|  |  |  |  |
| --- | --- | --- | --- |
|  | **Course Name: Design Patterns/Thinking LAB** | **EXPERIMENT NO. 4** | |
| **Course Code: 20CP210P**  **Faculty: Dr. Ketan Sabale** | **Branch: CSE** | **Semester: IV** |
| **Submitted by: Jangle Parth**  **Roll no: 22BCP083** | | | |

Objective: To familiarize students with standard Structural design patterns.

Experiment: Explain the prototype design pattern and write a program using any object-oriented programming language to demonstrate the working of facade design pattern.

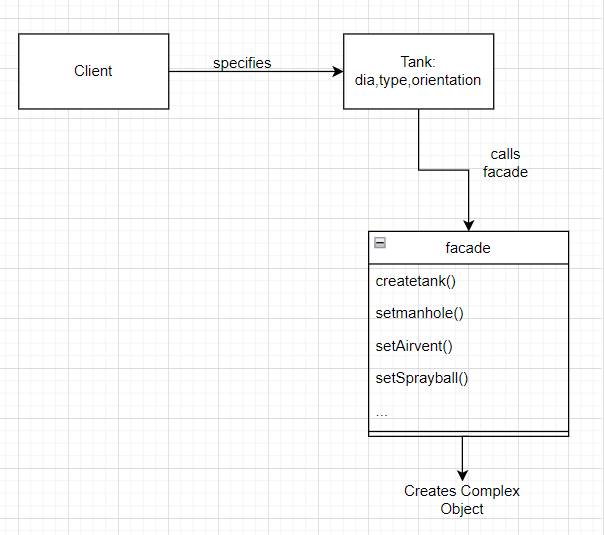
Theory: Imagine a Scenario where you are a customer and you want to make a complex product let say tank. How comfortable it would be if you just give dia and orientation of tank and you based on that a complex product i.e a tank is created.

In façade design pattern we can do it. In façade there is a intermediate between user and complex object called façade the task of façade is to create complex object based on dia and orientation of user.

**Problem Statement Explanation:**

We have a class tank which has 4 features type, position, dia where type, position, and Dia are the attributes required to create the complex object. We have a class façade which is responsible for creating tank. It has methods to create small object of tank like setmanhole, setAirvent… , and a method create tank to assemble all this product together. It has show info and getprice method to check the complex object created and price of the Tank

**Flowchart Explanation:**

****

**Code:**

class Tank {

    public String type;

    public String position;

    public int dia;

    public String client;

    public Tank(String position, String type, int dia, String client) {

        this.type = type;

        this.dia = dia;

        this.client = client;

        this.position = position;

    }

}

class facade {

    public String[] tankparts = new String[5];

    public int[] tankpartscost = new int[5];

    public void createtank(Tank t) {

        setManhole(t);

        setAirvent(t);

        setLiftinghook(t);

        setsprayball(t);

        setShell(t);

        getprice(t);

        getinfo(t);

    }

    public void setManhole(Tank t) {

        int manholedia;

        if (t.position == "Horizontal") {

            manholedia = (t.dia / 100);

            tankparts[0] = "Manhole of position: Top";

            tankpartscost[0] = manholedia \* 314;

        } else if (t.position == "Vertical") {

            manholedia = (int) ((t.dia \* 2 \* 3.14) / 100);

            tankparts[0] = "Manhole of position: Side";

            tankpartscost[0] = manholedia \* 314;

        } else {

            if (t.dia > 1000) {

                manholedia = 300;

                tankparts[0] = "Manhole of position: Side";

            } else {

                manholedia = (t.dia / 100);

                tankparts[0] = "Manhole of position: Side";

                tankpartscost[0] = manholedia \* 314;

            }

        }

    }

    public void setLiftinghook(Tank t) {

        int nooflifitnghook = (t.dia / 1000) \* 2;

        tankpartscost[1] = nooflifitnghook \* 300;

        if (nooflifitnghook > 4) {

            tankparts[1] = "Lifting Hook Position: Vertical";

        } else {

            tankparts[1] = "Lifting Hook Position: Horizontal";

        }

    }

    public void setsprayball(Tank t) {

        if (t.type == "Pectin Mixing") {

            tankparts[2] = "Sprayball of Speed: 500 rpm";

            tankpartscost[2] = 1000 \* (500 / 100);

        } else if (t.type == "Acid Storage") {

            tankparts[2] = "Sprayball of Speed: 100 rpm";

            tankpartscost[2] = 1000 \* (100 / 100);

        } else if (t.type == "Milk Storage") {

            tankparts[2] = "Sprayball of Speed: 200 rpm";

            tankpartscost[2] = 1000 \* (200 / 100);

        } else if (t.type == "Butter Storage") {

            tankparts[2] = "Sprayball of Speed: 300 rpm";

            tankpartscost[2] = 1000 \* (300 / 100);

        } else {

            tankparts[2] = "Sprayball of Speed: 500rpm";

            tankpartscost[2] = 1000 \* (500 / 100);

        }

    }

    public void setAirvent(Tank t) {

        if (t.type.equals("Acid Storage") || t.type.equals("Milk Storage") || t.type.equals("Butter Storage")) {

            tankparts[3] = "No Airvent Required";

            tankpartscost[3] = 0;

        } else if (t.position == "Horizontal" && t.type == "Pectin Mixing") {

            tankparts[3] = "Airvent of Type: Top";

            tankpartscost[3] = (t.dia / 8) \* 314;

        } else if (t.position == "Vertical" && t.type == "Pectin Mixing") {

            tankparts[3] = "Airvent of Type: Side";

            tankpartscost[3] = (t.dia / 8) \* 314;

        } else {

            tankparts[3] = "Airvent of Type: Top";

            tankpartscost[3] = (t.dia / 8) \* 314;

        }

    }

    public void setShell(Tank t) {

        if (t.type.equals("Acid Storage") || t.type.equals("Pectin Minxing")) {

            tankparts[4] = "Tank Shell Material: SS 316";

            tankpartscost[4] = t.dia \* 340;

        } else if (t.type.equals("Milk Storage") || t.type.equals("Butter Storage")) {

            tankparts[4] = "Tank Shell Material: SS 304";

            tankpartscost[4] = t.dia \* 314;

        }

    }

    public void getprice(Tank t) {

        int price = 0;

        for (int i = 0; i <= tankpartscost.length - 1; i++) {

            price = price + tankpartscost[i];

        }

        System.out.println("Price of Tank is: " + price);

        System.out.println();

    }

    public void getinfo(Tank t) {

        for (int j = 0; j <= tankparts.length - 1; j++) {

            System.out.println(tankparts[j]);

        }

        System.out.println();

    }

}

public class facadepattern {

    public static void main(String args[]) {

        Tank t = new Tank("Horizontal", "Acid Storage", 5000, "Coca-Cola");

        Tank t2 = new Tank("Vertical", "Milk Sotrage", 2500, "Nestle");

        facade atpl = new facade();

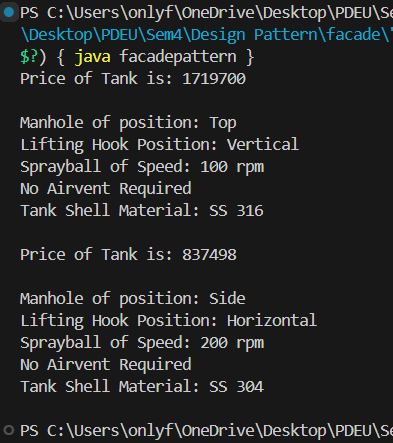
        atpl.createtank(t);

        atpl.createtank(t2);

    }

}

**Output:**

****