L20 :- Template Design pattern.
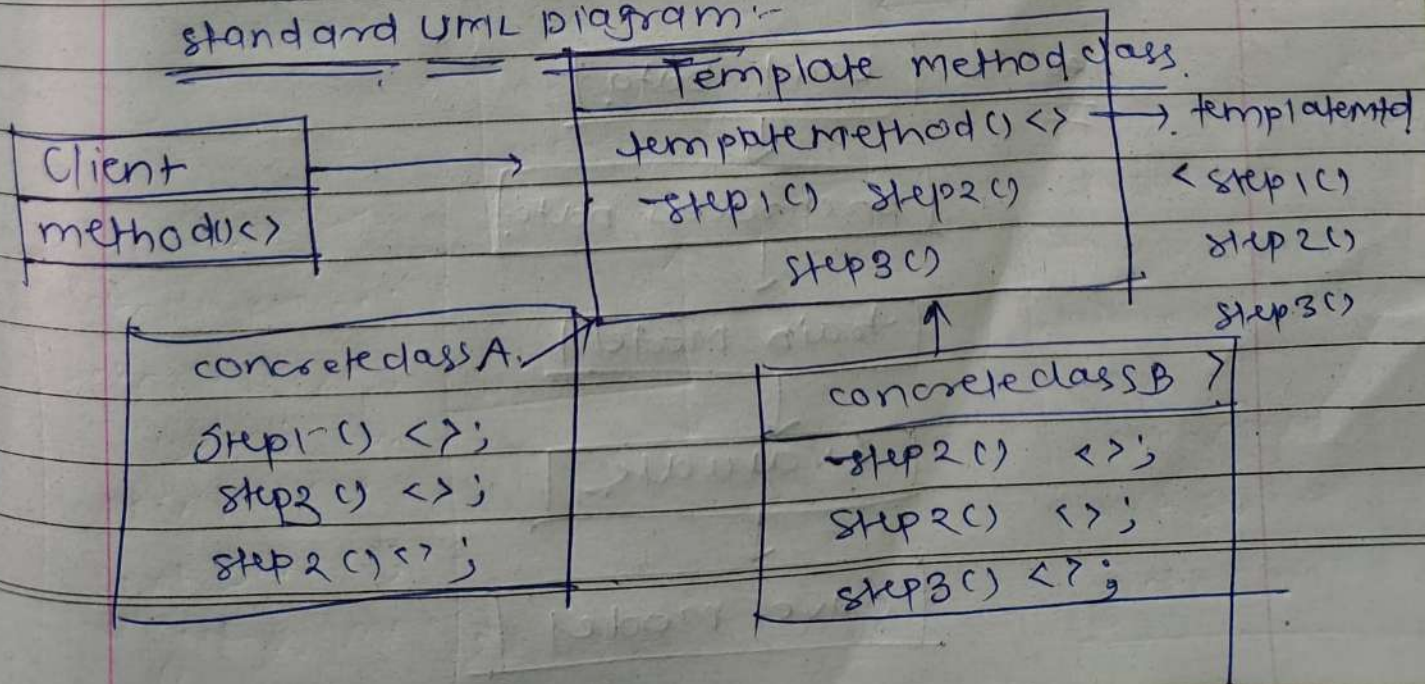
- It's not a commonly general used patterns it's a use case specific pattern. It's used in the problem where tasks will be executed sequentially. It solves the problem of pipleline.

Standard Defination

Template Design pattern is the skeleton of algorithm for an operation, defining some steps to subclasses. Template method .DP states the Let subclasses redefine certain steps of an algorithm without changing the algorithm structure.
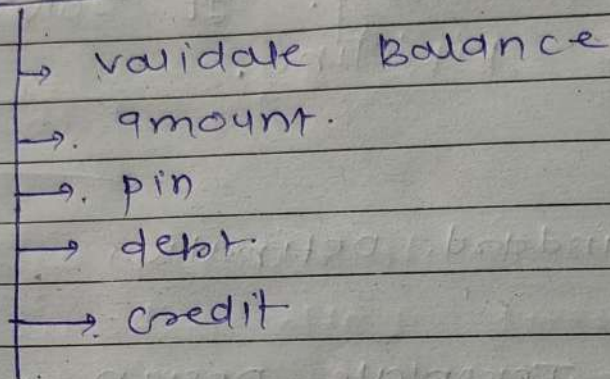
standard UML Diagram :-
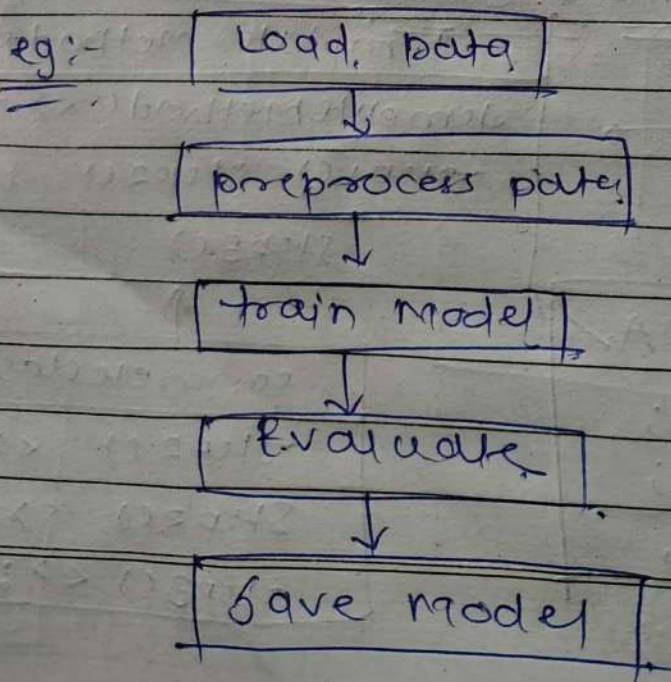

Template method class.
templatemethod() <> → templatemtd
-step1() step2() <step1()
step3() step2()
step3()

Client
method()()<>

concreteclass A.
step1() <>;
step3() <>;
step2()()<?;

concreteclass B >
step2() <>;
step2() <>;
step3() <?;

Real world use case

used in sequence scenario's when we want to follow order or sequence op operation.

ex:- ① UPI payment
- → validate Balance
- →. amount.
- →. pin
- → debit
- → credit

Let's Assume, A data scientist have to train models. He will follow seq. of steps to follow the train the models. this seq. of steps (pipeline) has same steps for all models.

eg:-

```
Load Data
    ↓
preprocess data
    ↓
train Model
    ↓
Evaluate
    ↓
Save model
```

New
prep
load
sav
eval
prep

- Here we don't want developer to make any mistake by skipping the step or alternate sequence. Hence already developed sequence template we have to use.

UML Diagram for model training.

- First declare the ~~this~~ abstract class having multiple child class mtds. abstract class have ~~di~~ template method which decide the sequence of method execution. no child class will override Template method.

- Template mtd gives us the power to. keep execution seq. constant.

Client
ModelTrainer mt;
m();
mt → template mtd();

<>
modeTrainer
load(path);
preprocess();
train()
evaluate();
save();
can overide

Templatemtd() {
load (path);
preprocess();
train()
evaluate()
save();
}

NeuralN/wmodel
prepare()
load()
save()
evaluate()
preprocess().

Decisiontreemodel
prepare()
save()
load()
preprocess()
evaluate()