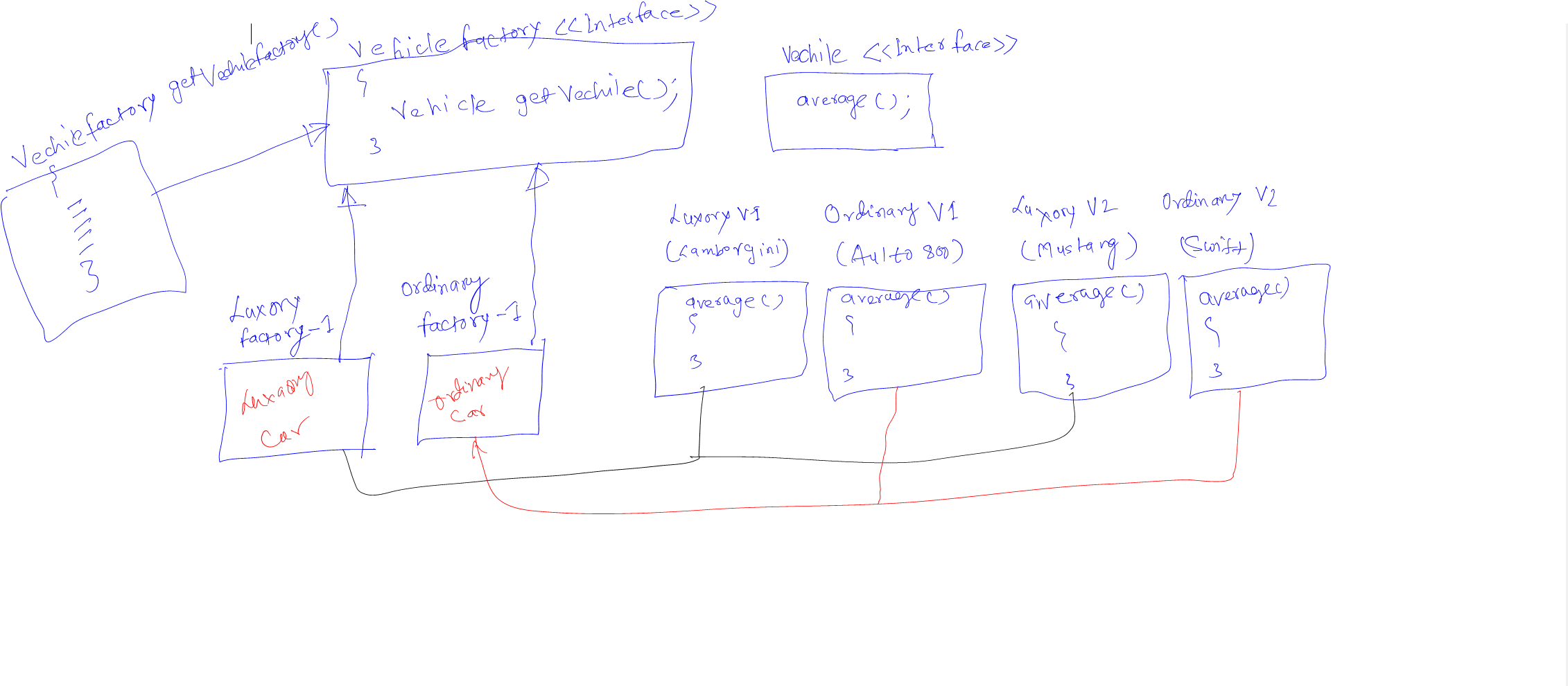
A white board with blue writing

Description automatically generated

package com;  
  
  
interface Shape  
{  
 public void draw();  
  
}  
class Circle implements Shape  
{  
  
 @Override  
 public void draw() {  
 System.*out*.println("Circle Shape Object Method");  
 }  
}  
class Square implements Shape  
{  
 @Override  
 public void draw() {  
 System.*out*.println("Square Shape Object Method");  
 }  
}  
  
class Rectangle implements Shape  
{  
 @Override  
 public void draw() {  
 System.*out*.println("Rectangle Shape Object Method");  
 }  
}  
  
  
  
 class Shapefactory {  
 public Shape getShape(String input)  
 {  
 switch (input)  
 {  
 case "Circle":  
 return new Circle();  
 case "Rectangle":  
 return new Rectangle();  
 case "Square":  
 return new Square();  
 default:  
 return null;  
 }  
 }  
}  
  
*//Dreiver Class*class MainDriver  
{  
 public static void main(String[] args) {  
 *// create the ShapeFactory Object* Shapefactory shapefactory = new Shapefactory();  
 *//get the Object based on The requirement  
 // use Interface Ref* Shape shapeObj = shapefactory.getShape("Rectangle");  
 *// based on the object received ,call the meethods of class* shapeObj.draw();  
 }  
}

Abstract Factory:



package com.factory;  
  
*// interface for all the vehicle*interface Vehicle  
{  
 public abstract void average();  
}  
  
*// implemented classes below*class LuxuryVehicleLamoborgini implements Vehicle  
{  
  
 @Override  
 public void average() {  
 System.*out*.println("LuxuryVehicleLamoborgini");  
 }  
}  
class LuxuryVehicleMustange implements Vehicle  
{  
 @Override  
 public void average() {  
 System.*out*.println("LuxuryVehicleMustange");  
 }  
}  
class OrdinaryVehicleAlto800 implements Vehicle  
{  
 @Override  
 public void average() {  
 System.*out*.println("OrdinaryVehicleAlto800");  
 }  
}  
  
class OrdinaryVehicleSwift implements Vehicle  
{  
 @Override  
 public void average() {  
 System.*out*.println("OrdinaryVehicleSwift");  
 }  
}  
 interface VehicleFactory {  
 Vehicle getVehicle(String name);  
  
}  
  
class LuxuryFactory implements VehicleFactory  
{  
 *//create Vehicle of Luxury on demand* @Override  
 public Vehicle getVehicle(String name) {  
 switch (name)  
 {  
 case "Lambo":  
 return new LuxuryVehicleLamoborgini();  
 case "Mustang":  
 return new LuxuryVehicleMustange();  
 default:  
 return null;  
  
 }  
  
 }  
  
  
}  
  
class OrdinaryFactory implements VehicleFactory  
{  
 *//create Vehicle of Ordinary on demand* @Override  
 public Vehicle getVehicle(String name)  
 {  
 switch (name)  
 {  
 case "Aulto":  
 return new OrdinaryVehicleAlto800();  
 case "Swift":  
 return new OrdinaryVehicleSwift();  
 default:  
 return null;  
  
 }  
 }  
  
  
}  
*// Driver Class*class MainVehicle  
{  
 public static void main(String[] args) {  
 *// use interface  
 // plug any Implemnted class ,according to requirement* VehicleFactory vehicleFactory = new OrdinaryFactory();  
 *// now pass the name* Vehicle vehicle = vehicleFactory.getVehicle("Swift");  
 vehicle.average();  
  
 VehicleFactory vehicleFactory2 = new LuxuryFactory();  
 *// now pass the name* Vehicle vehicle2 = vehicleFactory2.getVehicle("Lambo");  
 vehicle2.average();  
 }  
}