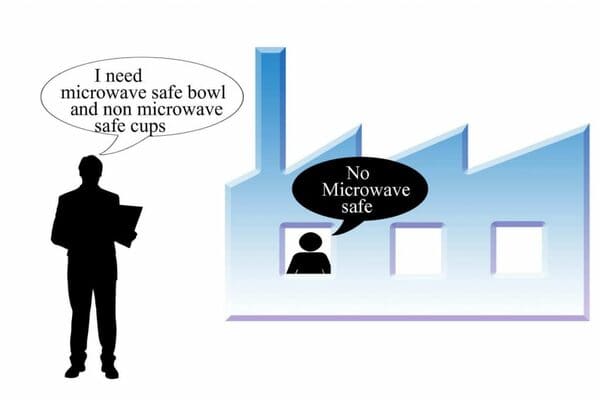
Exercise:

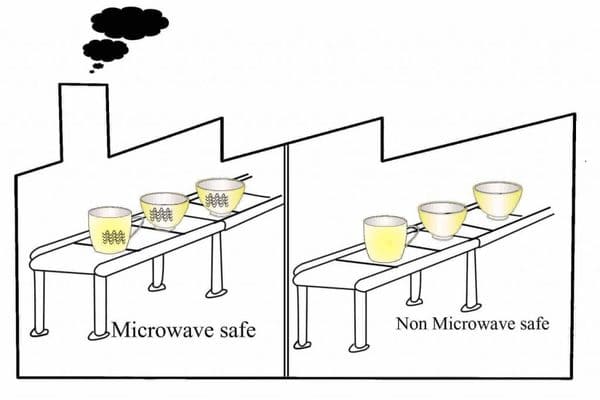
We have two groups of utensils Microwave safe and non-microwave safe products. If we need microwave safe products we should use microwave safe bowl, plate, and cup. We cannot mix microwave safe and non-microwave safe. When we need to avoid mixing, we can use abstract factory pattern by creating factories for each.



Microwave safe factory Non Microwave safe factory

Creating an object often requires complex processes. The object's creation may lead to a significant duplication of code. The factory method design pattern handles these problems by defining a separate method for creating the objects.

This pattern is useful when we have large number of objects in our program. This pattern classifies them into "Families".



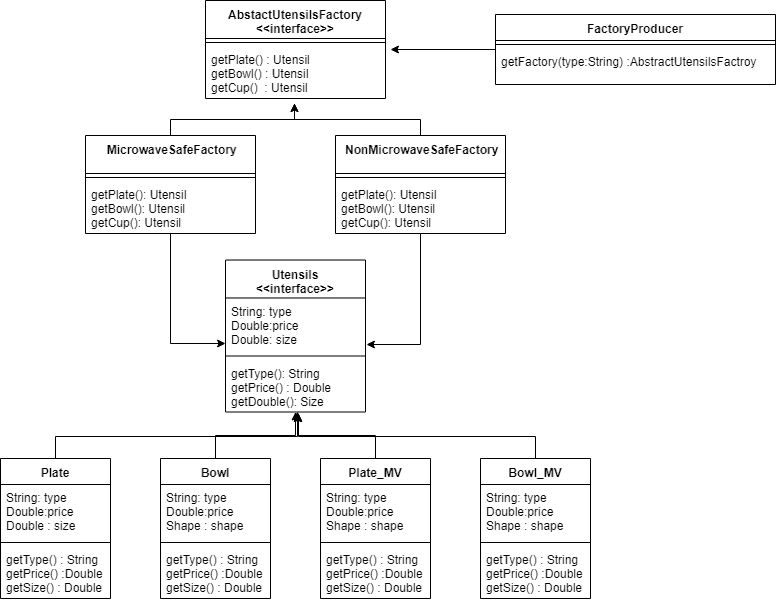
Abstract Factory pattern adds a "Factory" that brings together all these factories. Further, it decides at run time which factory should be invoked. This later creates an object of a specific family. Therefore, **this pattern is also known as "A Factory of Factories".** Abstract Factory Pattern in java encapsulates a group of factories in the process of object creation.

The essence of this pattern is that it separates the object creation from the object usage. Apart from that, this pattern also creates families of related objects. This helps to keep related objects together and different groups cannot be mixed

Steps to Implement Abstract Factory Pattern:

1. Find out the different object types in the application
2. Create Interface and implementing classes for each type  
   For eg, Utensils : Plate, Bowl, Cup, Plate\_MW, Bowl\_MW, Cup\_MW ( \_MW is used for microwave safe products)
3. Create factory classes to group the classes
   1. Microwave safe: Plate\_MW, Bowl\_MW, Cup\_MW
   2. Non Microwave safe: Plate, Bowl, Cup
4. Declare Abstract factory interface and declare all required methods from factory
5. Implement Abstract factory interface by created families
6. Create code which will use Abstract factory to get factory and then call the methods on that factory
7. Use the abstract factory in the code instate of objects directly

## UML : Basic understanding of the implementation



This is the [UML diagram](https://tallyfy.com/uml-diagram/) of the exercise discussed above for microwave safe and non microwave safe products. This diagram explains the basic blocks in abstract factory design pattern. Now you need to make the java code implementation of this exercise.

Consider a use case where we have a client application that produces different types of utensils. There are 2 major categories(factories) of utensils "Microwave safe" or "Non-Microwave safe". And the factories can create different types of utensils for eg: Plates, Bowls etc.

The details of the solution are available on: [Abstract Factory Pattern in Java - with real life examples (stacktraceguru.com)](https://stacktraceguru.com/abstract-factory-pattern/)