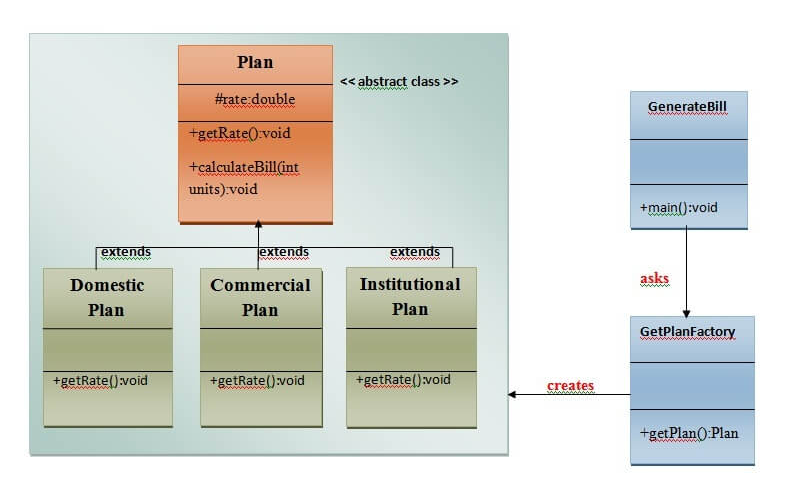
# Factory Pattern:

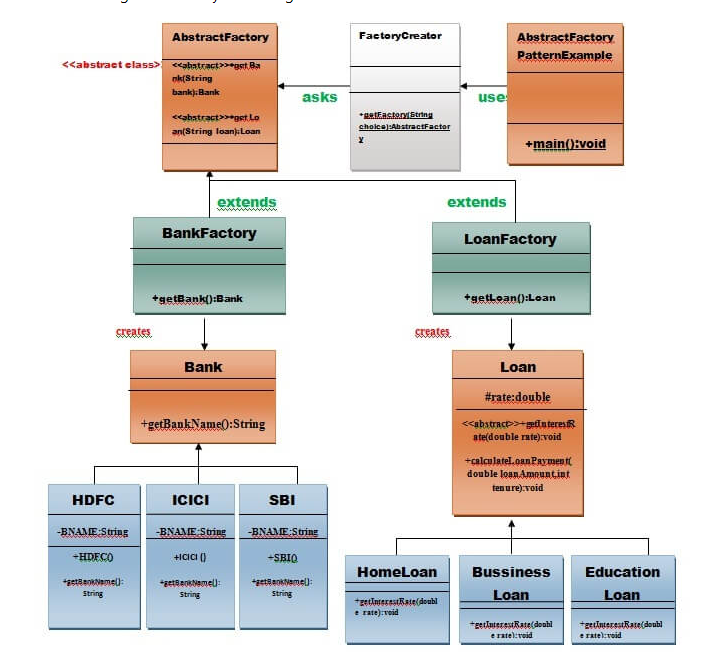
* **Official Definition:** It defines an interface or abstract class for creating an object but let the subclasses decide which class to instantiate.
* In other words, subclasses are responsible to create the instance of the class.



# Abstract Factory Pattern:

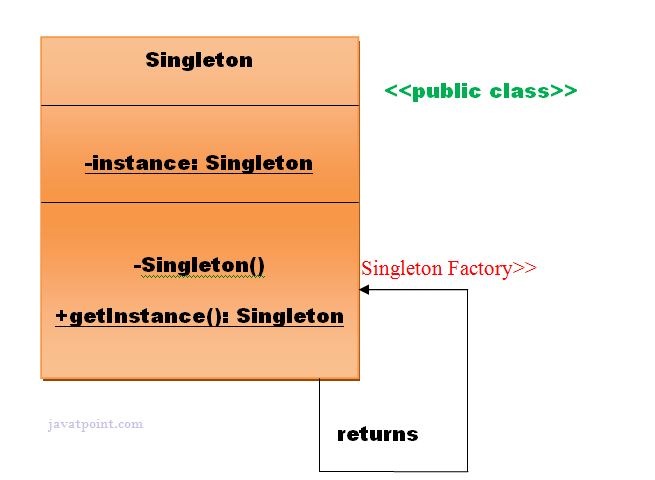
* Abstract Factory Pattern is one level higher than the Factory Pattern.
* **Official Definition:** It define an interface or abstract class for creating families of related (or dependent) objects but without specifying their concrete sub-classes.
* In short, main method calls **FactoryCreator.getFactory** method (which returns a new instance of Bank or Loan factory).

Those factory classes have a getLoan or getBank method (depending on which object) and they can return again different object instance.



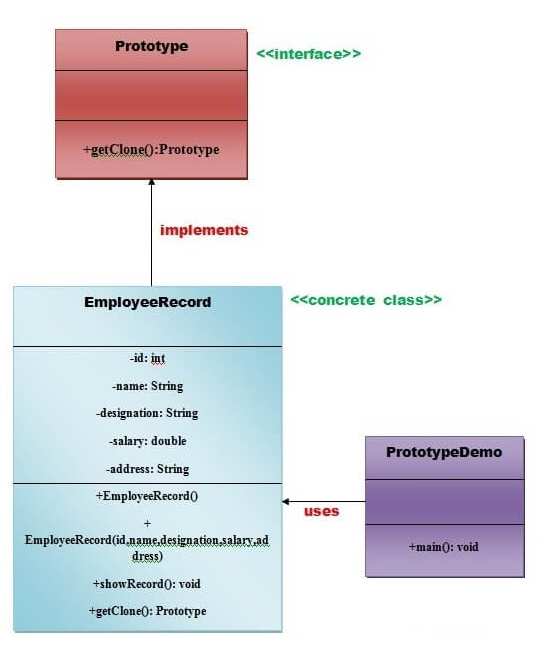
# Singleton Pattern

* **Official Definition:** define a class that has only one instance and provides a global point of access to it
* To create the singleton class, we need to have static member of class, private constructor and static factory method.



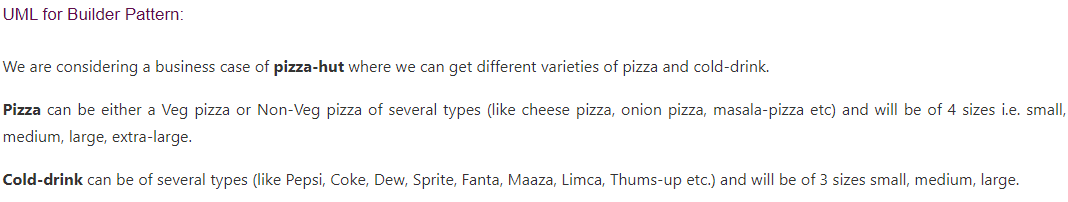
# Prototype Pattern

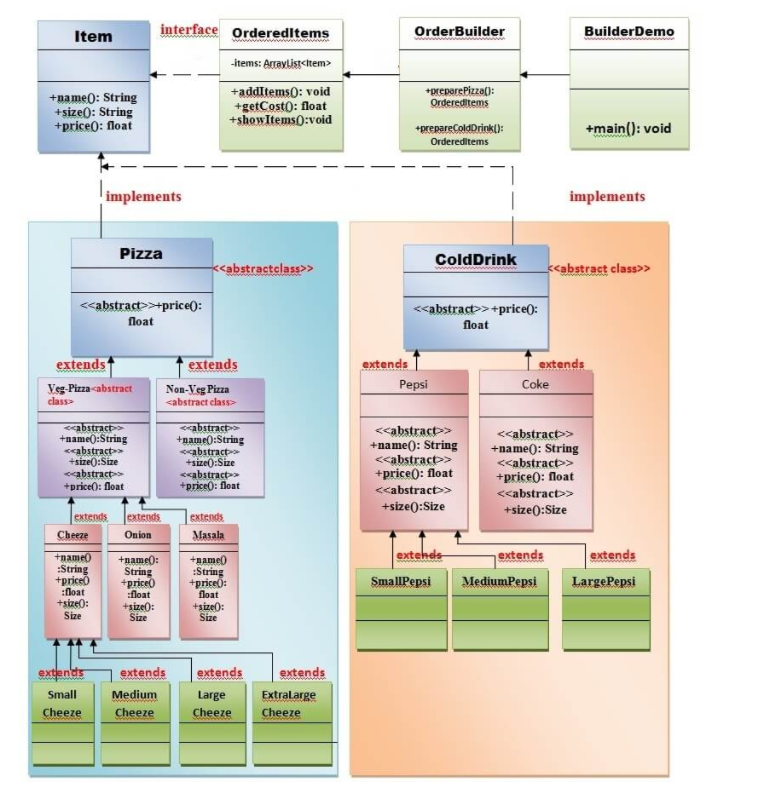
* **Official Definition:** Cloning of an existing object instead of creating new one and can also be customized as per the requirement.



# Builder Pattern

* **Official Definition:** construct a complex object from simple objects using step-by-step approach





# Object Pool Pattern

Yet to study

