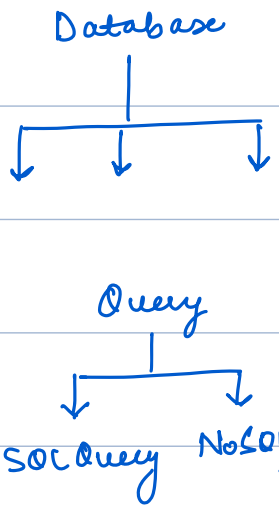
Creational

Factory
- simple
- factory method
- abstract factory

```
User Service {

    Database db = ____;   // type ↙ mysql / mongodb / pgsql

    createUser() {
        if( db.type == "mysql")
            Query q = new SQLQuery();
        else if( db.type = "____")
            ____
        else if( ____ )
            ____
    }
}
```

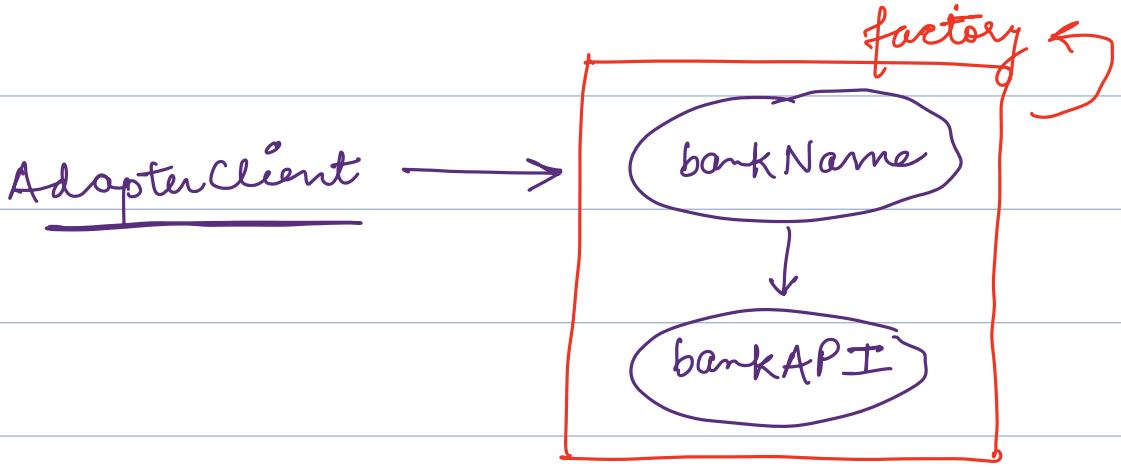


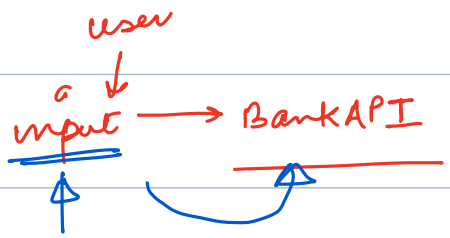
SRP, OCP

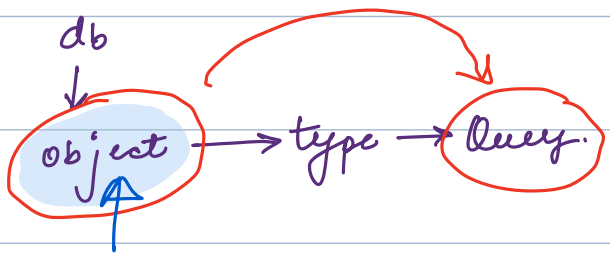
```
class QueryFactory {

    static Query getQueryByDBType(dbType) {
        if( db.type == "mysql")
            Query q = new SQLQuery();
        else if( db.type = "____")
            ____
        else if( ____ )
            ____
    }
}
```

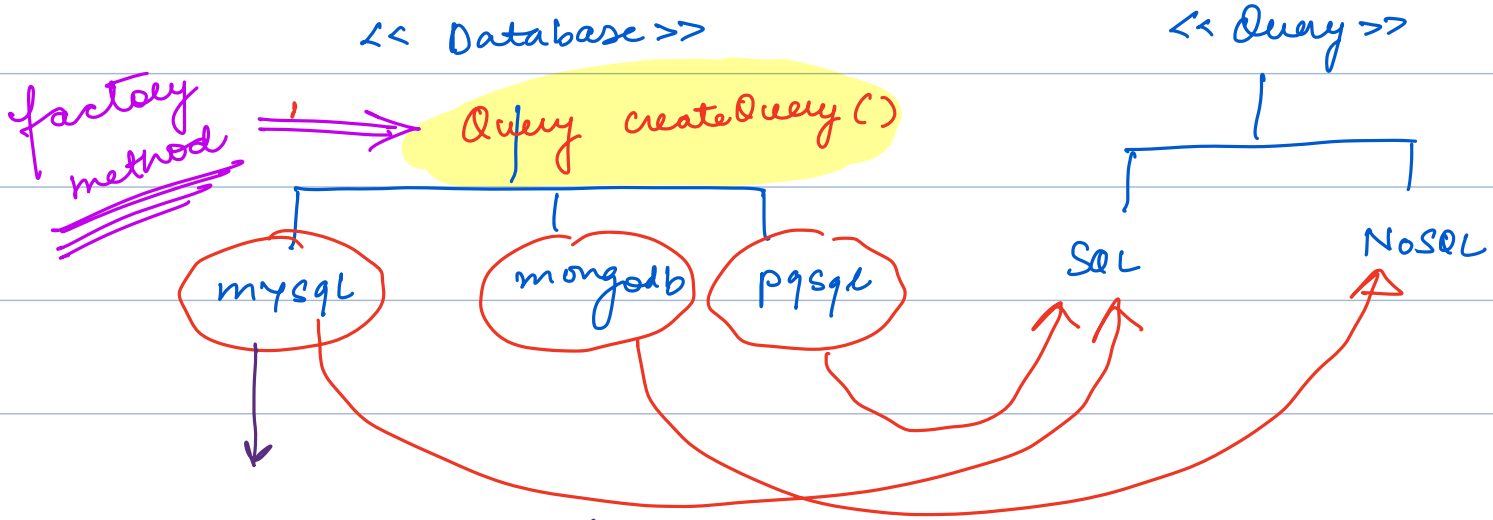
on the basis of some cond" decide which obj to produce!

```
createUser() {
    Query q = getQueryByDBType(dbType);
}
```










```
Query createQuery() {
    new SQLQuery();
}
```

User Service {

Database db = _____ ;

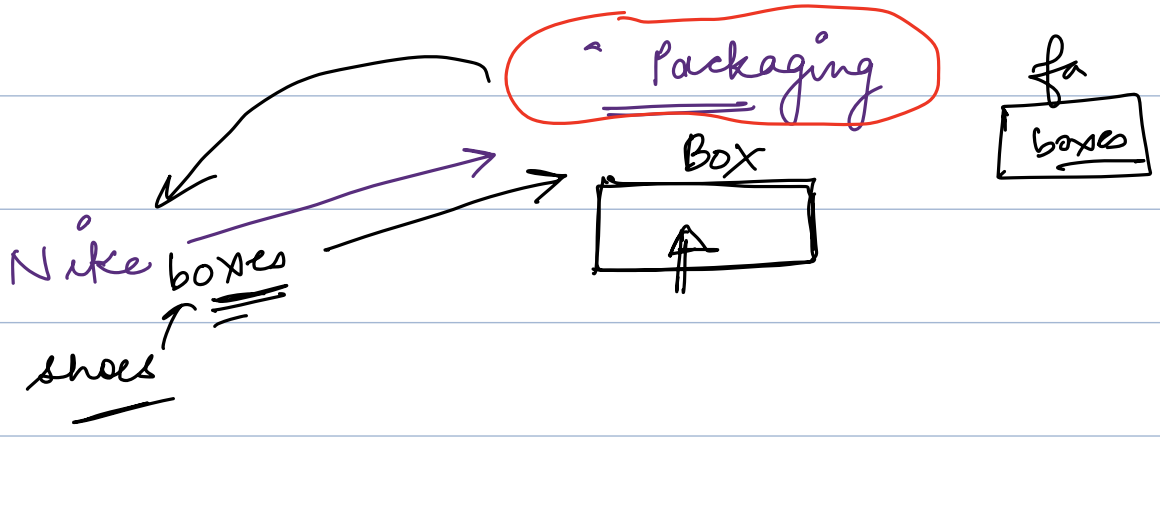
type

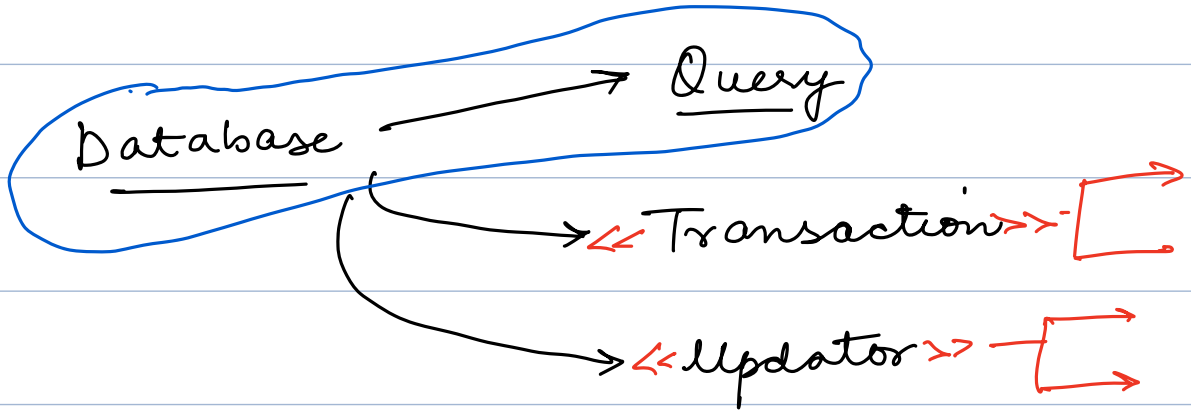
createUser() {

Query q = db.createQuery();

}

}





<< Database >>

createDbFact()

createQuery()

createTransn().

createUpdator()

methodA()
methodB()
methodC()

3 factory methods

<< DatabaseFactory >>

createQuery()

createTransn().

createUpdator()

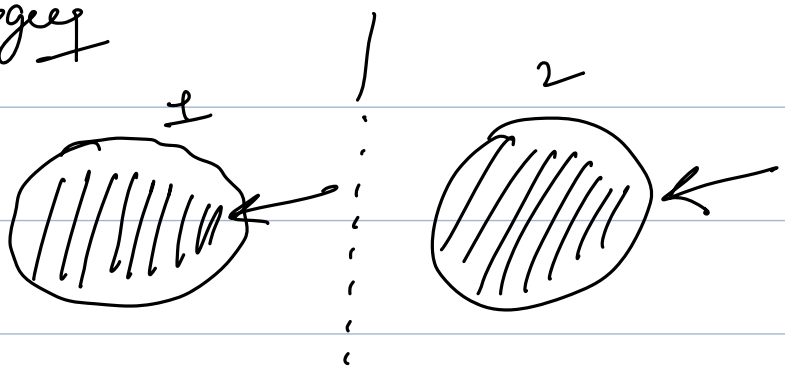
mysql

mysqlFact

mongodbFact

pgFact

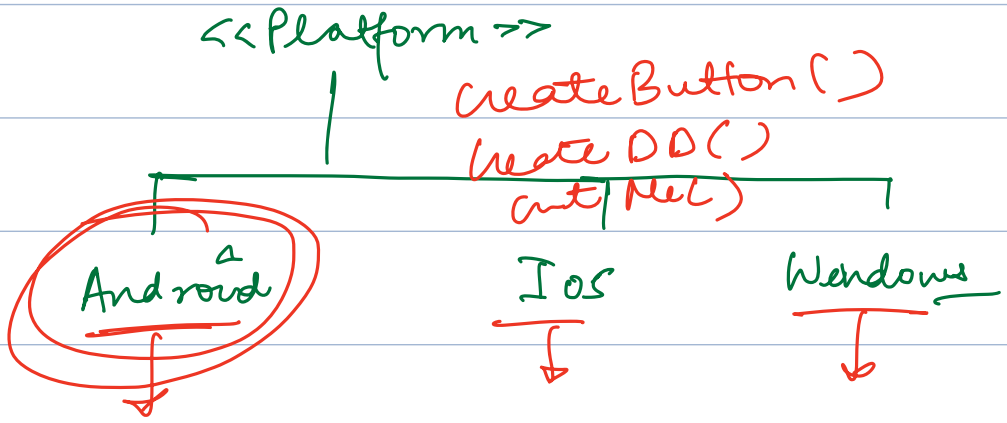
Interface segregation

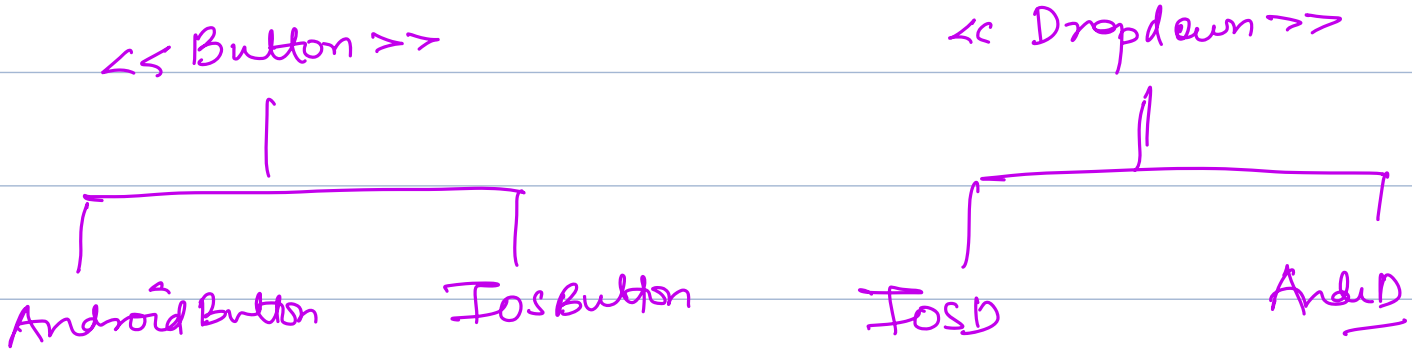


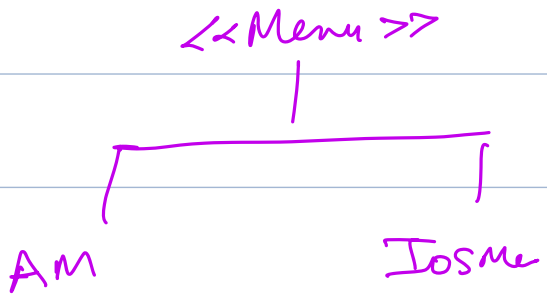
```
UserService {

    Database db = _____ ;

    DatabaseFactory dbfact = db.createfact();

    createUser() {

        Query q = dbfact.createQuery();
        _____ = dbfact.createTran();

    }

}
```

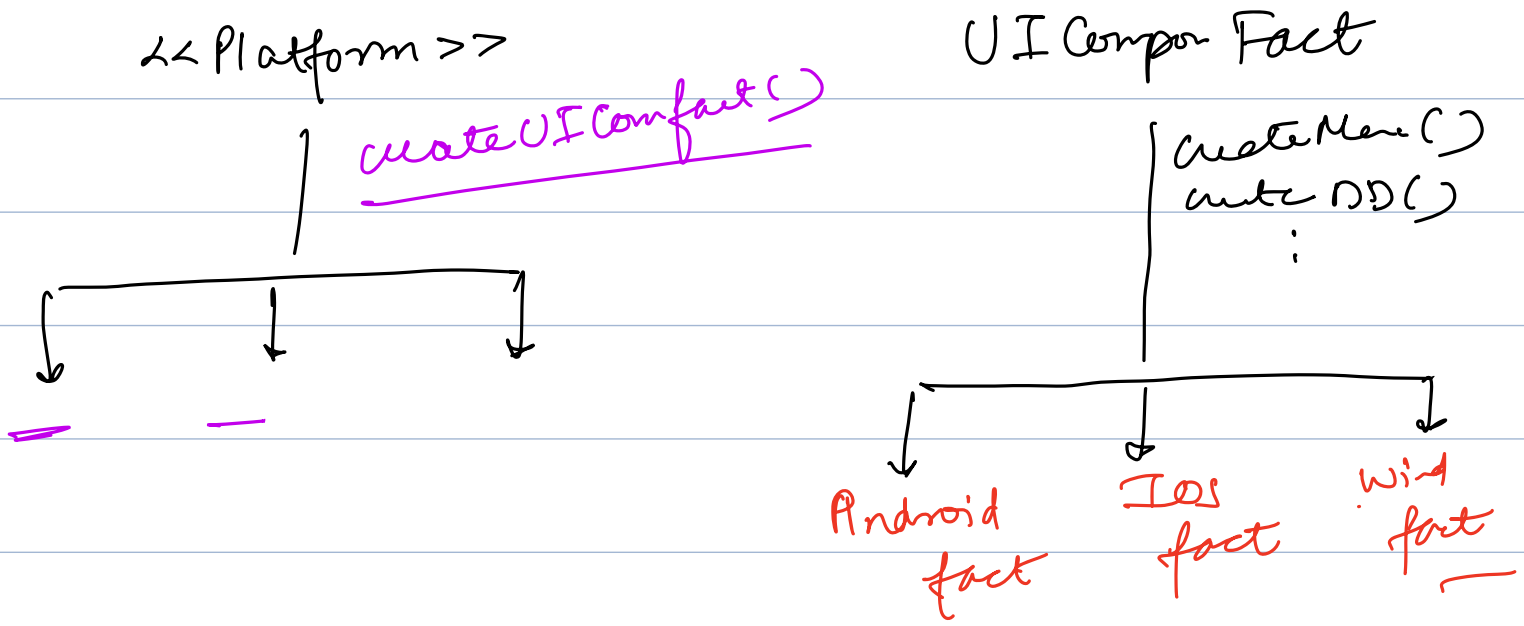
Flutter / React Native :

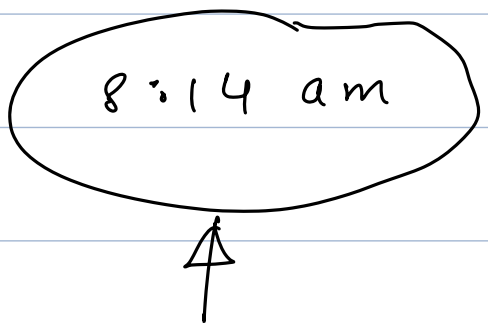
cross platform UI framework.








<<Platform>>
createUIComfact()
UI Compon Fact
createMen()
createDD()
Android fact
IOS fact
Wind fact


8:14 am