# Department of Computing

**SE-210: Software Design and Architecture**

**Class:** BESE-9AB

# Lab 08: Creational Design Patterns

# Instructor: Dr. Hasan Ali Khattak

# 

**Ahmed Hassan Ismail – BESE-9B – 237897**

# Lab 08: Abstract Factory Design Pattern

### Introduction:

Students will have hands-on experience of implementing a creational design pattern to a problem chosen by them.

### Lab Objectives:

This objective of this lab is to get a practical understanding and knowledge of Abstract Factory Pattern. After the completion of this lab, students will be able to apply abstract factory design pattern to any given scenario.

### Helping Material:

Please consult lecture slides on LMS.

### Lab Tasks

### Task

Chose any example and perform following tasks:

1. Draw UML Class diagram of the solution – after applying abstract factory design pattern.(3)
2. Fully functional code for the example. (4)
3. Write the understanding.(3)

**Answer:**

|  |
| --- |
| Solution |
| UML Class Diagram:    Source Code:  package abstractfactory;  public class Main {      public static void main(String[] args) {          System.out.println("Hala Hala");      }  }  // @author aismail.bese17seecs, Ahmed Hassan 237897 BESE-9B  abstract class Tyre {      Integer radius;      String material;      String rim;  }  class SedanTyre extends Tyre {      SedanTyre() {          super();          this.radius = 8;          this.material = "Sedan Tyre Material";          this.rim = "Sedan rims";      }  }  class SportsCarTyre extends Tyre {      SportsCarTyre() {          super();          this.radius = 6;          this.material = "Sports Car Tyre Material";          this.rim = "Sports Car rims";      }  }  abstract class Horn {      String noise;      public void honk() {          System.out.println(this.noise);      }  }  class SedanHorn extends Horn {      SedanHorn() {          this.noise = "\*\* Sedan Honking Noise \*\*";      }  }  class SportsCarHorn extends Horn {      SportsCarHorn() {          this.noise = "Sports Car Honking noise";      }  }  class Car implements Cloneable {      protected String bodyStyle;      protected String power;      protected String engine;      protected String breaks;      protected String seats;      protected String windows;      protected String fuelType;      protected String carType;      public Car(String carType) {          this.carType = carType;      }      public void setCarType(String type) {          this.carType = type;      }      public String getBodyStyle() {          return bodyStyle;      }      public void setBodyStyle(String bodyStyle) {          this.bodyStyle = bodyStyle;      }      public String getPower() {          return power;      }      public void setPower(String power) {          this.power = power;      }      public String getEngine() {          return engine;      }      public void setEngine(String engine) {          this.engine = engine;      }      public String getBreaks() {          return breaks;      }      public void setBreaks(String breaks) {          this.breaks = breaks;      }      public String getSeats() {          return seats;      }      public void setSeats(String seats) {          this.seats = seats;      }      public String getWindows() {          return windows;      }      public void setWindows(String windows) {          this.windows = windows;      }      public String getFuelType() {          return fuelType;      }      public void setFuelType(String fuelType) {          this.fuelType = fuelType;      }      @Override      public String toString() {          StringBuilder sb = new StringBuilder();          sb.append("--------------" + carType + "--------------------- \n");          sb.append(" Body: ");          sb.append(bodyStyle);          sb.append("\n Power: ");          sb.append(power);          sb.append("\n Engine: ");          sb.append(engine);          sb.append("\n Breaks: ");          sb.append(breaks);          sb.append("\n Seats: ");          sb.append(seats);          sb.append("\n Windows: ");          sb.append(windows);          sb.append("\n Fuel Type: ");          sb.append(fuelType);          return sb.toString();      }      public Car clone() throws CloneNotSupportedException {          return (Car) super.clone();      }  }  class Sedan extends Car {      Sedan() {          super("Sedan");          this.setBodyStyle("External dimensions: overall length (inches): 202.9, "                  + "overall width (inches): 76.2, overall height (inches): ←-60.7, wheelbase (inches): 112.9,"                  + " front track (inches): 65.3, rear track (inches): 65.5 and curb to curb turning circle (feet): 39.5");          this.setPower("285 hp @ 6,500 rpm; 253 ft lb of torque @ 4,000 rpm");          this.setEngine("3.5L Duramax V 6 DOHC");          this.setBreaks("Four-wheel disc brakes: two ventilated. Electronic brake distribution");          this.setSeats("Front seat center armrest.Rear seat center armrest. Split folding rear seats");          this.setWindows("Laminated side windows.Fixed rear window with defroster");          this.setFuelType("Gasoline 19 MPG city, 29 MPG highway, 23 MPG combined and 437 mi. range");      }  }  class SportsCar extends Car {      SportsCar() {          super("Sports Car");          this.setCarType("Sports");          this.setBodyStyle("External dimensions: overall length (inches): 192.3,"                  + " overall width (inches): 75.5, overall height (inches): 54.2, wheelbase (inches): 112.3,"                  + " front track (inches): 63.7, rear track (inches): 64.1 and curb to curb turning circle (feet): 37.7");          this.setPower("323 hp @ 6,800 rpm; 278 ft lb of torque @ 4,800 rpm");          this.setEngine("3.6L V 6 DOHC and variable valve timing");          this.setBreaks(                  "Four-wheel disc brakes: two ventilated. Electronic brake distribution. StabiliTrak stability control");          this.setSeats(                  "Driver sports front seat with one power adjustments manual height, front passenger seat sports front seat with one power adjustments");          this.setWindows("Front windows with one-touch on two windows");          this.setFuelType("Gasoline 17 MPG city, 28 MPG highway, 20 MPG combined and 380 mi. range");      }  }  interface CarFactory {      Car createCar();      Tyre createTyre();      Horn createHorn();  }  class SedanFactory implements CarFactory {      public Car createCar() {          return new Sedan();      }      public Tyre createTyre() {          return new SedanTyre();      }      public Horn createHorn() {          return new SedanHorn();      }  }  class SportsCarFactory implements CarFactory {      public Car createCar() {          return new SportsCar();      }      public Tyre createTyre() {          return new SportsCarTyre();      }      public Horn createHorn() {          return new SportsCarHorn();      }  }  class AbstractFactory {      /\*\*       \* @param args the command line arguments       \*/      public static void main(String[] args) {          CarFactory sportsCarFactory = new SportsCarFactory();          CarFactory sedanFactory = new SedanFactory();          Car sedan = sedanFactory.createCar();          Tyre sedanTyre = sedanFactory.createTyre();          Horn sedanHorn = sedanFactory.createHorn();          Car sportsCar = sportsCarFactory.createCar();          Tyre sportsCarTyre = sportsCarFactory.createTyre();          Horn sportsCarHorn = sportsCarFactory.createHorn();          System.out.println(sedan);          sedanHorn.honk();          System.out.println(sportsCar);          sportsCarHorn.honk();      }  } |

### Deliverables

Compile a single word document by filling in the solution part and submit this Word file on LMS. This lab grading policy is as follows: The lab is graded between 0 to 10 marks. The submitted solution can get a maximum of 5 marks. At the end of each lab or in the next lab, there will be a viva related to the tasks. The viva has a weightage of 5 marks. Insert the solution/answer in this document. You must show the implementation of the tasks in the designing tool, along with your completed Word document to get your work graded. You must also submit this Word document on the LMS. In case of any problems with submissions on LMS, submit your Lab assignments by emailing it to **Sundas Dawood** <sundas.dawood@seecs.edu.pk>