| | | | |
|---|---|---|---|
| | **Course Name: Design Patterns/Thinking LAB** | **EXPERIMENT NO. 10** | |
| | **Course Code: 20CP210P**<br>**Faculty: Dr. Ketan Sabale** | **Branch: CSE** | **Semester: IV** |
| **(To be filled by Student)**<br>**Submitted by: Jangle Parth**<br>**Roll no: 22BCP083** | | | |

Objective: To familiarize students with standard Structural design patterns.

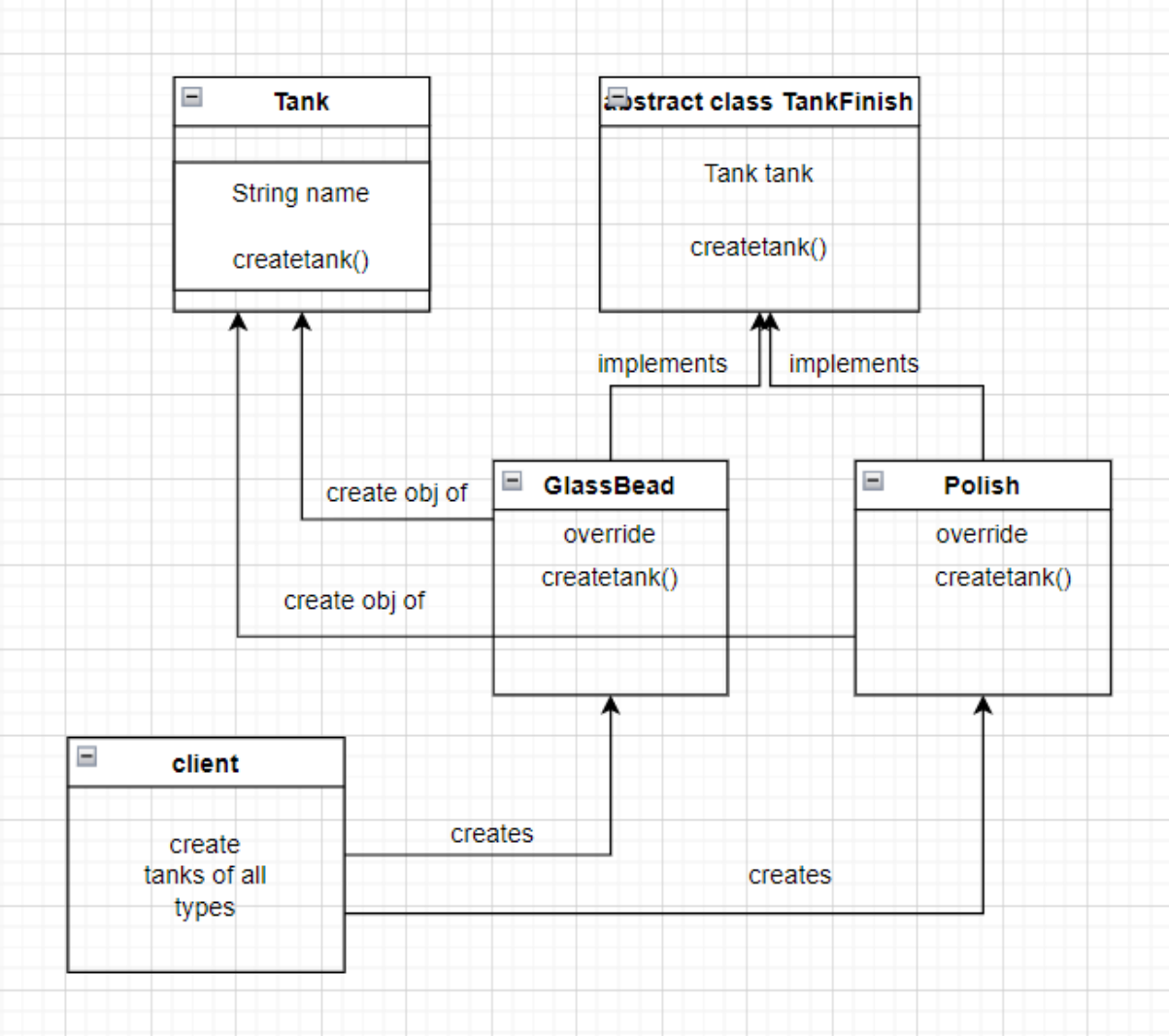
Experiment: Explain the Decorator design pattern and write a program using any object-oriented programming language to demonstrate the working of Decorator design pattern.

Theory: Imagine a Scenario where you own a tank manufacturing factory and you know also start polishing of tank in your factory as you are smart owner you know that creating a extra class of polished product would not be most efficient way so you create a abstract class Tank Finish which is responsible for creating diff kind of finished tank. This addition of a class which takes normal object and make it decorated is called is called Decorator design pattern

# Problem Statement Explanation:

We have a Tank class which has a method create tank which is responsible for creating tanks. Then we have a decorator class like Glass Bead ,Polish etc. which add some decoration to the base product.

# Flowchart Explanation:




# Code:

```
class Tank {
    String name;

    Tank(String name) {
        this.name = name;
    }

    public void createtank() {
        System.out.println();
        System.out.println(name);
        System.out.println("Created Tank");
    }
```

```
}

abstract class TankFinish {
    Tank tank;

    TankFinish(Tank tank) {
        this.tank = tank;
    }

    public void createtank() {
        tank.createtank();
    }
}

class GlassBead extends TankFinish {

    GlassBead(Tank tank) {
        super(tank);
    }

    public void createtank() {
        tank.createtank();
        System.out.println("Added Glass Bead Finish");
    }

}

class Polish extends TankFinish {

    Polish(Tank tank) {
        super(tank);
    }

    public void createtank() {
        tank.createtank();
        System.out.println("Added Polished Finish");
    }

}

public class decorator {
    public static void main(String[] args) {
        Tank t1 = new Tank("Milk Sotrage Tank");
        GlassBead t2 = new GlassBead(new Tank("Acid Sotrage Tank"));
        Polish t3 = new Polish(new Tank("Beer Sotrage Tank"));
        t1.createtank();
        t2.createtank();
        t3.createtank();
```

```
    }
}
```

# Output:

```
PS C:\Users\onlyf\OneDrive\Desktop\PDEU\Sem4\Design Pattern> cd "c:\Users\onlyf\OneDriv
op\PDEU\Sem4\Design Pattern\Decorator\" ; if ($?) { javac decorator.java } ; if ($?) {
corator }

Milk Sotrage Tank
Created Tank

Acid Sotrage Tank
Created Tank
Added Glass Bead Finish

Beer Sotrage Tank
Created Tank
Added Polished Finish
PS C:\Users\onlyf\OneDrive\Desktop\PDEU\Sem4\Design Pattern\Decorator>
```