LAB_ANP_C6339_CLASSES

Student ID: AF0339439

YELLA UDAY KUMAR

Assignment 1:

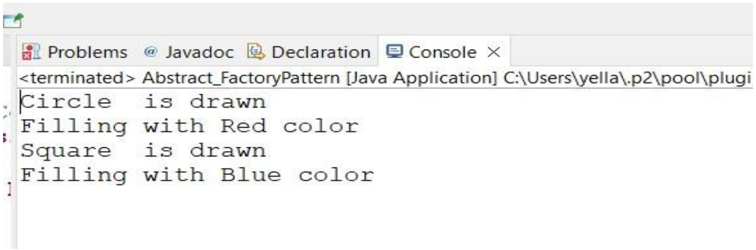
Write a program to Illustrate Abstract Factory method pattern.

Program:

```
package Designpatterns;

                  // Abstract Product A
                  interface Shape{
                      void draw();
                  }

                  // Concrete Product A1
                  class Circle implements Shape {
                      @Override
                      public void draw() {
                          System.out.println("Drawing Circle");
                      }
                  }

                  // Concrete Product A2
                  class Square implements Shape {
                      @Override
                      public void draw() {
                          System.out.println("Drawing Square");
                      }
                  }

                  // Abstract Product B
                  interface Color {
                      void fill();
                  }

                  // Concrete Product B1
                  class Red implements Color {
                      @Override
                      public void fill() {
                          System.out.println("Filling with Red color");
                      }
                  }

                  // Concrete Product B2
                  class Blue implements Color {
                      @Override
                      public void fill() {
                          System.out.println("Filling with Blue color");
                      }
                  }

                  // Abstract Factory
```

```java
interface AbstractFactory {
    Shape createShape();
    Color createColor();
}

// Concrete Factory 1
class RedShapeFactory implements AbstractFactory {
    @Override
    public Shape createShape() {
        return new Circle();
    }

    @Override
    public Color createColor() {
        return new Red();
    }
}

// Concrete Factory 2
class BlueShapeFactory implements AbstractFactory {
    @Override
    public Shape createShape() {
        return new Square();
    }

    @Override
    public Color createColor() {
        return new Blue();
    }
}

public class Abstract_FactoryPattern {
    public static void main(String[] args) {
        // Create a Red-themed shape factory
        AbstractFactory redFactory = new RedShapeFactory();
        Shape redCircle = redFactory.createShape();
        Color redColor = redFactory.createColor();

        redCircle.draw();
        redColor.fill();

        // Create a Blue-themed shape factory
        AbstractFactory blueFactory = new BlueShapeFactory();
        Shape blueSquare = blueFactory.createShape();
        Color blueColor = blueFactory.createColor();

        blueSquare.draw();
        blueColor.fill();
    }
}
```

Output:



```
<terminated> Abstract_FactoryPattern [Java Application] C:\Users\yella\.p2\pool\plugi
Circle  is drawn
Filling with Red color
Square  is drawn
Filling with Blue color
```

Assignment 2 :

Home Theater System: Create a home theater system using the Facade Pattern.Define classes for various components like DVD player, amplifier, projector, and screen. Implement a HomeTheaterFacade class that provides a simplified interface to controlthe entire home theater system, including turning it on, playing a movie, and turning it off.

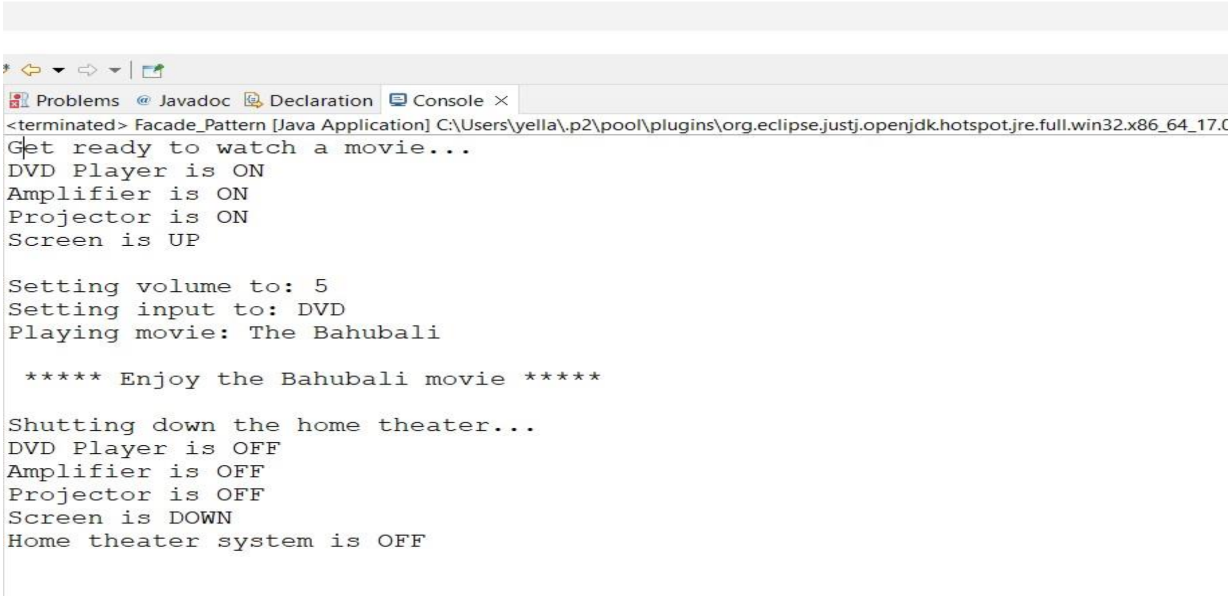
Program:

```
package Designpatterns;


        // DVD Player class
        class DVDPlayer {
            public void on() {
                System.out.println("DVD Player is ON");
            }

            public void play(String movie) {
                System.out.println("Playing movie: " + movie);
            }

            public void off() {
                System.out.println("DVD Player is OFF");
            }
        }

        // Amplifier class
        class Amplifier {
            public void on() {
                System.out.println("Amplifier is ON");
            }

            public void setVolume(int volume) {
                System.out.println("Setting volume to: " + volume);
            }

            public void off() {
                System.out.println("Amplifier is OFF");
            }
        }

        // Projector class
        class Projector {
            public void on() {
                System.out.println("Projector is ON");
            }

            public void setInput(String input) {
                System.out.println("Setting input to: " + input);
            }

            public void off() {
                System.out.println("Projector is OFF");
            }
```

```
      }

      // Screen class
      class Screen {
          public void up() {
              System.out.println("Screen is UP");
          }

          public void down() {
              System.out.println("Screen is DOWN");
          }
      }

      // Home Theater Facade class
      class HomeTheaterFacade {
          private DVDPlayer dvdPlayer;
          private Amplifier amplifier;
          private Projector projector;
          private Screen screen;

          public HomeTheaterFacade(DVDPlayer dvdPlayer, Amplifier amplifier,
Projector projector, Screen screen) {
              this.dvdPlayer = dvdPlayer;
              this.amplifier = amplifier;
              this.projector = projector;
              this.screen = screen;
          }

          public void watchMovie(String movie) {
              System.out.println("Get ready to watch a movie...");

              // Turn on the necessary components
              dvdPlayer.on();
              amplifier.on();
              projector.on();
              screen.up();

             System.out.println("");

              // Set up the components
              amplifier.setVolume(5);
              projector.setInput("DVD");

              // Play the movie
              dvdPlayer.play(movie);
              System.out.println("");

              System.out.println(" ***** Enjoy the Bahubali movie *****");
          }

          public void endMovie()
          {
             System.out.println("");
              System.out.println("Shutting down the home theater...");

              // Turn off the components
              dvdPlayer.off();
              amplifier.off();
              projector.off();
              screen.down();
```

```
            System.out.println("Home theater system is OFF");
        }
    }

    // Client code
    public class Facade_Pattern {
        public static void main(String[] args) {
            // Create components
            DVDPlayer dvdPlayer = new DVDPlayer();
            Amplifier amplifier = new Amplifier();
            Projector projector = new Projector();
            Screen screen = new Screen();

            // Create the Home Theater Facade
            HomeTheaterFacade homeTheater = new HomeTheaterFacade(dvdPlayer,
amplifier, projector, screen);

            // Watch a movie using the facade
            homeTheater.watchMovie("The Bahubali");

            // End the movie and shut down the home theater
            homeTheater.endMovie();
        }
    }
```

Output:



<terminated> Facade_Pattern [Java Application] C:\Users\yella\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0

```
Get ready to watch a movie...
DVD Player is ON
Amplifier is ON
Projector is ON
Screen is UP

Setting volume to: 5
Setting input to: DVD
Playing movie: The Bahubali

 ***** Enjoy the Bahubali movie *****

Shutting down the home theater...
DVD Player is OFF
Amplifier is OFF
Projector is OFF
Screen is DOWN
Home theater system is OFF
```