Design patterns:

1. Factory design pattern:

***INTERFACE:***

**public** **interface** OperatingSystem {

**void** specification();

}

***1st class:***

**public** **class** Android **implements** OperatingSystem {

@Override

**public** **void** specification() {

System.***out***.println("Most powerful OS!! ");

}

}

***2nd class:***

**public** **class** Ios **implements** OperatingSystem {

@Override

**public** **void** specification() {

System.***out***.println("Most secured OS!!");

}

}

***3rd class:***

**public** **class** Windows **implements** OperatingSystem{

@Override

**public** **void** specification() {

System.***out***.println("I am not much in use nowadays!!");

}

}

***FACTORY CLASS:***

**public** **class** OperatingSystemFactory {

**public** OperatingSystem getOS(String string) {

**if** (string.equalsIgnoreCase("Powerful")) {

**return** **new** Android();

} **else** **if** (string.equalsIgnoreCase("Secured")) {

**return** **new** Ios();

} **else** {

**return** **new** Windows();

}

}

}

***MAIN METHOD:***

public class User {

public static void main(String[] args) {

OperatingSystemFactory system = new OperatingSystemFactory();

OperatingSystem operatingSystem = system.getOS("secured");

operatingSystem.specification();

}

}

1. SINGLETON DESIGN PATTERN:

***MOVIES CLASS:***

**public** **class** Movie {

**int** tickets = 50;

**public** **void** bookTickets(**int** noOfTickets) {

System.***out***.println("Total no. of seats available in the movie theatre: " + tickets);

tickets -= noOfTickets;

System.***out***.println(noOfTickets + " tickets successfully booked.");

System.***out***.println("No. of tickets available: " + tickets);

}

}

***MAIN CLASS (CUSTOMER):***

**public** **class** Customer {

**public** **static** **void** main(String[] args) {

Movie movie = **new** Movie();

movie.bookTickets(6);

Movie movie2 = **new** Movie();

movie2.bookTickets(9);

}

}

SINGLETON CLASS IN THE SAME MOVIE CLASS:

**public** **class** Movie {

//Private constructor

**private** Movie() {

}

//static reference of the class

**private** **static** Movie *movie* ;

**static** **int** *tickets* = 50;

//Public static method to get the instance

**public** **static** Movie bookTickets(**int** noOfTickets) {

**if**(*movie* == **null**) {

*movie* = **new** Movie();

}

System.***out***.println("Total no. of seats available in the movie theatre: " + *tickets*);

*tickets* -= noOfTickets;

System.***out***.println(noOfTickets + " tickets successfully booked.");

System.***out***.println("No. of tickets available: " + *tickets*);

**return** *movie*;

}

}

CUSTOMER (MAIN CLASS FOR SINGLETON)

**public** **class** Customer {

**public** **static** **void** main(String[] args) {

Movie movie = Movie.*bookTickets*(7);

Movie movie2 = Movie.*bookTickets*(9);

//To show the addresses

System.***out***.println(movie);

System.***out***.println(movie2);

}

}

1. Builder pattern:

***Phone class:***

**public** **class** Phone {

**private** String os;

**private** **int** ram;

**private** **double** screenSize;

**private** String processor;

**private** **int** battery;

**public** Phone(String os, **int** ram, **double** screenSize, String processor, **int** battery) {

**super**();

**this**.os = os;

**this**.ram = ram;

**this**.screenSize = screenSize;

**this**.processor = processor;

**this**.battery = battery;

}

@Override

**public** String toString() {

**return** "Phone [os=" + os + ", ram=" + ram + ", screenSize=" + screenSize + ", processor=" + processor

+ ", battery=" + battery + "]";

}

}

***SHOP CLASS: (MAIN CLASS)***

**public** **class** Shop {

**public** **static** **void** main(String[] args) {

Phone phone = **new** Phone("Android", 6 , 5.5 , "Snapdragon", 4000);

System.***out***.println(phone);

//2nd part after writing the PhoneBuilder class

// Phone builder = new PhoneBuilder().setBattery(4100).setOs("Android").getPhone();

//

// System.out.println(builder);

}

}

***PHONEBUILDER CLASS:***

**public** **class** PhoneBuilder {

**private** String os;

**private** **int** ram;

**private** **double** screenSize;

**private** String processor;

**private** **int** battery;

**public** PhoneBuilder setOs(String os) {

**this**.os = os;

**return** **this**;

}

**public** PhoneBuilder setRam(**int** ram) {

**this**.ram = ram;

**return** **this**;

}

**public** PhoneBuilder setScreenSize(**double** screenSize) {

**this**.screenSize = screenSize;

**return** **this**;

}

**public** PhoneBuilder setProcessor(String processor) {

**this**.processor = processor;

**return** **this**;

}

**public** PhoneBuilder setBattery(**int** battery) {

**this**.battery = battery;

**return** **this**;

}

@Override

**public** String toString() {

**return** "PhoneBuilder [os=" + os + ", ram=" + ram + ", screenSize=" + screenSize + ", processor=" + processor

+ ", battery=" + battery + "]";

}

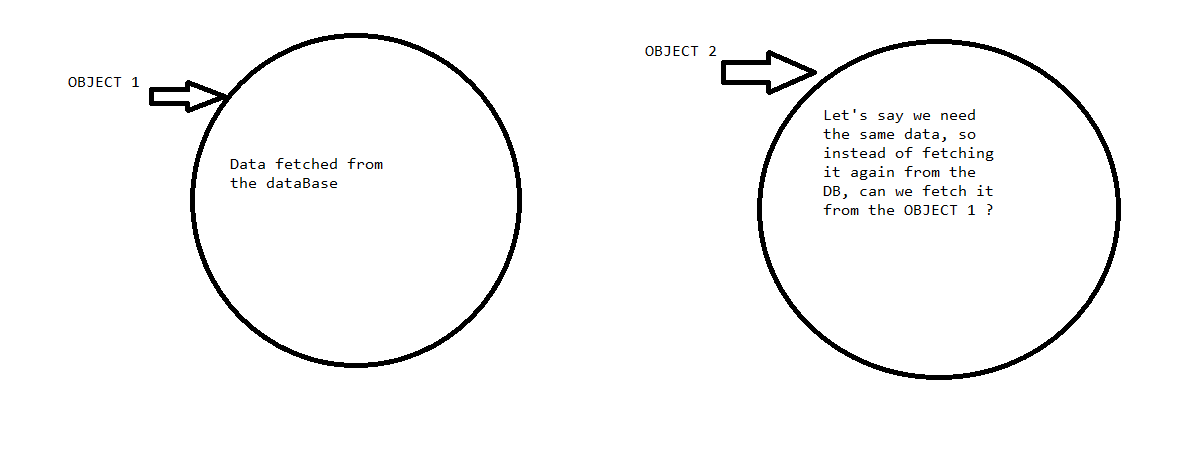
**public** Phone getPhone() {

**return** **new** Phone(os, ram, screenSize, processor, battery);

}

}

1. PROTOTYPE DESIGN PATTERN:



***BOOK CLASS:***

**public** **class** Book {

**private** **int** bid;

**private** String bname;

**public** **int** getBid() {

**return** bid;

}

**public** **void** setBid(**int** bid) {

**this**.bid = bid;

}

**public** String getBname() {

**return** bname;

}

**public** **void** setBname(String bname) {

**this**.bname = bname;

}

@Override

**public** String toString() {

**return** "Book [bid=" + bid + ", bname=" + bname + "]";

}

}

***BOOKSHOP CLASS:***

**public** **class** BookShop **implements** Cloneable {

**private** String shopName;

**private** List<Book> books = **new** ArrayList<Book>();

**public** String getShopName() {

**return** shopName;

}

**public** **void** setShopName(String shopName) {

**this**.shopName = shopName;

}

**public** List<Book> getBooks() {

**return** books;

}

**public** **void** setBooks(List<Book> books) {

**this**.books = books;

}

@Override

**public** String toString() {

**return** "BookShop [shopName=" + shopName + ", books=" + books + "]";

}

**public** **void** loadBooks() {

**for** (**int** i = 1; i <= 5; i++) {

Book book = **new** Book();

book.setBid(i);

book.setBname("Book " + i);

getBooks().add(book);

}

}

//SHALLOW CLONING

// @Override

// public Object clone() throws CloneNotSupportedException {

// return super.clone();

// }

//DEEP CLONING

@Override

**public** Object clone() **throws** CloneNotSupportedException {

BookShop shop = **new** BookShop();

**for** (Book book : **this**.getBooks()) {

shop.getBooks().add(book);

}

**return** shop;

}

}

***MAINCLASS:***

**public** **class** MainClass {

**public** **static** **void** main(String[] args) **throws** CloneNotSupportedException {

BookShop bookShop = **new** BookShop();

bookShop.setShopName("Novelty");

bookShop.loadBooks();

BookShop bookShop1 = (BookShop) bookShop.clone();

bookShop.getBooks().remove(2);

bookShop1.setShopName("Orchid book house");

System.***out***.println(bookShop);

System.***out***.println("-----------------");

System.***out***.println(bookShop1);

}

}

4) Template method:

***ABSTRACT CLASS:***

**public** **abstract** **class** Game {

**public** **abstract** **void** init();

**public** **abstract** **void** start();

**public** **abstract** **void** end();

**public** **void** play() {

init();

start();

end();

}

}

***1st IMPL CLASS***

**public** **class** CallOfDuty **extends** Game {

@Override

**public** **void** init() {

System.***out***.println("Loading ....");

}

@Override

**public** **void** start() {

System.***out***.println("Call of duty playing!!");

}

@Override

**public** **void** end() {

System.***out***.println("Call of Duty ended!!");

}

}

***2nd IMPL CLASS***

**public** **class** ClashOfClans **extends** Game {

@Override

**public** **void** init() {

System.***out***.println("Loading ....");

}

@Override

**public** **void** start() {

System.***out***.println("Welcome back to your village Chief!!");

}

@Override

**public** **void** end() {

System.***out***.println("Come back Chief! Your village needs you!");

}

}

*MAIN METHOD*

**public** **class** MainClassForTemplate {

**public** **static** **void** main(String[] args) {

ClashOfClans clashOfClans= **new** ClashOfClans();

clashOfClans.play();

System.***out***.println("-----------------------");

CallOfDuty callOfDuty= **new** CallOfDuty();

callOfDuty.play();

}

}