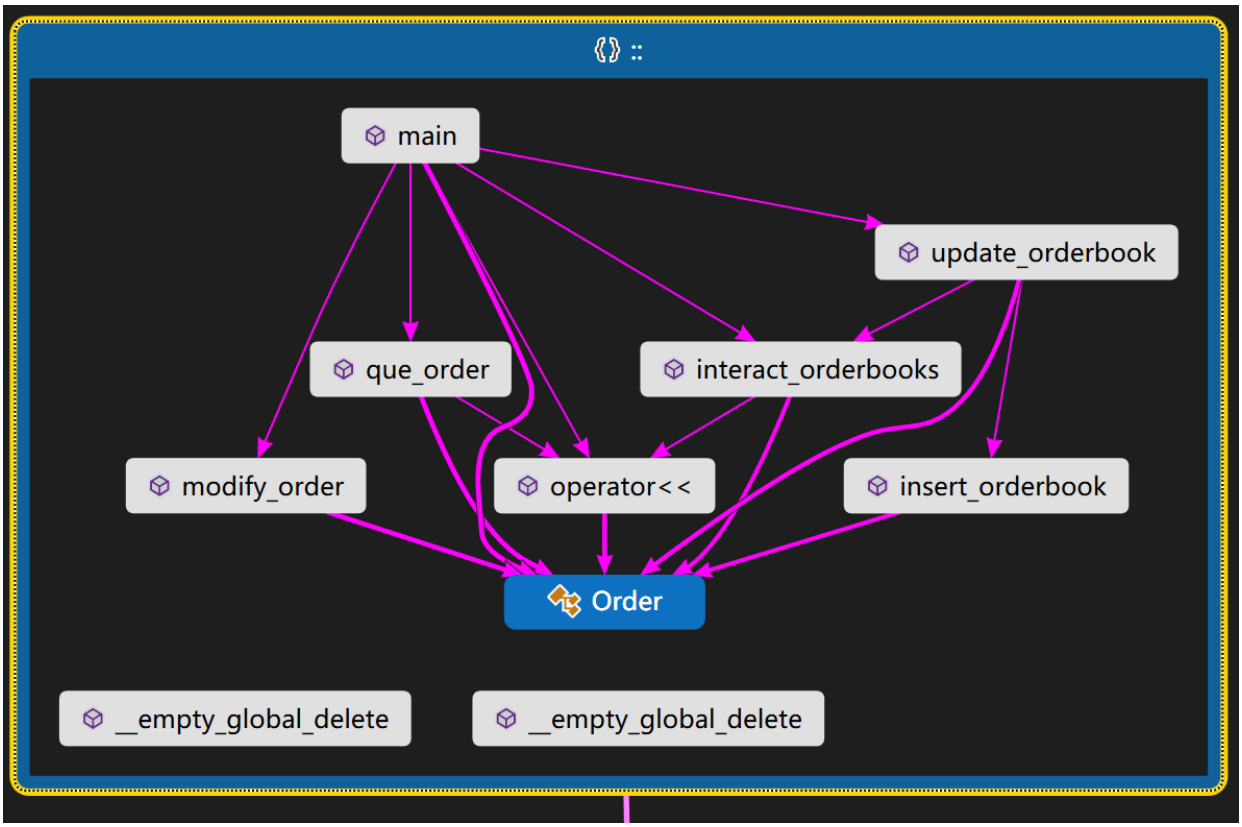
Hw5 Developer Manual

Developer Manual and Design of our program

0. Overall :

- We use infinite message loop to achieve the interact of users and program.
- We designed a Class called Order to handle single order data.
- We designed a data structure with map and vector to maximized the speed of dealing orders. We believe that the speed of dealing trades is the first priority and it is reasonable to sacrifice the speed of query to enhance the program's performance in dealing.

The design of overall program is shown below:

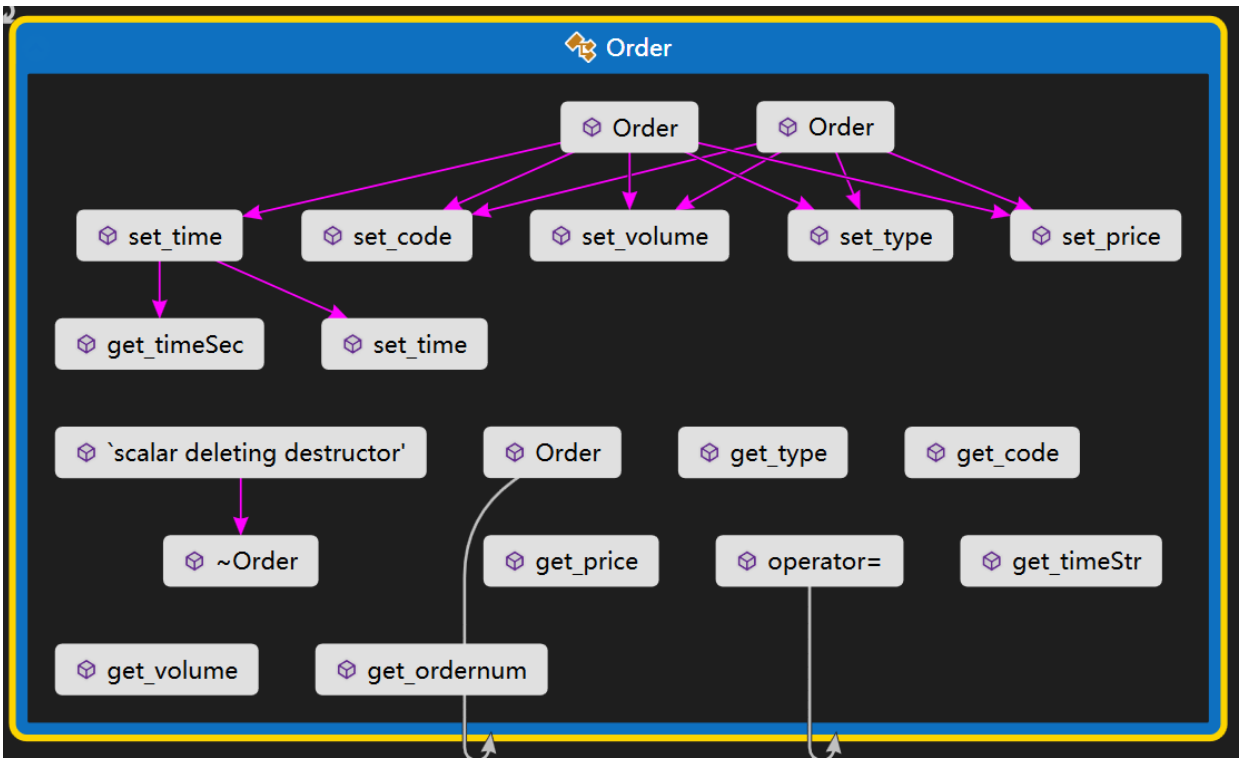


1. Class:

- We designed a Class called Order to handle single order data.
- Order class has many getters and setters to work on it's attributes. Each getter or setter has it's own error check module to ensure the input data is in correct format and value.
- It also has two different constructors to construct itself. User can define the time of it's order or not.
- Every order has it's own unique order#, achieved by a global variable called global_ordernum. The order number of each order can not be modified.

- We overload the << to output orders easily.
- What's more, the time in class is stored as timestamp as default. We can output the time in string format via get_timeStr.

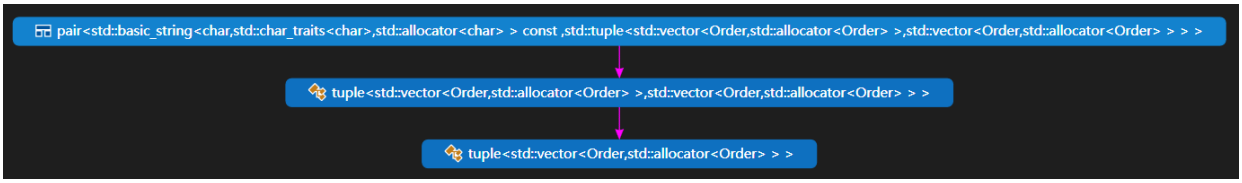
The design of Order class is shown below:



2. Data Structure

- We designed a data structure to maximize the speed of dealing orders.
- We use map to generate a dictionary of order list. The key is the code of every order and the value is a tuple with two different vector, one stores buy orders and the other keeps sell orders, in this tuple. We used structured bindings in this section.
- The reasons we choose this data structure are: Firstly, We believe that the speed of dealing trades is the first priority and it is reasonable to sacrifice the speed of query to enhance the program's performance in dealing. And every deal has a common base requirement that is dealing on the same security. So the code of order is the main key. Secondly, orders can have many keys and they do not have a certain order while searching or dealing except the code of orders. So we use vector to store the orders in maps.
- The reason we use two vectors for different trading direction is due to the performance and reality issues.

The design of Data structure is shown below:



3. Local functions

- Every time the order book changes, there is a check for changed order list to find whether there are deals to make.
- The design of query function is inspired by homework2. And it allows AND and OR to search.