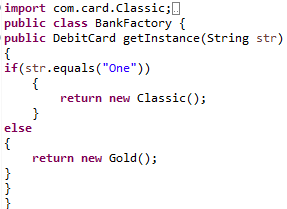
# **Task 1 – Debit Card (Factory Pattern)**

### BankFactory.java



### Client.Java

**import** com.card.DebitCard;

**import** com.card.Gold;

**import** com.card.Classic;

**public** **class** Client {

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

{

BankFactory BK = **new** BankFactory();

DebitCard obj = BK.getInstance("One");

obj.getCardType();

obj.getCardLimit();

}

}

### classic.java

**package** com.card;

**public** **class** Classic **implements** DebitCard {

@Override

**public** **void** getCardType()

{

System.***out***.println("Type: Classic");

}

@Override

**public** **void** getCardLimit()

{

System.***out***.println("Limit: 50,000");

}

}

### Gold.java

**package** com.card;

**public** **class** Gold **implements** DebitCard{

@Override

**public** **void** getCardType()

{

System.***out***.println("Type: Gold");

}

@Override

**public** **void** getCardLimit()

{

System.***out***.println("Limit: 100,000");

}}

### DebitCard.java

**package** com.card;

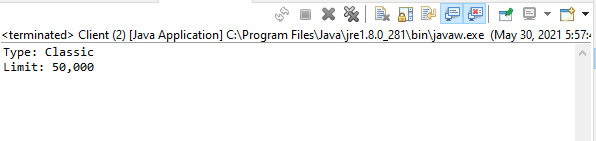
**public** **interface** DebitCard {

**void** getCardType();

**void** getCardLimit();

}

### Output



# **Task 2 – Laptop (Factory Pattern)**

### Client.java

**import** com.LAPTOP.Dell;

**import** com.LAPTOP.HP;

**import** com.LAPTOP.Laptop;

**import** com.LAPTOP.Lenovo;

**public** **class** Client {

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

{

LaptopFactory LF = **new** LaptopFactory();

Laptop obj = LF.getInstance("HP");

obj.Performance();

obj.Specs();

obj.Price();

}

}

### LaptopFactory.java

**import** com.LAPTOP.Dell;

**import** com.LAPTOP.HP;

**import** com.LAPTOP.Laptop;

**import** com.LAPTOP.Lenovo;

**public** **class** LaptopFactory {

**public** Laptop getInstance(String str)

{

**if** (str.equals("Dell")) {

**return** **new** Dell();

}

**else** **if** (str.equals("HP")) {

**return** **new** HP();

}

**else**

**return** **new** Lenovo();

}}

### Dell.java

**package** com.LAPTOP;

**public** **class** Dell **implements** Laptop{

@Override

**public** **void** Performance(){

System.***out***.println("Performance: Very Fast.");}

@Override

**public** **void** Specs(){

System.***out***.println("Specification: Core i5, 4GB RAM.");}

@Override

**public** **void** Price(){

System.***out***.println("Price: 98,000.");}

}

### Lenovo.Java

**package** com.LAPTOP;

**public** **class** Lenovo **implements** Laptop{

@Override

**public** **void** Performance()

{

System.***out***.println("Performance: Great.");

}

@Override

**public** **void** Specs()

{

System.***out***.println("Specification: Core i3, 8GB RAM.");

}

@Override

**public** **void** Price()

{

System.***out***.println("Price: 80,000.");

}

}

### HP.Java

**package** com.LAPTOP;

**public** **class** HP **implements** Laptop{

@Override

**public** **void** Performance(){

System.***out***.println("Performance: Amazing.");}

@Override

**public** **void** Specs(){

System.***out***.println("Specification: Core i5, 8GB RAM.");}

@Override

**public** **void** Price(){

System.***out***.println("Price: 104,000.");

}}

### Laptop.java

**package** com.LAPTOP;

**public** **interface** Laptop {

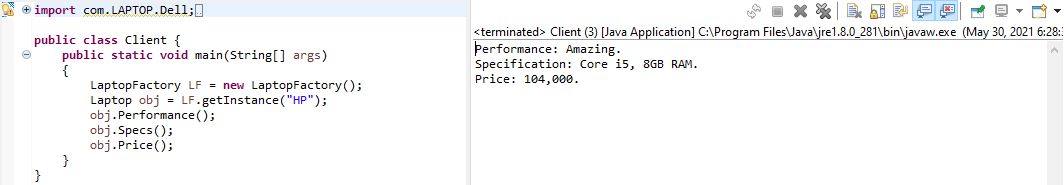
**void** Performance();

**void** Specs();

**void** Price();

}

### Output



# Task 3 – Rar(s) (Adapter Pattern)

### Client.java

**public** **class** Client {

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

JavaCodeRar rarFile = **new** JavaCodeRar();

rarFile.OpenCode();

rarFile.UpdateCode();

JavaCodeTar tarFile = **new** JavaCodeTar();

tarFile.UnlockCode();

tarFile.ChangeCode();

ZipCodeExtractor zipFile= **new** FileAdapter(tarFile);

zipFile.OpenCode();

zipFile.UpdateCode();}

}

### JavaCoderar.java

**public** **class** JavaCodeRar **implements** ZipCodeExtractor

{

**public** **void** OpenCode(){

System.***out***.println("Opening Rar........");}

**public** **void** UpdateCode(){

System.***out***.println("Updating Rar........\n");}

}

### JavaCodeTar.java

**public** **class** JavaCodeTar

{

**public** **void** UnlockCode(){

System.***out***.println("Unlocking Tar........");}

**public** **void** ChangeCode(){

System.***out***.println("Changing Tar........");}

}

### FileAdapter.java

**public** **class** FileAdapter **implements** ZipCodeExtractor

{

JavaCodeTar tar;

**public** FileAdapter(JavaCodeTar tar){

**this**.tar=tar;}

**public** **void** OpenCode() {

tar.UnlockCode(); }

**public** **void** UpdateCode(){

tar.ChangeCode();}

}

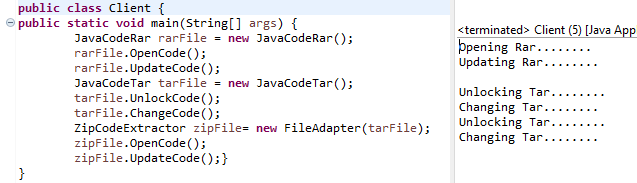
### ZipcodeExtractor.java

**public** **interface** ZipCodeExtractor {

**public** **void** OpenCode();

**public** **void** UpdateCode(); }

### Output



# Task 4 – Image Viewer (Adapter Pattern)

### Client.java

**public** **class** Client {

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

JPGFiles jpg = **new** JPGFiles();

jpg.JPG();

jpg.open();

PNGFiles png=**new** PNGFiles();

png.PNG();

png.open();

GIF gif= **new** GIF();

gif.open();

ImageViewer iv = **new** GIFAdapter(gif);

iv.open();

}

}

### Gif.java

**public** **class** GIF

{

**public** GIF(){

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

**public** **void** open(){

System.***out***.println("GIF Opened..........");}

}

### GifAdapter.java

**public** **class** GIFAdapter **implements** ImageViewer

{

GIF gif=**new** GIF();

**public** GIFAdapter(GIF gif){

**this**.gif=gif;}

**public** **void** open(){

System.***out***.println("GIF Opened using Adapter..........");}

}

### ImageViewer.java

**public** **interface** ImageViewer {

**public** **void** open();}

### JPGFiles.java

**public** **class** JPGFiles **implements** ImageViewer

{

**public** **void** JPG(){

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

**public** **void** open(){

System.***out***.println("JPG Opened..........");}

}

### PNGFiles.java

**public** **class** PNGFiles **implements** ImageViewer

{

**public** **void** PNG(){

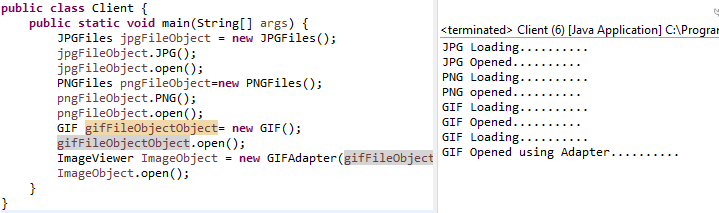
System.***out***.println("PNG Loading..........");}

**public** **void** open(){

System.***out***.println("PNG opened..........");}

}

### Output



# Task 5 – Subway Shop (Template Pattern)

### SubwayInterface.java

**public** **abstract** **class** SubwayInterface {

**public** SubwayInterface()

{

**super**();

}

**public** **abstract** **void** prepareSubway();

**public** **abstract** **void** selectBread();

**public** **abstract** **void** addingVeggies();

**public** **abstract** **void** addingChicken();

**public** **abstract** **void** addingSauces();

**public** **void** makeSubway()

{

prepareSubway();

selectBread();

addingVeggies();

addingChicken();

addingSauces();

}

@Override

**public** String toString() {

StringBuilder builder = **new** StringBuilder();

builder.append("Subway [Bread=").append("PitaBread").append(",Veggies=").append("No Veggies").append(",Chicken=").append("Roasted").append(", Sauces=").append("Jalpeno").append("]");

**return** builder.toString();

}

}

### Client.java

**public** **class** Client {

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

SubwayInterface sub = **new** Veg();

sub.makeSubway();

**if** (sub != **null**) {

System.***out***.println("Below Subway delievered: ");

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

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

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

}

}

}

### Veg.Java

**public** **class** Veg **extends** SubwayInterface {

@Override

**public** **void** prepareSubway()

{

System.***out***.println("Preparing Your Order.....");

}

@Override

**public** **void** selectBread()

{

System.***out***.println("Selecting Bread.....");

}

@Override

**public** **void** addingVeggies()

{

}

@Override

**public** **void** addingSauces()

{

System.***out***.println("Adding Sauces.....");

}

@Override

**public** **void** addingChicken()

{

System.***out***.println("Adding Roasted Chicken.....");

}

}

### NonVeg.Java

**public** **class** NonVeg **extends** SubwayInterface {

@Override

**public** **void** prepareSubway()

{

System.***out***.println("Preparing Your Order......");

}

@Override

**public** **void** selectBread()

{

System.***out***.println("Selecting Bread.....");

}

@Override

**public** **void** addingVeggies()

{

System.***out***.println("Adding Vegetables.....");

}

@Override

**public** **void** addingSauces()

{

System.***out***.println("Adding Sauces.....");

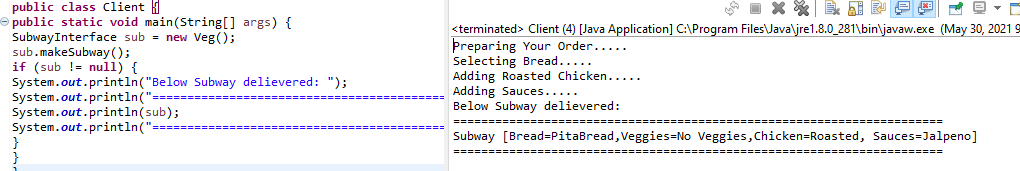
}

@Override

**public** **void** addingChicken()

{}}

### Output



# Singleton Pattern

### Prefrences.java

**public** **class** Preferences {

**private** **static** Preferences *obj* = **new** Preferences();

**public** String color;

**public** String carddesign;

**public** String music;

**private** Preferences() {}

**public** **static** Preferences getpreference() {

**return** *obj*;

**public** **void** SetMusic(String Music) {

music = Music;}

**public** **void** SetCardDesign(String CardDesign) {

carddesign = CardDesign;}

**public** **void** SetColour(String Colour) {

color = Colour;}

**public** **void** showpreferences() {

System.***out***.println("Colour ==> " + color);

System.***out***.println("Music ==> " + music);

System.***out***.println("Design ==> " + carddesign);}

### User1.Java

**public** **class** User1 {

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

Preferences object = Preferences.*getpreference*();

System.***out***.println("========== User No 1 ==========");

System.***out***.println("Object Address ==> " + object);

object.SetMusic("AURORA - Runaway");

object.SetCardDesign("Diamonds");

object.SetColour("Red");

object.showpreferences();

User2.*main*(args);}}

### User2.Java

**public** **class** User2 {

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

Preferences object2 = Preferences.*getpreference*();

System.***out***.println("========== User No 2 ==========");

System.***out***.println("Object Address ==> " + object2);

object2.SetMusic("MEMBA - For Aisha");

object2.SetCardDesign("Spades");

object2.SetColour("Black");

object2.showpreferences();}}

# façade pattern

### Chrome.Java

**public** **class** Chrome {

**void** getChromeDriver()

System.***out***.println("\nChrome Driver initaited.");

**void** generateHtmlReport()

System.***out***.println("Generating Chrome HTML Report.")

**void** generateJUnitReport()

System.***out***.println("Chrome JUnit Test Failed.....\n\n");}

### Firefox.Java

**public** **class** Firefox {

**void** getFirefoxDriver() {

System.***out***.println("\nFirefox Driver initaited.");}

**void** generateHtmlReport(){

System.***out***.println("Generating Firefox HTML Report.");}

**void** generateJUnitReport(){

System.***out***.println("Firefox JUnit Test Passed.....\n\n");}

### WebExplorerHelper.Java

**public** **class** WebExplorerHelper {

Chrome c = **new** Chrome();

Firefox f = **new** Firefox();

**void** generateReport(String selection){

**switch**(selection){

**case** "Chrome":{

c.getChromeDriver();

c.generateHtmlReport();

c.generateJUnitReport();

**break**;}

**case** "Firefox":{

f.getFirefoxDriver();

f.generateHtmlReport();

f.generateJUnitReport();

**break**;}

**default**:{

System.***out***.println("Invalid Browser")}

### FacadePatternExample.java

**public** **class** FacadePatternExample {

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

WebExplorerHelper web = **new** WebExplorerHelper();

web.generateReport("Chrome");

web.generateReport("Firefox");

}

}