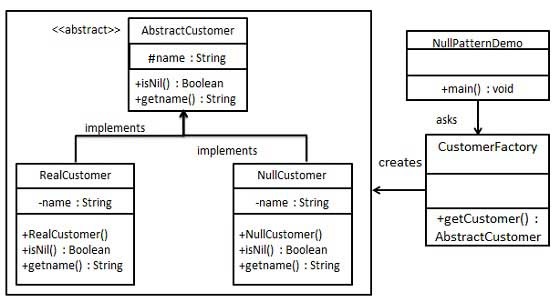
Lab ASSIGNMENT

**Behavioral Patterns: Null Object Pattern**

**Null-Object Pattern:**

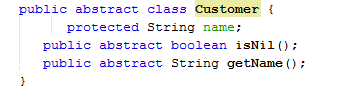
In Null Object pattern, a null object replaces check of NULL object instance. Instead of putting if check for a null value, Null Object reflects a do nothing relationship. Such Null object can also be used to provide default behavior in case data is not available.

In Null Object pattern, we create an abstract class specifying various operations to be done, concrete classes extending this class and a null object class providing do nothing implementation of this class and will be used seamlessly where we need to check null value.

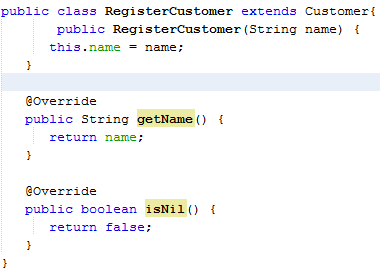


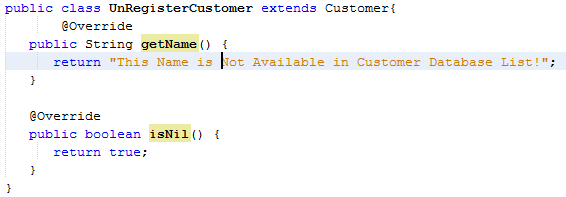
**SOURCE CODE:**

Create an abstract class.

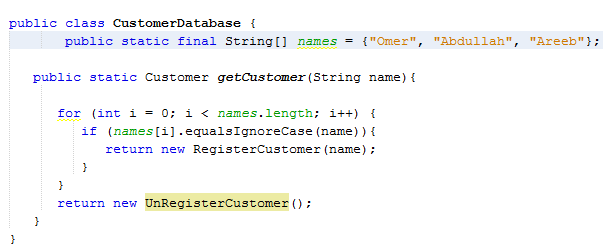
****

Create concrete classes extending the above class.

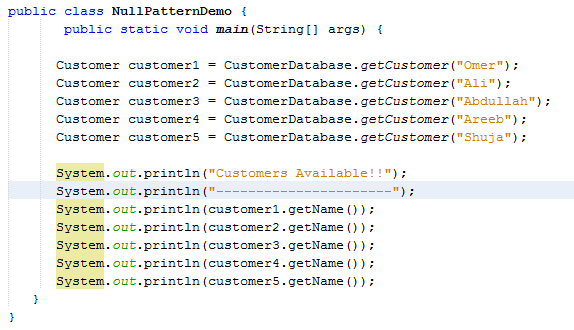
****

****

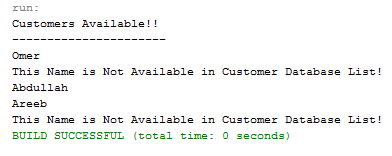
Create *CustomerDatabase* Class.

****

Use the *CustomerDatabase* to get either *RegisterCustomer* or *UnRegisterCustomer* objects based on the name of customer passed to it.

****

**OUTPUT:**

****