**BCPR301 – Advanced Programming**

**Assessment 3 Marking Sheet**

Student Name Zhong Wei

# The compulsory (i.e., ZERO mark if not being provided)

1. You MUST submit a filled self-marking sheet to indicate how many marks you think you can get for each section in the marking guide provided below.

**I reckon that I can get a full marks of 20 \* 2, that is 40. :)**

# Marking guide (20 \* N marks in total where N = 2)

1. The class diagram before your modification (2 \* N marks): 4



1. The locations of code involved (i.e., code reference) **in your target assignment 1 or 2 solution** (2 \* N marks): 4

For Adapter pattern:

Client: tigr/drawer/drawer.py

Target: tigr/drawer/abstract\_worker.py

AdapterA: tigr/drawer/tkinter\_worker.py

AdapterB: tigr/drawer/turtle\_worker.py

For Abstract Factory pattern:

Client: tigr/\_\_init\_\_.py (line 18~21)

Abstract Factory: tigr/drawer/abstract\_worker\_factory.py

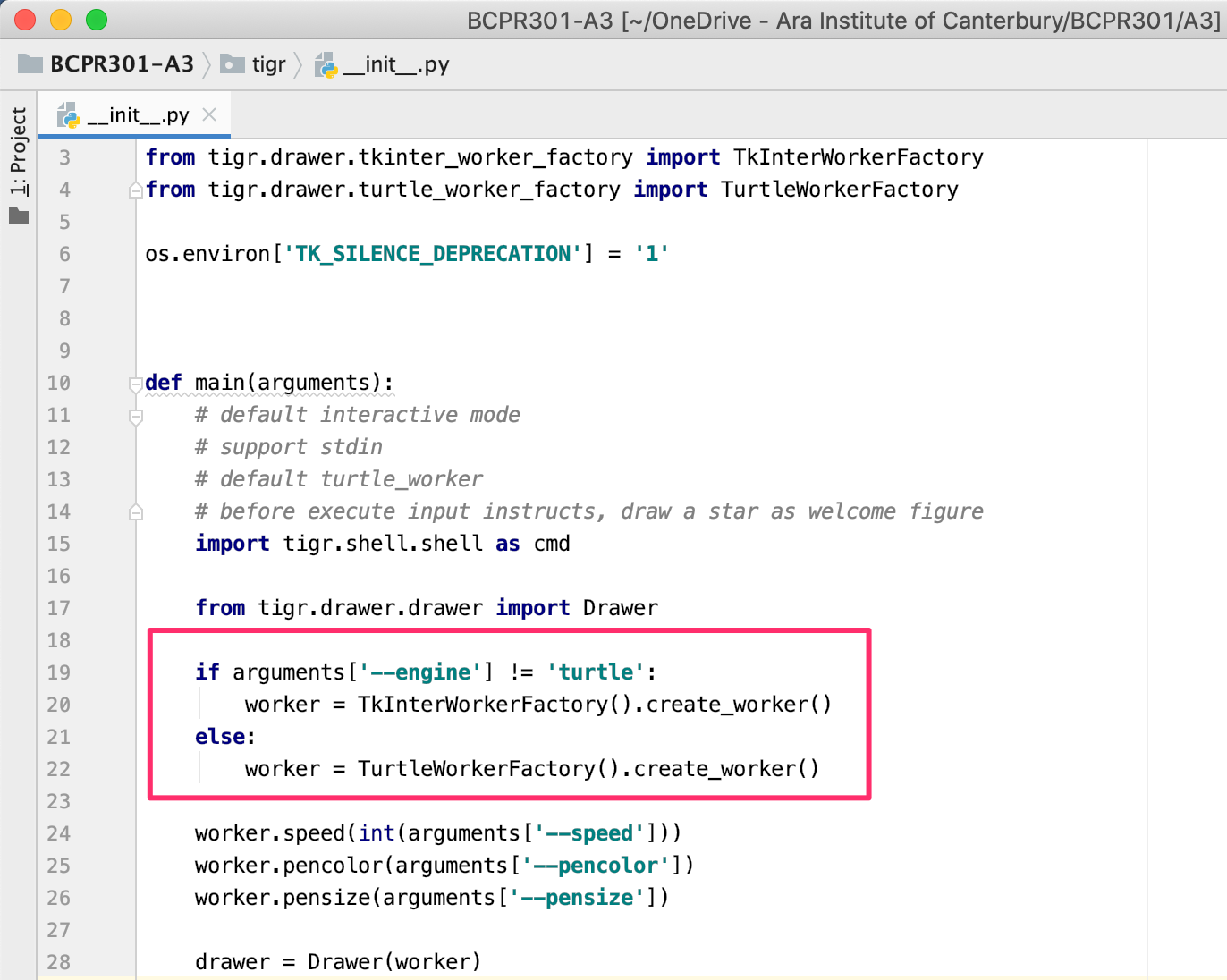
Concrete Factory A: tigr/drawer/tkinter\_worker\_factory.py

Concrete Factory B: tigr/drawer/turtle\_worker\_factory.py

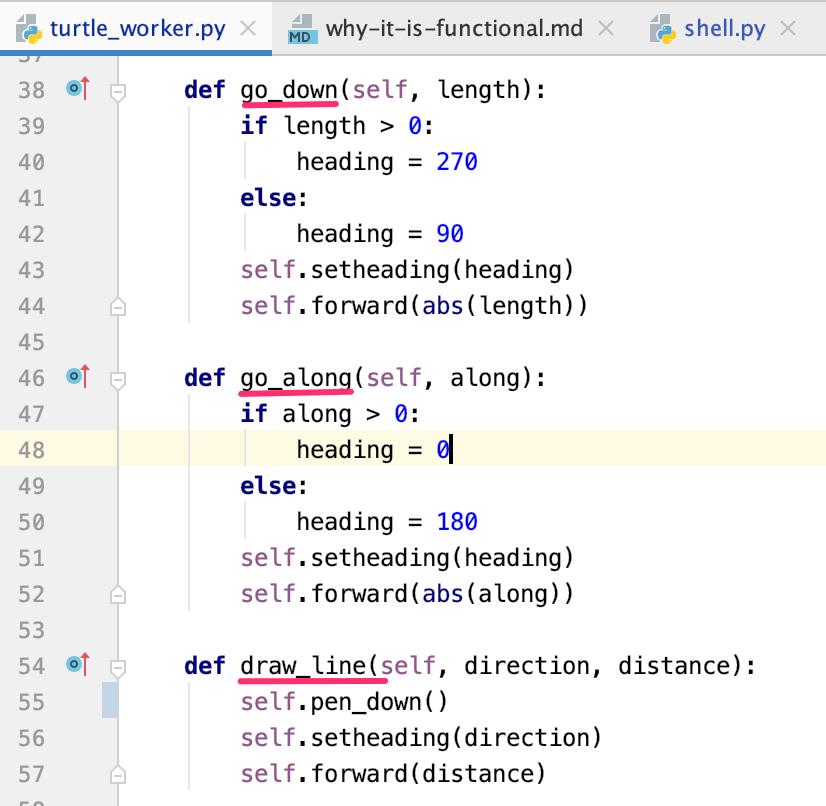
Concrete Product A: tigr/drawer/turtle\_worker.py

Concrete Product B: tigr/drawer/tkinter\_worker.py

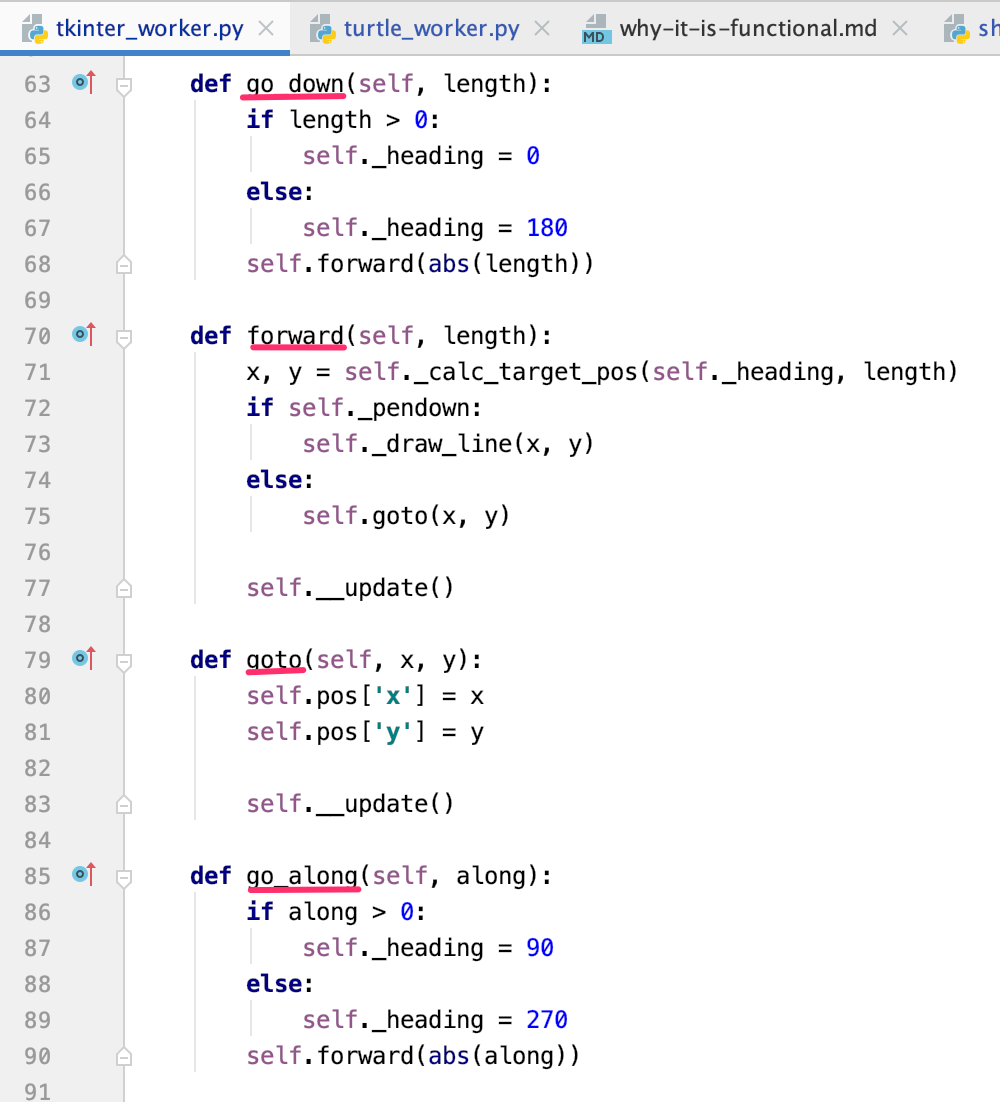
1. The name of the design pattern applied (2 \* N marks): 4
2. Adapter Pattern
3. Abstract Factory
4. The reasons why applying this design pattern is suitable; the reasons **MUST be specified** for the particular situation you try to apply, i.e., **do not just give general reasons** why using that design pattern is good. (2 \* N marks): 4
5. Let us look at the following code, the read lines are the before code, and the green lines are the after code. Before I applied the Abstract Factory Pattern, the caller or the client code have to know exactly what its name is and where it locate for the specific Worker Class. In addition, it also needs to know the initialization details of these classes (which parameters are needed). Those are bad coding practices and error-prone. The after code only need to know the work’s name and its behaviour. That is quite better than before.



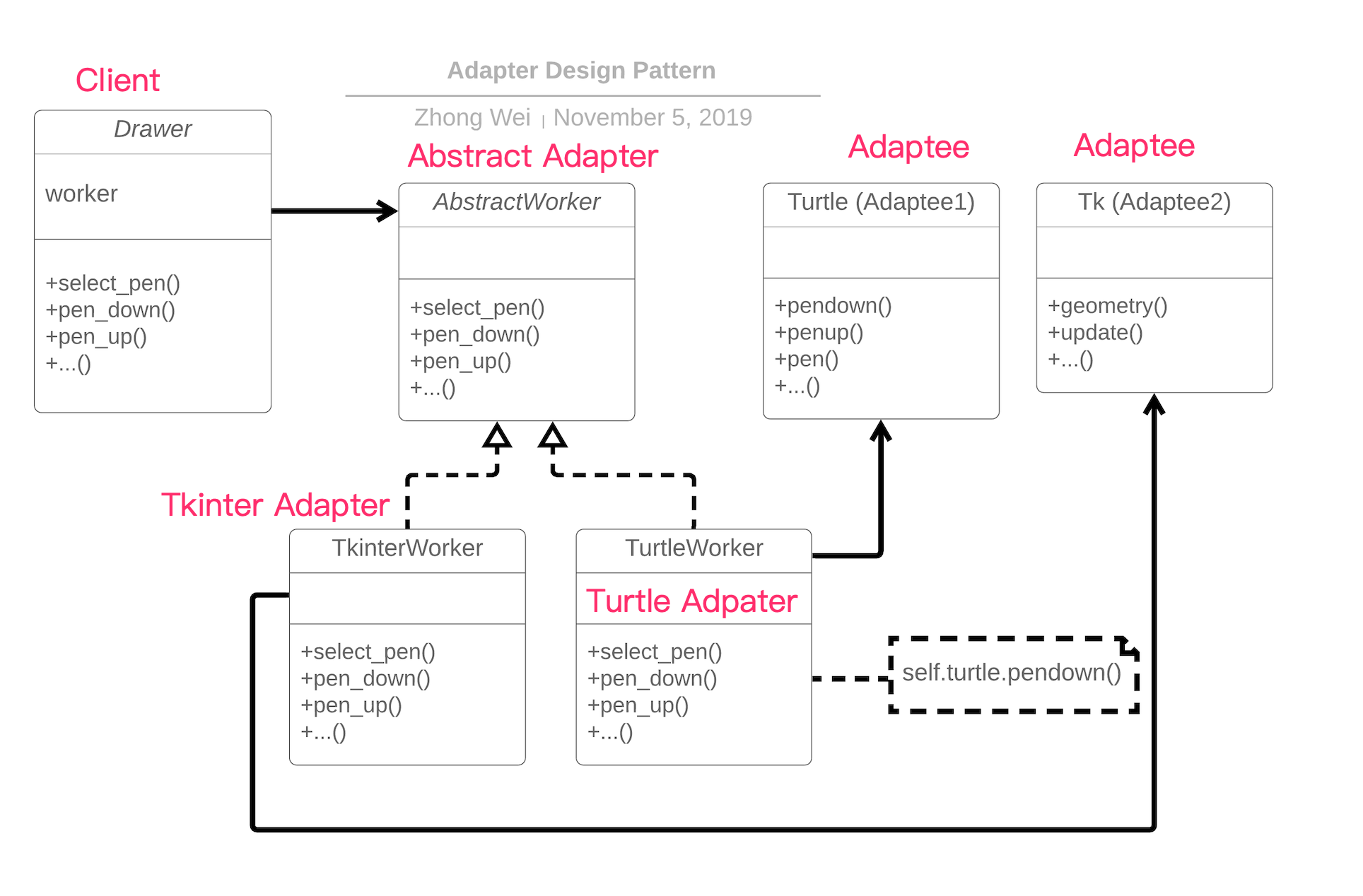
1. Let us observe the code below.

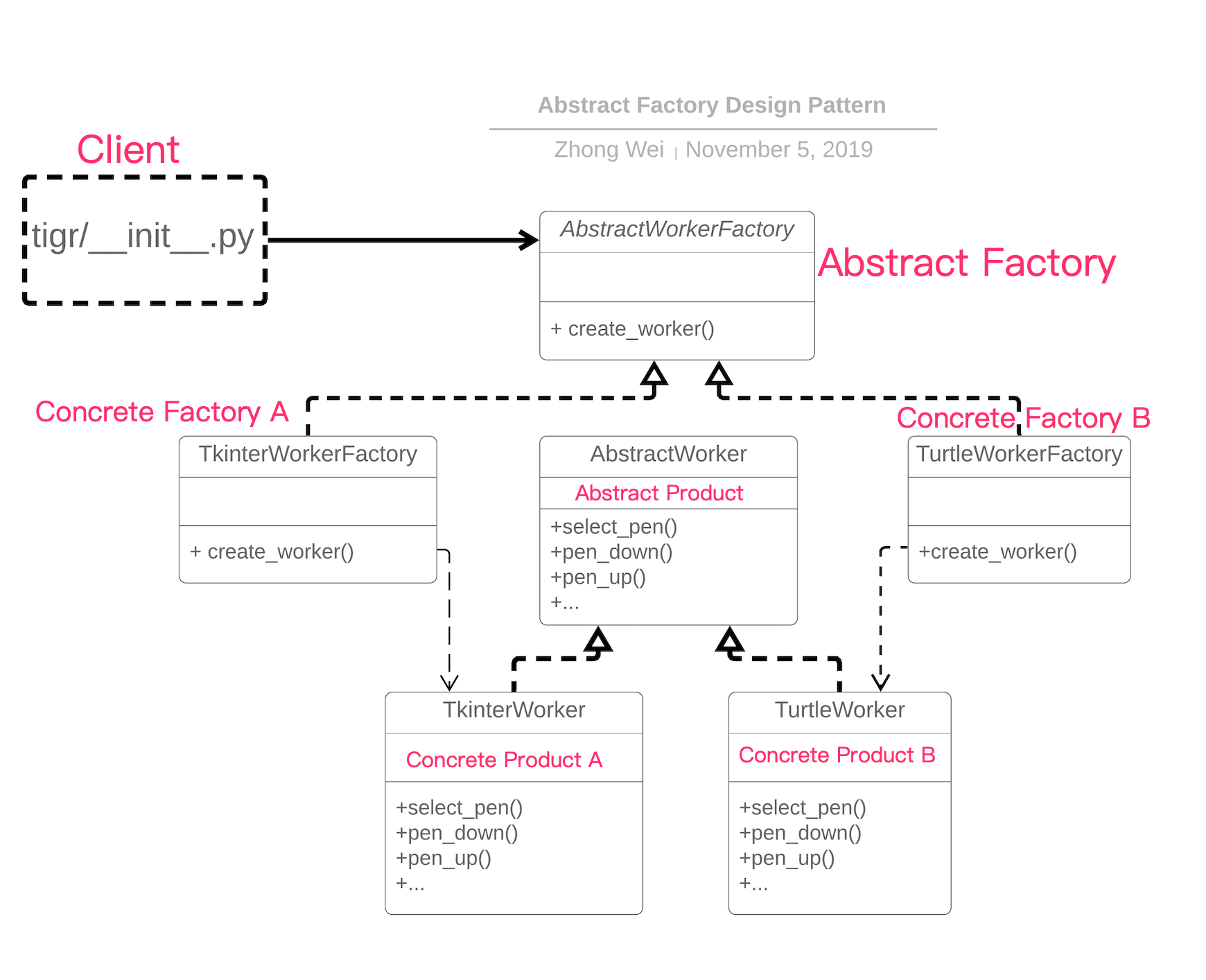


The adaptee Turtle does not have **go\_down, go\_along and draw\_line** method at all. However, it does have other methods that can be used to implement these methods. This means that Adapter pattern does suite this situation. You can see I have called several other methods to implement the necessary interface. It’s a good practice to resolve similar scenarios. You can also see similar code in my TkinterWorker.py, for example:



1. The class diagram after your modification; all the components in the design pattern class diagram provided in our textbook should be **explicitly labelled** in your class diagram. (2 \* N marks): 4





1. Applying the design pattern proposed. Your assignment 3 solution needs to pass PEP8 check (10 \* N marks): 20

Please see the code ☺