到原因

(五)、当前课程教授内容评价与课程进一步改进建议（不计入分数）

总体而言内容比较完整和丰富，建议扩大图形化界面内容的比重。以及对于作业具体条目（对于非英语母语的学生来说，一两个字的歧义可能就意味着设计方向的跑偏），进行进一步的解读

七、主要参考资料

<https://doc.qt.io/qtforpython/modules.html>

<https://www.riverbankcomputing.com/static/Docs/PyQt5/sip-classes.html>