

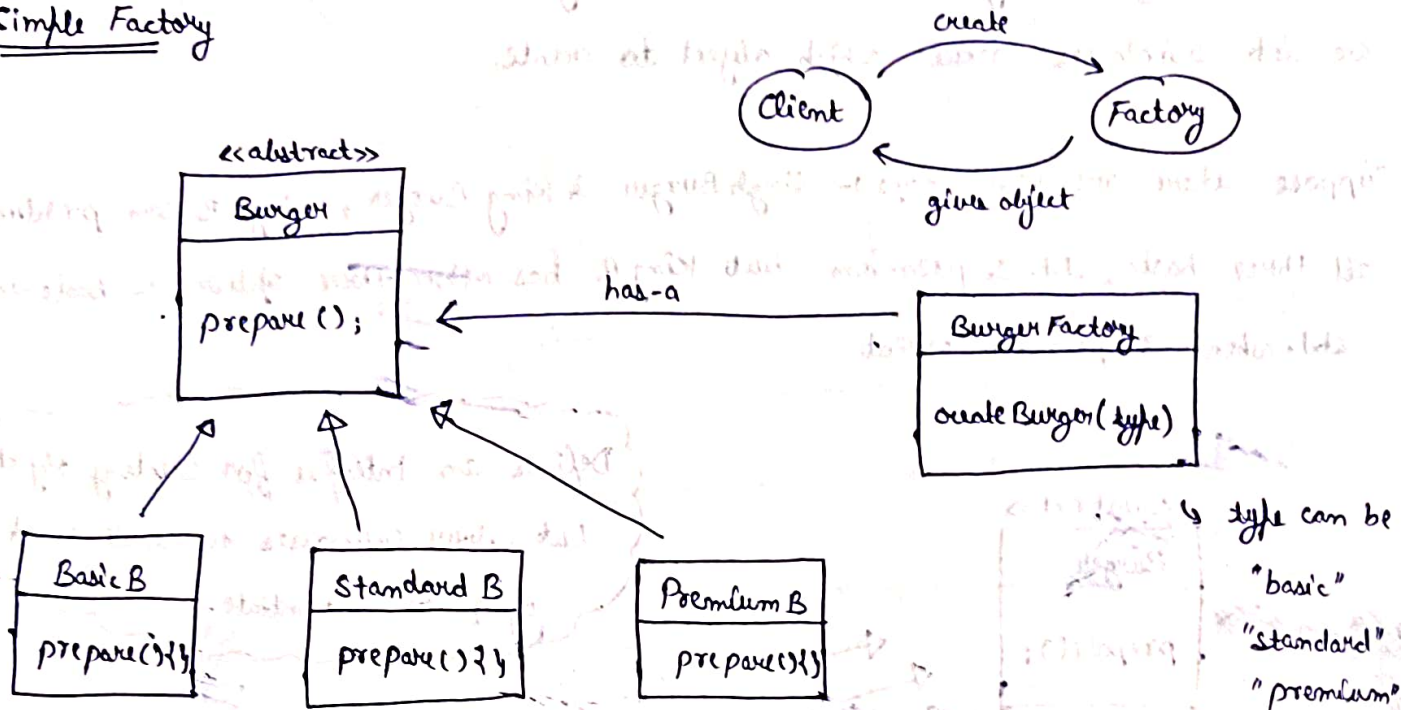
Factory Design Pattern

① Simple Factory Design Principle

② Factory Method

③ Abstract Factory Method

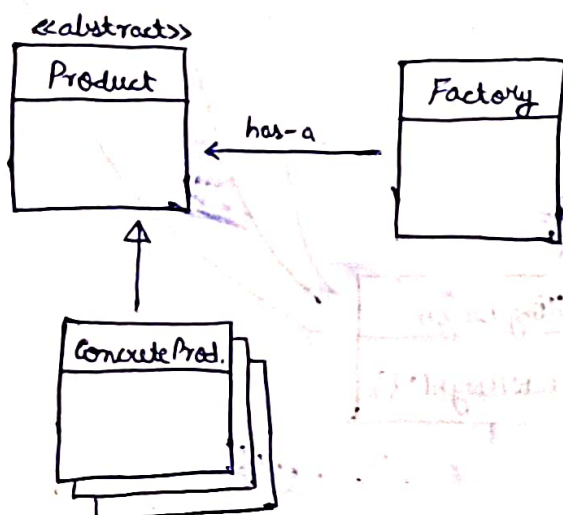
Simple Factory



So, basically what we are doing is instead of creating objects directly using new in your code everywhere, put that object-creation logic inside a separate factory class.

⇒ Centralize creation logic inside one factory class.

Standard UML for Simple Factory



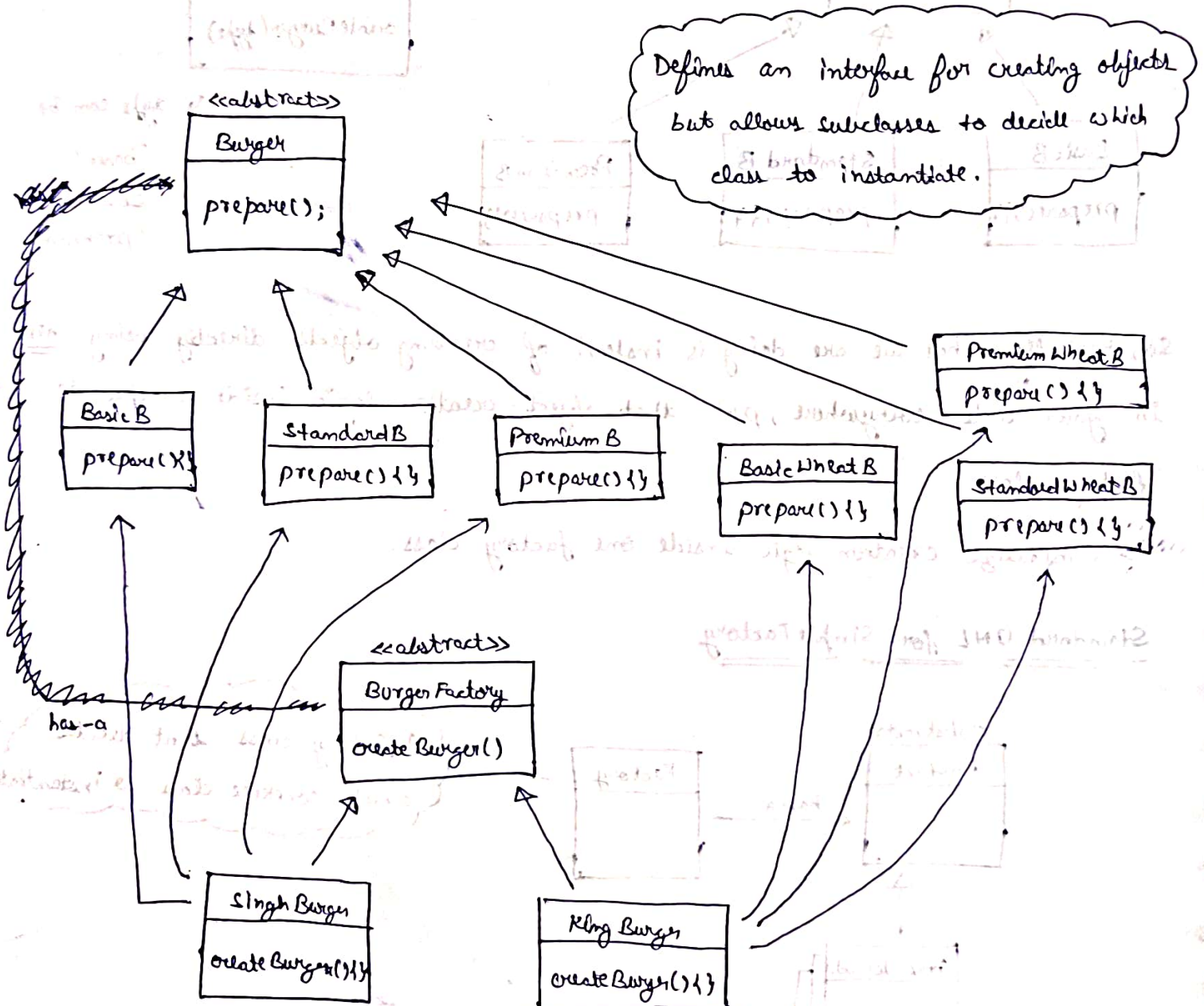
A factory class that decides which concrete class to instantiate

Factory Method

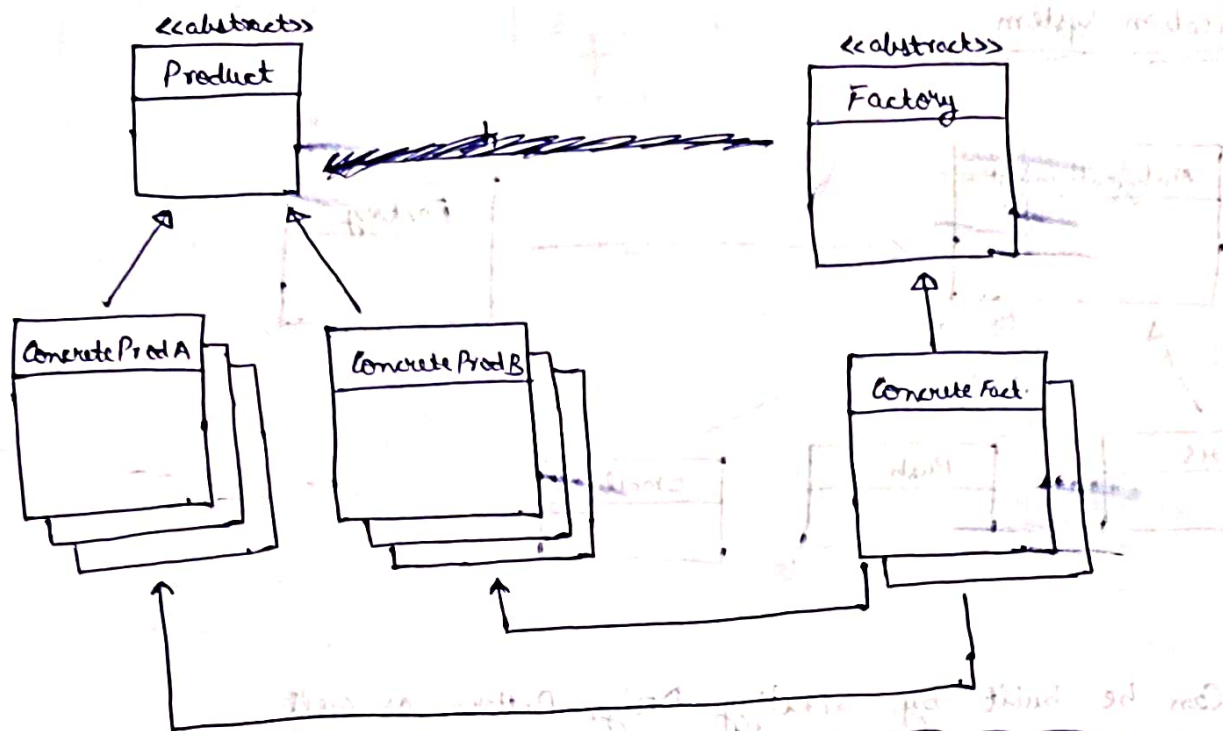
In simple factory if we add a new product, we must modify the factory!
(breaks OCP)

So, Instead of having one centralized factory that decides what to create, we let subclasses decide which object to create.

Suppose there are two shops :- Singh Burger & King Burger, Singh B can produce all three basic, std. & premium but King B has other three options :- basic wheat, std. wheat & premium wheat

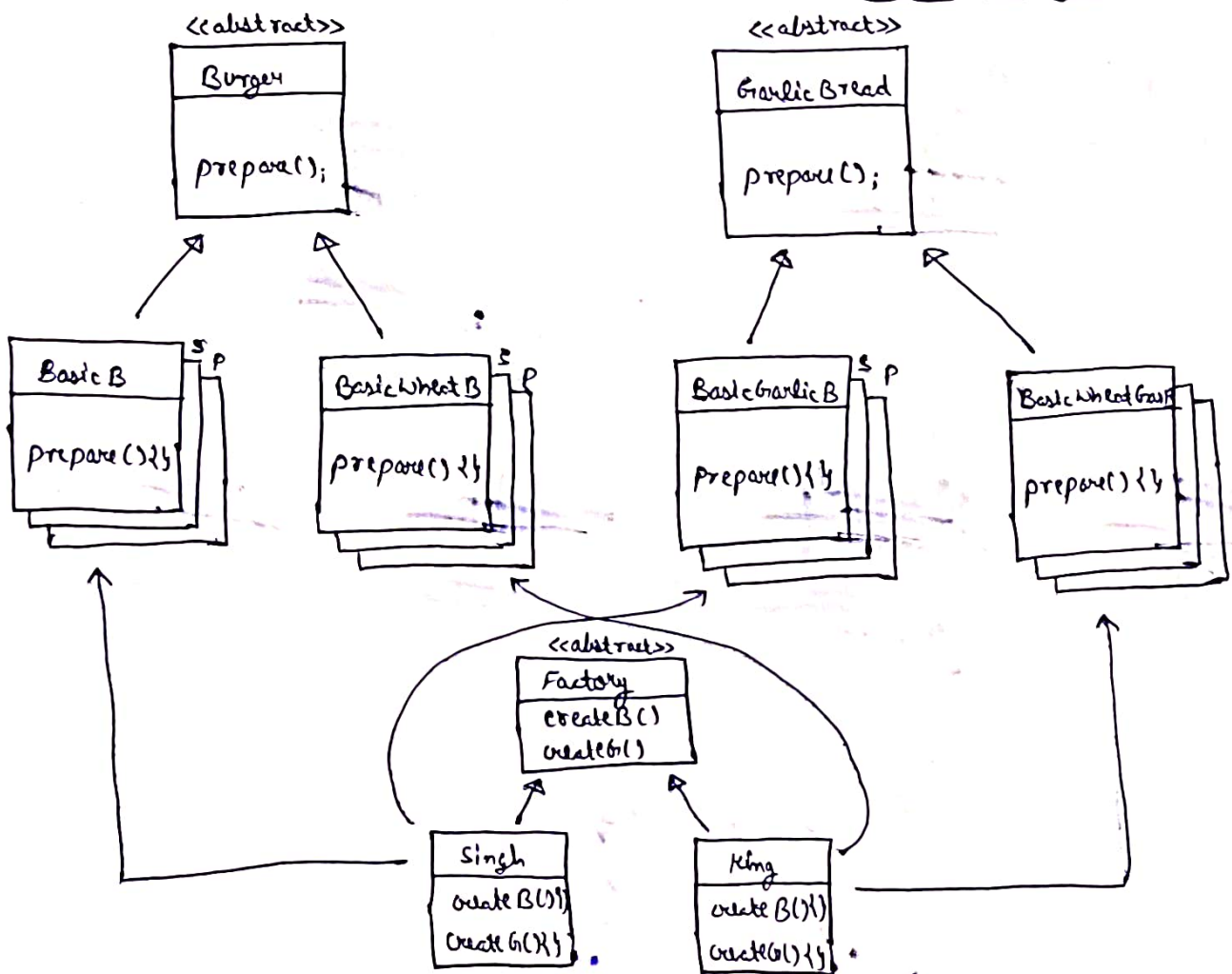


Standard UML for Factory method



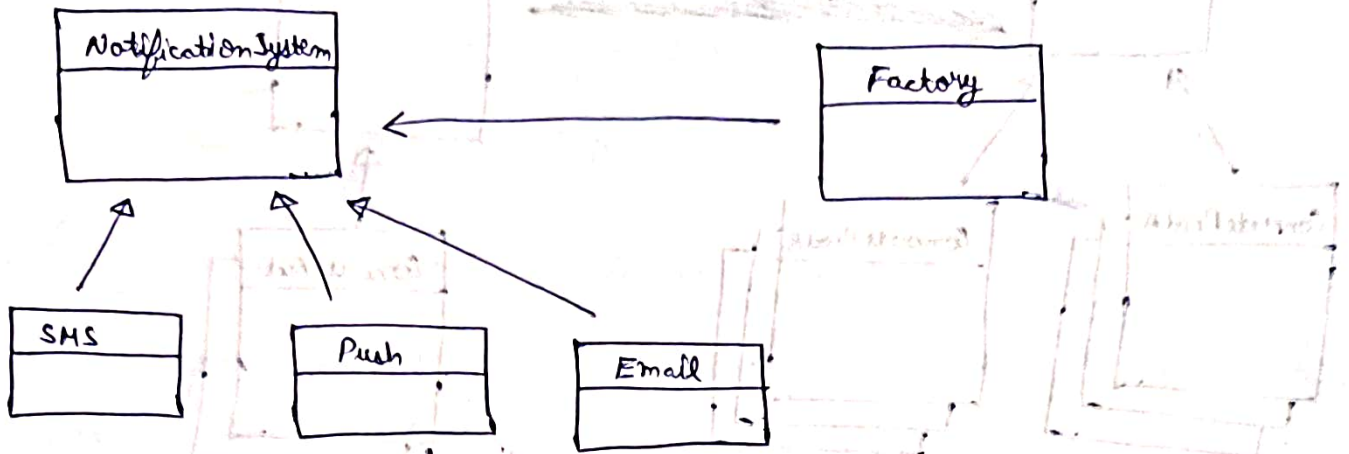
Abstract Factory Method

Provides an interface for creating families of related objects without specifying their concrete classes



Real world Application for Factory Design Pattern

① Notification System



Can be built by Strategy Design Pattern as well
(Which to use \Rightarrow Depends on use case)

