

类的功能与实现——以项目2为例

使用类代替结构体与函数式编程的好处在这里不作详细讲解，大家可以自行看书；一般来说利用类对代码进行组织，不同功能的函数与数据结构会被联系起来，使得整个项目的逻辑更加清晰，维护、协作开发更加便捷

以项目12306为例：

对于这样规模的任务，利用函数式编程的话我们需要编写若干结构体与函数用于主函数的调用来进行实现，这样做并非不行，但是十多个甚至几十个函数来回调用不仅复杂麻烦，而且容易看晕

利用面向对象的思想考虑这一任务：所谓购票系统，参与这一活动的对象其实就是**管理员与用户**；管理员可以独立地对系统中的车次信息进行录入、修改、查询，用户则可以独立地进行票务信息保存；售票与退票则需要管理员与用户共同完成

因此，实现这一项目，至少我们可以先定义**管理员与用户**这两个类。

一个最简单的类也应该包含它的成员变量与成员函数，细节可以参考课本、网上的讲解与实例

对于管理员类，它要操作的数据结构就是一系列的车次信息，而车次信息的内容就是班次号、最大承载数、各停靠站点，所以我们可以先将车次信息封装为一个基本的数据结构体：

```
1 struct train
2 {
3     string shift;//班次号
4     int max_num;//最大承载数
5     vector<string> route;//路线
6 }
```

这里通过 `#include <string>` 与 `#include <vector>` 引入了string与vector这两个数据结构，它们的功能与实现等细节在此不再赘述，有兴趣的同学可以查询相关资料学习并自己完成实现

有了这一结构体，管理员类的成员变量即可定义：

```

1 class admin
2 {
3     private:
4         vector<train> trains;
5     public:
6         admin();//构造函数声明
7         .....
8 }

```

关于private与public，大家可以结合课本在实际练习中感受它们的区别 绝对不是因为我忘了

管理员要完成的具体任务通过成员函数实现：

```

1 class admin
2 {
3     private:
4         vector<train> trains;
5     public:
6         admin();//构造函数声明
7         void import();//用于从文件导入车次信息，该函数要读取文件信息存
            储到trains中
8         void read();//手动导入车次信息
9         void add();//添加车次信息
10        void del();//删除
11        void revise();//修改
12        void query_shift();//按班次号查询
13        void query_station();//按起始站查询
14        .....
15 }

```

上述代码只是对成员函数进行了声明，之后需要各位进行具体实现；实现时函数传入什么参数、具体如何完成自定

这样，之后如果我们要从文件导入车次信息，就可以：

```

1 admin administer;//注意类和对象的区别
2 administer.import(.....);//从文件导入车次信息

```

tips:添加时注意是否重复，修改删除要保证该车次无人购票

对于用户类，它需要哪些成员变量与成员函数请自行思考(tips：需要存储自己买的票、自己的账号密码、自己的历史订单(因为之后可能会从文件中将票务信息导入到历史订单中，可以考虑实现一个ticket结构体用于存储订单的数据)；需要有查询功能与买票退票的功能、登录功能、查询历史订单(从文件中导入票务信息).....)

之前提到过售票退票需要管理员与用户交互完成，在有以上两个类的基础上，我们可以写一个函数，根据输入来调用这两种对象的函数完成任务，但更好的选择是构造第三类来整体管理：

```
1  class My12306
2  {
3      private:
4          admin adminster;
5          vector<user> users; //用户不止一个
6          .....
7      public:
8          My12306();
9          //构造函数
10         void buy_ticket(.....); //这里的传参需要思考：购票可以拆分为管理
            员检查余票、
11         //订单浏览、选择是否保存票务信息、管理员更新车次信息、用户更新自己
            状态
12         //以上每一点都应该作为子函数封装起来，思考这些函数应该分别放在哪个
            类的成员函数中
13         //Q： 你如何确定是哪一个用户在买票
14
15         void refund(.....); //仿照买票进行实现
16         .....
17 }
```

这样，在最终的main函数中，实际上我们只要定义一个My12306的对象就可以满足任务的需求；可以在My12306类中实现成员函数 `run`，在其中接收输入，并执行相应的功能函数。

编写以上框架代码可以体现出你对这个问题的思考方式与解决思路，之后的成员函数具体实现或是函数添加在此基础上都能够顺利推进。希望大家能够逐渐习惯这种思维方式，一步步构建起自己的项目代码