|  |  |  |  |
| --- | --- | --- | --- |
|  | **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 Creational design patterns.

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

Theory : Imagine a Scenario where you own a company which gets bulk orders from various company and the product is almost same due to which the company gets repeat orders now in this scenario to register repeat orders the company either has to add everything manually which is directly eliminated or else the company has to fetch the data from the data base and add them to the object of new order. This is costly when the order size is big. Hence we use Prototype design pattern .

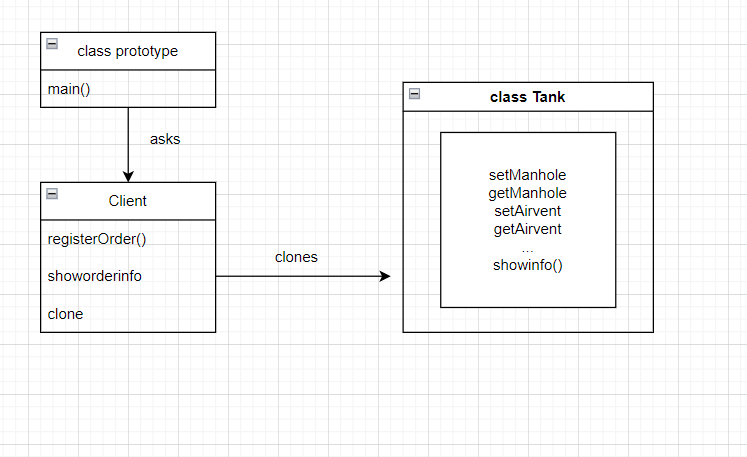
In prototype design pattern we take help of cloning which is a interface available in Java it take the parameters of older object that was created earlier and pass it to the new object. There are two types of cloning 1) Shallow clone which means change in old object will reflect in new and 2) Deep changes won’t reflect

**Problem Statement Explanation:**

We have a Class Tank which has attributes Manhole , Airvent and Sprayball it has method like setAirvent() , getAirvent()… and a method showinfo() to show all info of the particular Tank object then we have a class client that implements a Cloneable interface it has 2 attributes clientname,totaltank and a List of Tanks

It has 2 methods register order which allows user to enter the info of the order and showorderinfo which shows all the info of the order then it has 2 cloning methods shallow clone and deep clone.

**Flowchart Explanation:**

****

**Code:**

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

class Tank {

    private String Manhole;

    private String Airvent;

    private String Sprayball;

    public void setAirvent(String airvent) {

        Airvent = airvent;

    }

    public void setManhole(String manhole) {

        Manhole = manhole;

    }

    public void setSprayball(String sprayball) {

        Sprayball = sprayball;

    }

    public void showinfo() {

        System.out.println();

        System.out.println("Tank Has a Manhole of Type: " + Manhole);

        System.out.println("Tank Has a Airvent of Type: " + Airvent);

        System.out.println("Tank Has a Sprayball of Type: " + Sprayball);

        System.out.println();

    }

    public String getAirvent() {

        return Airvent;

    }

    public String getManhole() {

        return Manhole;

    }

    public String getSprayball() {

        return Sprayball;

    }

}

class Client implements Cloneable {

    private String clientname;

    private int totaltank;

    List<Tank> tanks = new ArrayList<>();

    Client(String name, int total\_tanks) {

        this.clientname = name;

        this.totaltank = total\_tanks;

    }

    public void RegisterOrder() {

        Scanner sc = new Scanner(System.in);

        System.out.println();

        for (int i = 0; i < totaltank; i++) {

            tanks.add(new Tank());

            System.out.println();

            System.out.println("Enter the Airvent Type for Tank " + (i + 1));

            String pairvent = sc.nextLine();

            System.out.println("Enter the Manhole Type for Tank " + (i + 1));

            String pmanhole = sc.nextLine();

            System.out.println("Enter the SprayBall Type for Tank " + (i + 1));

            String psprayball = sc.nextLine();

            tanks.get(i).setAirvent(pairvent);

            tanks.get(i).setManhole(pmanhole);

            tanks.get(i).setSprayball(psprayball);

        }

    }

    public void showorderinfo() {

        System.out.println();

        System.out.println("Client name: " + clientname);

        System.out.println("Total Tanks: " + totaltank);

        int i = 0;

        for (Tank tank : tanks) {

            tanks.get(i).showinfo();

            i++;

        }

    }

    public Object Shallowclone() throws CloneNotSupportedException {

        return super.clone();

    }

    public Client DeepClone() throws CloneNotSupportedException {

        Client c = new Client(clientname, totaltank);

        int i = 0;

        for (Tank tank : tanks) {

            Tank originalTank = tanks.get(i);

            Tank clonedtank = new Tank();

            clonedtank.setAirvent(originalTank.getAirvent());

            clonedtank.setManhole(originalTank.getManhole());

            clonedtank.setSprayball(originalTank.getSprayball());

            c.tanks.add(clonedtank);

            i++;

        }

        return c;

    }

}

public class prototype {

    public static void main(String[] args) throws CloneNotSupportedException {

        Client Tetrapack = new Client("Tetrapack", 4);

        Tetrapack.RegisterOrder();

        Tetrapack.showorderinfo();

        Client SPXFlow = Tetrapack.DeepClone();

        Client Amul = (Client) Tetrapack.Shallowclone();

        Tetrapack.tanks.remove(2);

        SPXFlow.showorderinfo();

        Amul.showorderinfo();

    }

}

**Output:**

