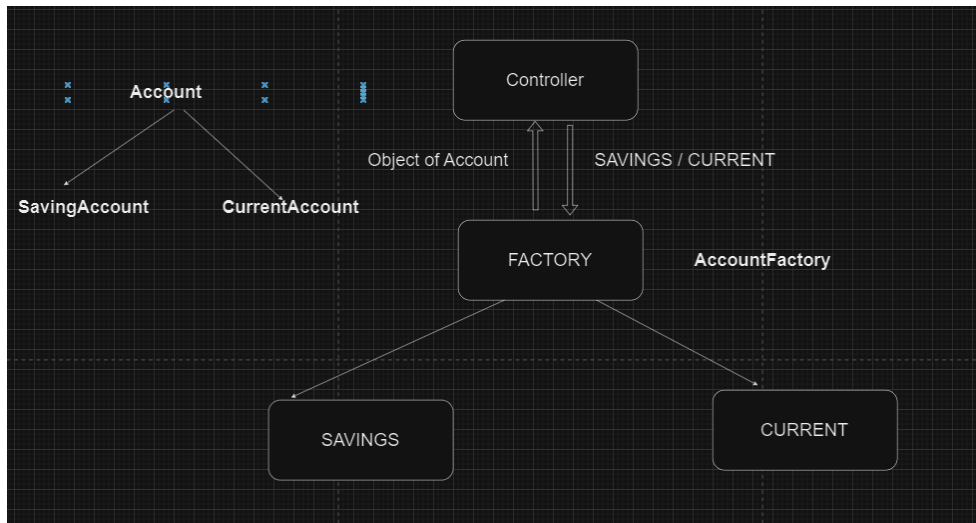


Creational Design Pattern: Factory



In above diagram, as you can see, Controller does not directly reach out to feature classes like Savings and Current. Instead it goes to factory and fetches the object of whatever class it needs.

So this is single point of Object fetch for controller which makes it convenient.

This is how controller reaches out to factory

```
AccountFactory factory = new AccountFactory(); //reaching out to factory
Account account = factory.getAccount(AccountType.SAVINGS); //fetching object of SAVINGS
System.out.println(account.getAccountDetails());

account = factory.getAccount(AccountType.CURRENT); //fetching object of CURRENT
System.out.println(account.getAccountDetails());
```

And this is how Factory is implemented:

```
public class AccountFactory {

    public Account getAccount(AccountType type) {
        if(type.equals(AccountType.SAVINGS))
            return new SavingAccount();
        if(type.equals(AccountType.CURRENT))
            return new CurrentAccount();

        return null;
    }
}
```