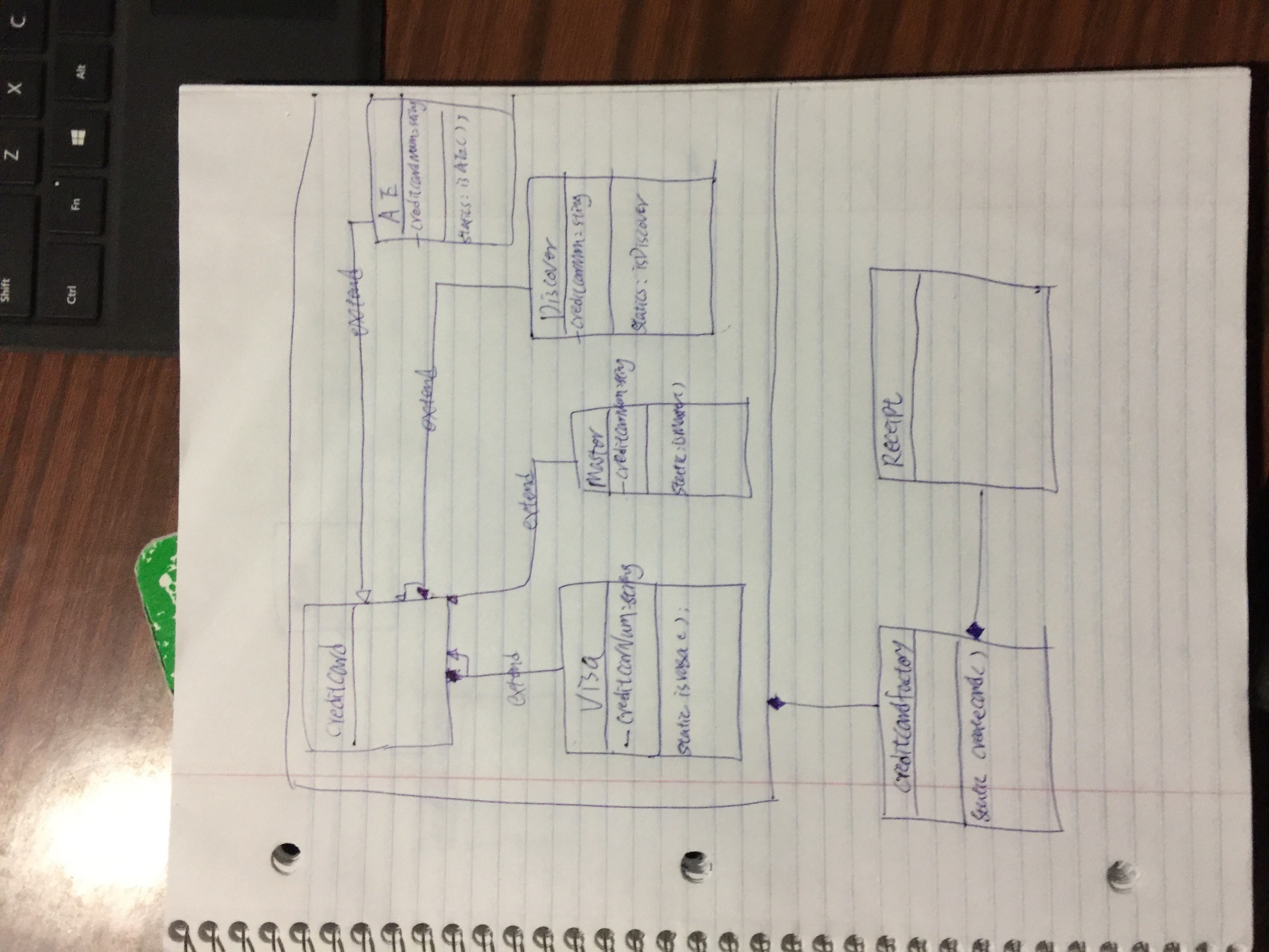
****

**Advantages**

* As the factory class remains unchanged even when new products are added, this implementation *doesn't violate the open and closed principle*.
* This implementation provides the loose coupling between Factory class and concrete products. This is because creditCard objects are created using reflection so the factory need not know the instantiation details of creditCard class.
* The client doesn't need to be aware of all the available products. Only the required products need to be registered before use. Moreover product creation is abstracted from the client.

**Disadvantages**

* Reflection makes system slow.
* Not all the languages support reflection.
* This implementation is more difficult to implement.

Class diagram

Class: View – only takes care of the UI. Creates an receipt obj to contain the information

Method: createAndShowGUI ();

ClearAll – clear the txtarea to default

AddItemtoOrder – when the user press the button, then add the order to the order panel

Class :receipt

Method: AddItem- when the user press the button and then add the food they select into the arraylist.

And add the total

GetRectiptSting – convet all the information to string return to the obj

