**第六次实训工作报告**

软件2班 杨璐玮

**本周工作规划：**

1. **新增日程排序功能！**
2. **完善用户的团队操作（删除团队、退出团队）**
3. **完善团队的日程管理模块**
4. **新增日程排序功能！**

**自定义排序函数，根据combobox的值调用不同的排序函数**

**排序的过程中修改了几个小bug（结束时间的格式和删除操作程序崩溃）之后更新QTableWidget的数据，遇到问题**QTableWidget: cannot insert an item that is already owned by another QtableWidget

**使用 clone() 方法是成功解决问题的关键，因为它创建了原始单元格对象的副本，而不是直接移动原始对象。下面是对最后一次修改成功的原因进行总结：**

**克隆副本保留了数据完整性：**

使用 clone() 方法创建原始单元格对象的副本，保留了原始数据的完整性。这样，在重新设置行顺序时，不会直接移动原始对象，从而避免了数据丢失的问题。

**避免了对象所有权混乱：**

单元格对象只能属于一个表格，当尝试将已拥有的单元格对象插入到另一个表格时，会引发错误。通过克隆副本，避免了对象所有权的混乱，确保每个副本单元格都属于正确的目标表格。

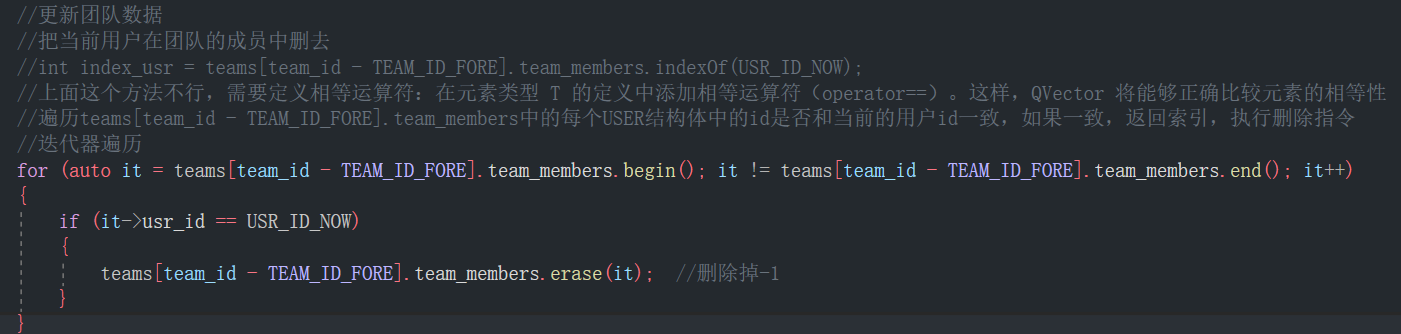
正确处理对象的生命周期：

在使用 clone() 方法创建副本时，需要确保在不再需要副本时及时释放副本对象的内存，以避免内存泄漏。这样可以正确处理对象的生命周期，避免了潜在的内存问题。

总结起来，使用 clone() 方法创建副本是一种常用的解决方案，可以保留原始数据的完整性，并避免对象所有权的混乱。通过正确处理对象的生命周期，可以确保代码的健壮性和可靠性。然而，每个问题都有其独特性，因此在实际应用中，根据具体情况选择适当的方法和技术是非常重要的。

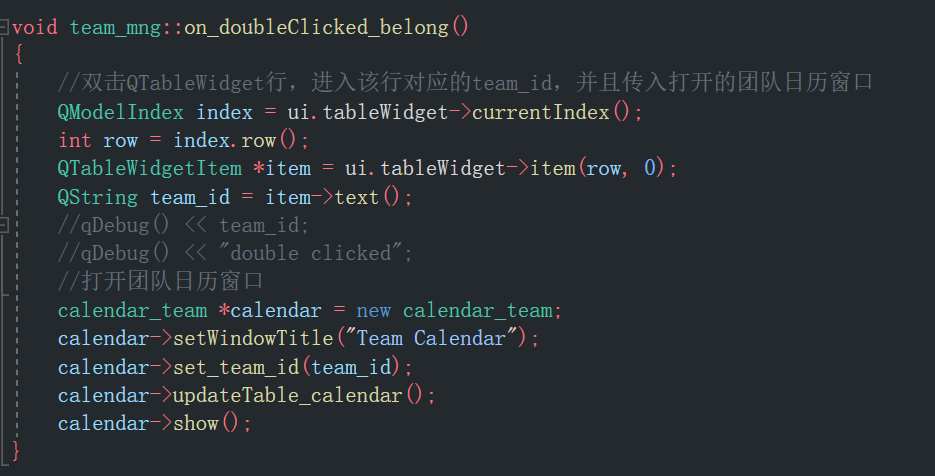
1. **完善用户的团队操作（删除团队、退出团队、唤起团队管理窗口）**

需要注意的是，在实现删除和退出团队的功能的时候，我需要根据用户选中的随机的一个团队进行定向的操作。于是我就需要根据团队的id检索对应的文件以及数据结构进行操作。由于我是用QVector进行删除和退出操作的，而一旦这个操作发生，元素索引会发生变化。所以如果使用普通的循环遍历会导致程序崩溃。这里必须使用迭代器来实现这个操作。



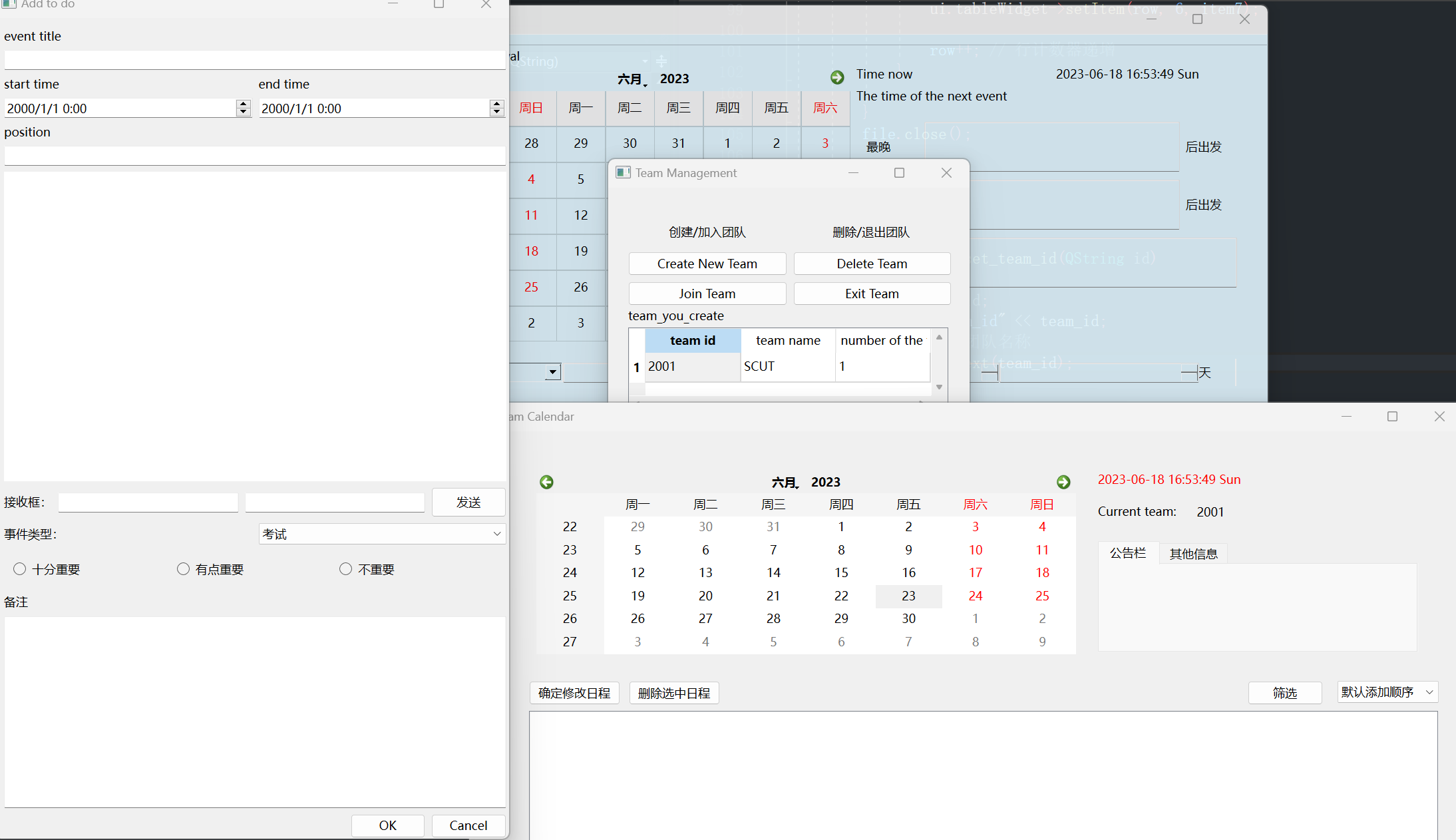
在此之前我也探索了一些其他的方式，但是涉及到QVector的外部代码限制，没有成功。

为了改善用户的体验，我设计了一个双击团队信息视图中的某一行，就能唤起团队日程管理窗口的功能。

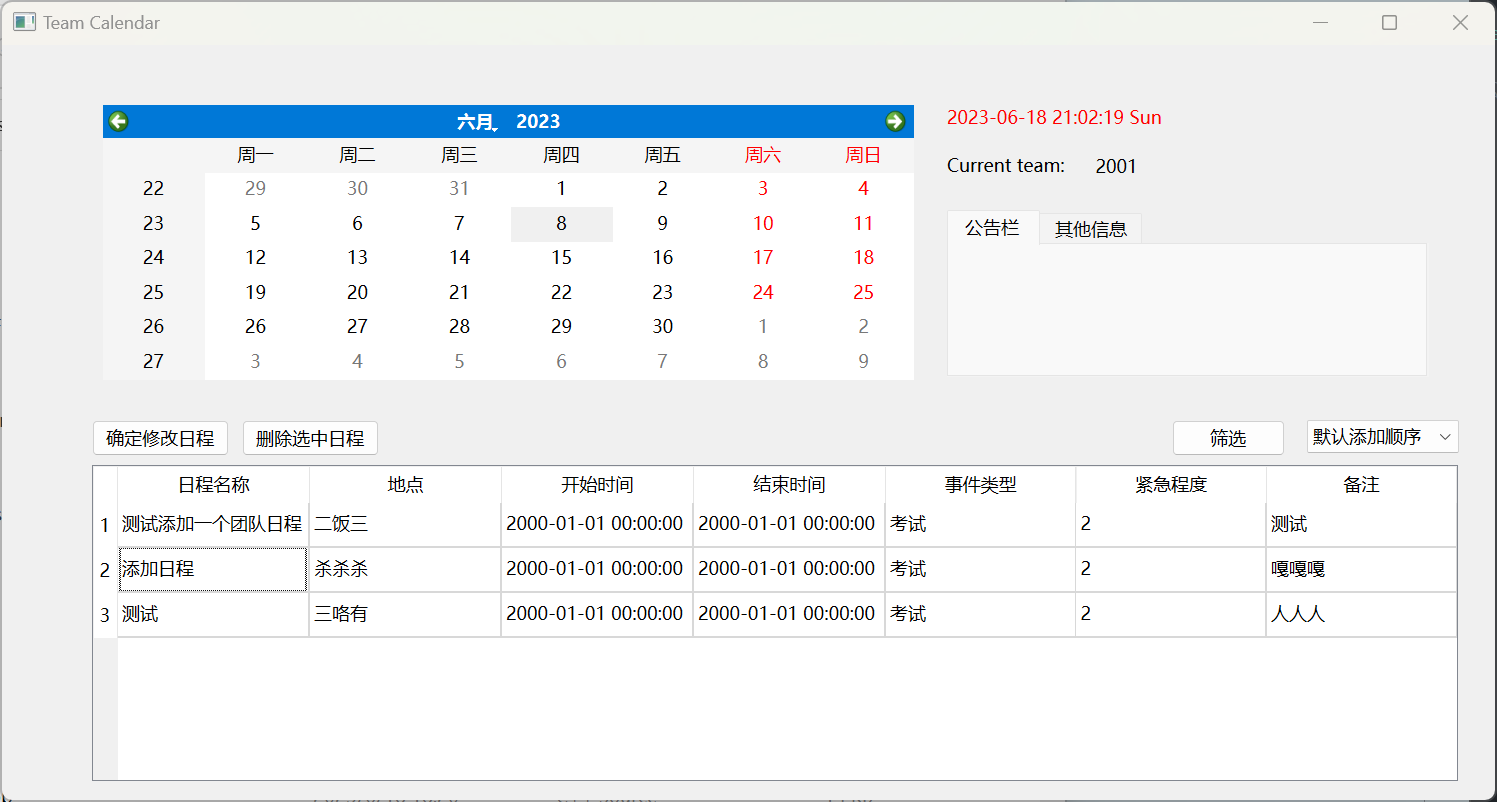


1. **继承并重写QTableWidget类，新增团队日程增删改实时显示功能！使用发送信号的方式替代比较蠢和麻烦的计时器来改变实时显示的实现。日程新增改变后会会后重新读取文件进行更新）**

**关于删除，使用迭代器循环遍历检索**

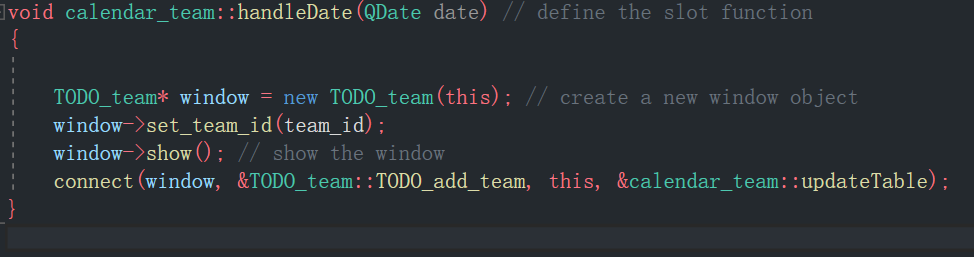


由于窗口的大体ui和功能和用户添加日程是类似的，于是我选择继承TODOWidget类，派生出TODO\_team来特殊化团队日程的管理。并且重写一些需要更改的函数，以满足团队日程的管理需要。



这个功能的实现涉及到一个问题。这里因为同一个用户可以对多个团队的日历进行操作，所以需要让每次操作的对象准确定位到对应的团队以及其相应的文件。

而原本的设计是在calendar\_team(上周创建的用于管理团队的窗口类)中双击日期唤起日程添加窗口的时候，连带着把所需要的参数通过调用对象的函数传给对象。但是测试的时候发现没有如愿。



仔细检查之后发现，这其中存在一个逻辑问题，因为原本是想在日程添加窗口被唤起的同时显示日程信息，所以我是在构造函数中调用更新表格的函数的，但是参数只能等到窗口对象被创建成功之后才传进去，这就导致了在显示日程信息的动作结束之后，所需要的参数还没有交接。所以最后我选择了在传入参数之后调用更新表格的函数。也就是在获取参数的函数内部调用更新日程信息的函数，这样就解决了这个问题。

需要提到的是，曾经实时显示功能是使用计时器实现的。但是在后续使用和测试的过程中发现实在是不方便。因为我还需要再写一个能够停止计时器，让用户完成操作之后再继续计时的一组信号和槽函数。而且在单次添加日程的过程中不需要一直刷新，使用定时器实现“伪实时”其实是开销很大的。所以我最后把**实时显示功能迭代为在改变发生后重新读取。**我只需要在用户完成操作之后的函数结尾的地方调用一次更新表格内容的函数即可，非常方便高效。

**小结：**

**非常遗憾的是，由于低估了团队管理的代码量和复杂度，我在这上面花的时间比较多了些，而且很大一部分是在debug。在尽量保持代码复用性的情况下，目前基本实现用户和团队的所有基本功能，代码已经到达5000多行。下周我会将重心放在日程调度上，希望能够如期实现项目的目标。**