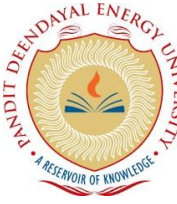


| | | | | |
|---|--|--|------------------------------|---------------------|
|  | Course Name: Design Patterns/Thinking LAB | | EXPERIMENT NO. 15 | |
| | Course Code: 20CP210P Faculty: Dr. Ketan Sabale | | Branch: CSE | Semester: IV |
| (To be filled by Student) Submitted by: Jangle Parth Roll no: 22BCP083 | | | | |

Objective: To familiarize students with standard Architectural design patterns.

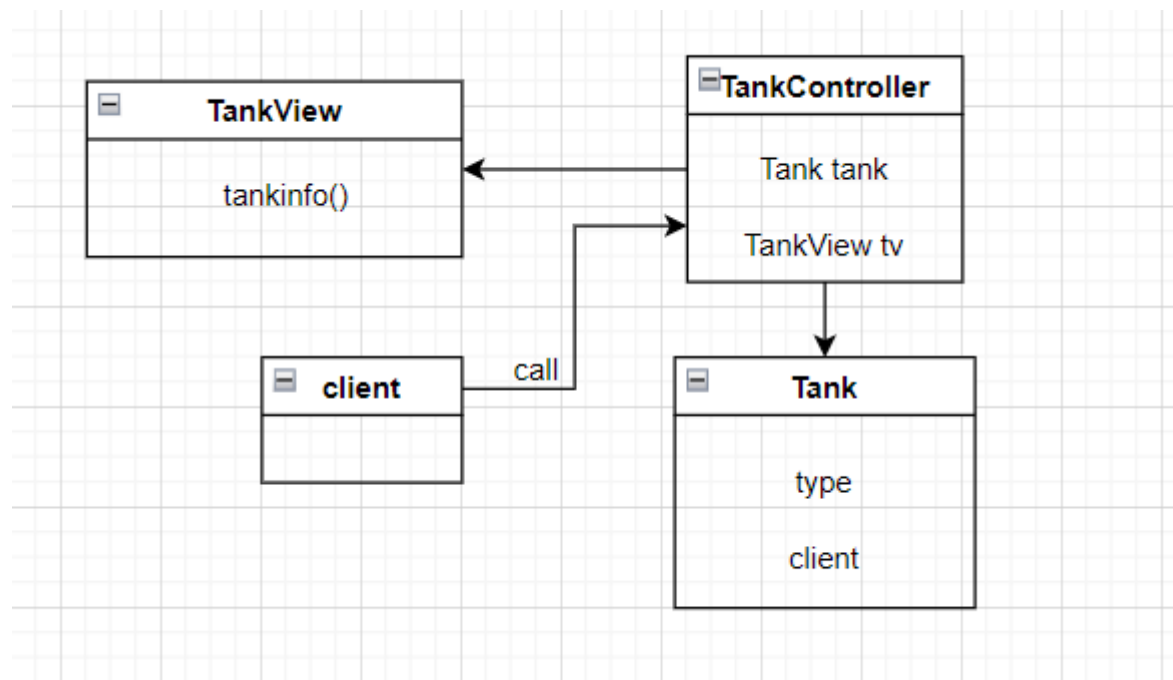
Experiment: Explain the Model-View-Controller (MVC) design pattern and write a program using any object-oriented programming language to demonstrate the working of MVC design pattern.

Theory: It is a software architecture pattern that separates an application into three main components: Model, View, and Controller, making it easier to manage and maintain the codebase. It also allows for the reusability of components and promotes a more modular approach to software development.

Problem Statement Explanation:

Tank Model Represent the data of like type of tank and client name and provide method to access and modify data. Tank View Represents how the data should be displayed to the user. Contains a method **tankinfo** to print the tank details. Tank Controller Acts as an intermediary between the Model and the View. Contains references to the Model and View objects. Provides methods to update the View (**update View**).

Flowchart Explanation:



Code:

```
package MVC;

class Tank {
    String Type;
    String Client;

    Tank(String type, String client) {
        this.Type = type;
        this.Client = client;
    }
}

class TankView {
    public void tankinfo(String type, String client) {
        System.out.println("Tank Name: " + type);
        System.out.println("Client Name: " + client);
    }
}
```

```

class TankController {
    Tank model;
    TankView view;

    public TankController(Tank model, TankView view) {
        this.model = model;
        this.view = view;
    }

    public void updateView() {
        view.tankinfo(model.Type, model.Client);
    }
}

public class MVCPattern {
    public static void main(String[] args) {
        Tank model = new Tank("Milk Storage", "Amul");
        TankView view = new TankView();
        TankController controller = new TankController(model, view);
        controller.updateView();
        controller.model.Type = "Acid Storage tank";
        controller.model.Client = "Coca-Cola";
        controller.updateView();
    }
}

```

Output:

```

Code\User\workspaceStorage\0dabdf8b0a2dea3cfa522958b7e603a2\redhat.ja
89891cc3\bin' 'MVC.MVCPattern'
Tank Name: Milk Storage
Client Name: Amul
Tank Name: Acid Storage tank
Client Name: Coca-Cola
PS C:\Users\onlyf\OneDrive\Desktop\PDEU\Sem4\Design Pattern>

```

