Intro Excercises to Optibook

These excersizes are designed to familiarize yourself with Optibook, and how to use it. They will walk you through the different "interactions" that are possible with Optibook. You should have successfully completed the "Getting Started" guide before you tackle these problems.

Important: All problems are solvable using the "Manual.ipynb" notebook, so make sure to have that open. You should also open the raw code of the client find the file called `common_types.py` in order to see what fields different objects have.

Problem 1

Buy 10 lots of SP500 at any current price, and use the optibook client to double check your position. You should have +10 lots.

Problem 2

Wait a minute or two, and then sell 10 lots of SP500 at any current price, and use the optibook client to double check your position. You should now have 0 lots. Go ahead and play around with inserting orders at different prices into the order book. Insert an order and see what happens in the visualizer! Try to make a few trades.

Only move on to problem 3 once you are comfortable with the idea of buying and selling. Also make sure your position is 0 when you move on to problem 3, it will make the problem a lot easier.

Problem 3

Get the trade info of the trades you made, and calculate the "PnL" (Profit & Loss) of your trades. If you are unsure how to do this, reason through this on paper first, what price did you buy SP500 at, and how many lots did you buy? At what price did you sell it again, and how many lots did you sell? Did you make any money? Now turn this into code.

Use the `e.get_pnl()` function to verify that your calculation is correct.

Problem 4

Insert 10 orders to buy SP500 at a random number higher than 100000. This order should not trade, as the price is way to high. Now, check your outstanding orders, do you see it? Next, ammend your order to only trade 5 lots, and check your outstanding orders again, do you see the change?

Finally, cancel all outstanding orders, and check again to make sure they have been cancelled correctly.

Problem 5

Look at all public trades that have happened since the exchange has started, for all users, and print them!

Problem 6

Get the current order book using `e.get_last_price_book(instrument_id)` and write a function to print it with descending prices, in the "common" order book representation. This would be an example:

```
bid | price | ask
    |  100  | 2
    |  99   | 1
 1  |  98   |
```

Don't look at the Visualizer.ipynb notebook, try to do this yourself.

To print, simply do something like this (this is purely meant to give you some inspiration, your solution does not have to look like this):

```
print("bid | price | ask")

for level in processed_order_book:
  print(f"{level.bid_volume} | {level.price_level} | {level.ask_volume}")
```

Bonus: Problem 7

Write a generic PnL function that calculates your current PnL. This is similar to Problem 3, but you will need to take a few more things into account:

1. What happens if you have an outstanding position? You need to value it against the current market price. For simplicity, take the mean between the bid and the ask.
2. What happens if you have a short position?
3. As you enter trades, you will accumulate "cash", you can see this in the visualizer as well, next to "Pnl" and "Position". This is the amount of money you spent, or received upon entering a trade. For example, say you buy 100 lots at 11 dollars each. Then you will have a position of 100, and a cash amount of (100 * -11) = -1,100. The 11 is negative, because you are paying money. You will have spent 1,100 dollars to aquire 100 lots. When you short-sell, it is the other way around! You will have negative positions, but a positive cash, as the counterparty pays you. So say you sell 10 lots at 11 dollars, then your position will be -10, but you will have received (100 * 11) = 1,100 cash in return. Depending on how you build your PnL function, you might have to take this into account.

Double-Bonus: Problem 8

By now you should have a pretty good idea of how you can interact with Optibook. If you are bored and want to continue, I challenge you to think of how you can combine all this information into an "automated" trading algorithm. The goal is not to make profit here, but rather to come up with a structured approach of how you might trade in an automated way, regardless of the strategy employed.