**OOAD LAB 9&10**

|  |  |  |
| --- | --- | --- |
| NAME : SAHANA RAO | SRN: PES1UG20CS588 | SECTION: J |

1. **Problem statement**

Given a company’s Leave system management which has 3 kinds of leaves: CL, SL and VL each having its own conditions. The leaves are approved by a hierarchy of management staff based on conditions. We are required to represent the design (using appropriate design patterns) in a UML Class Diagram and implement the same.

1. **Design patterns considered**

The design patterns considered in our approach are mainly **creational** and **behavioural**. Under creational we are looking forward to use Factory pattern and in behavioural we are using chain of responsibility.

*What is Creational pattern and factory pattern?*

Creational design patterns deals with object creation mechanisms, trying to create objects in a manner suitable to the situation.

Factory pattern is a creational design pattern that provides an interface for creating objects in a superclass, but allows subclasses to alter the type of objects that will be created. It is a way to encapsulate the creation of objects.

*What is behavioural pattern and chain of responsibility?*

Behavioural patterns help to identify common communication patterns among objects and define how those objects should interact with one another.

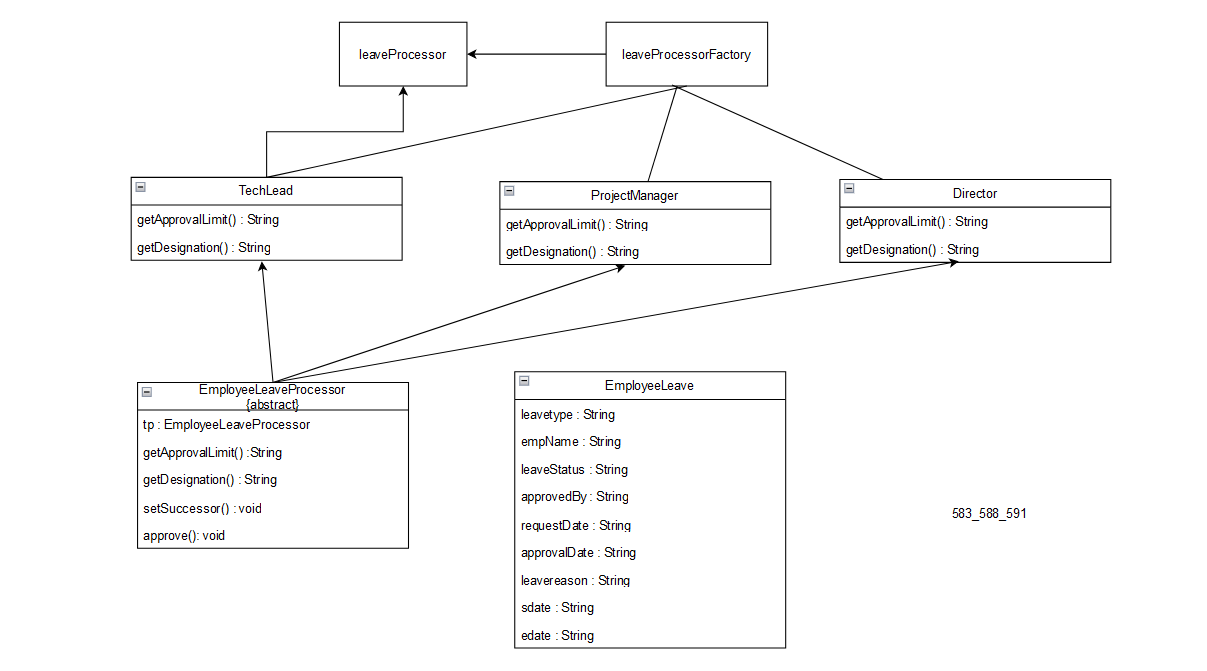
Chain of Responsibility is one of the behavioural design patterns. It allows you to pass requests along a chain of handlers until one of them can handle the request.

1. **Design patterns used**

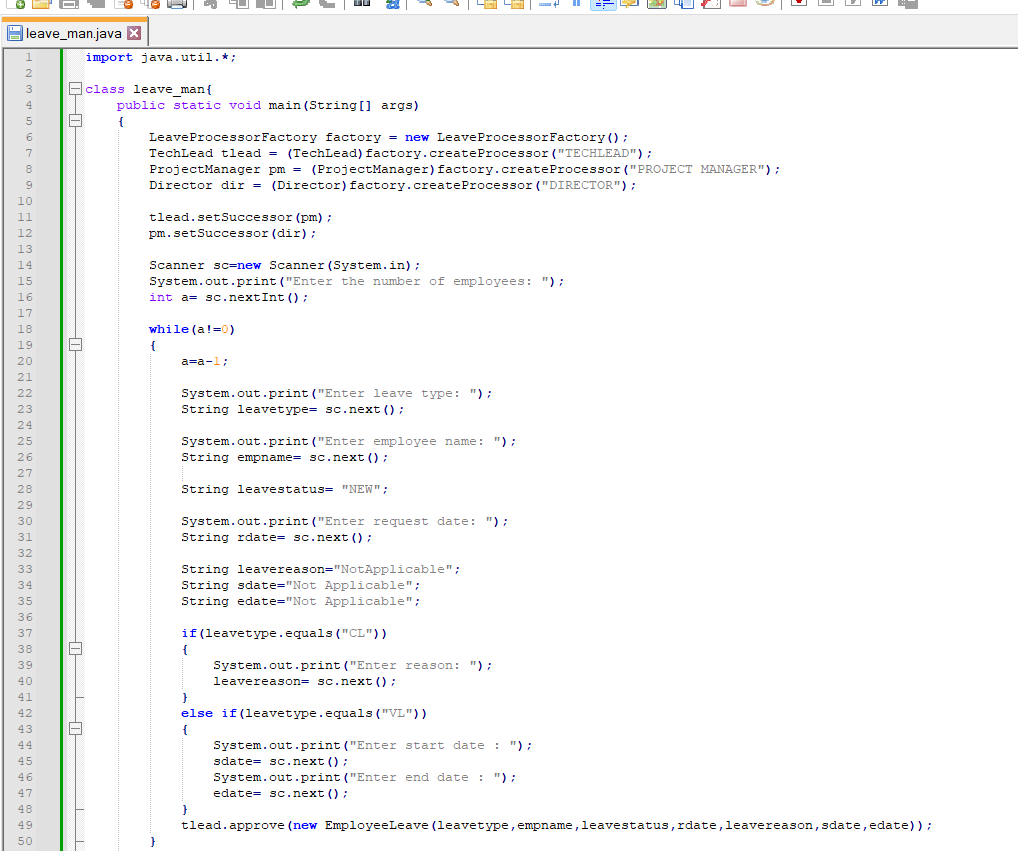
**Factory pattern:** The LeaveProcessorFactory class creates objects of the EmployeeLeaveProcessor type based on the input argument type. This factory pattern allows the client code to create objects without knowing the specific class of the object that will be created, thereby providing an abstraction layer between the client code and the object creation process.

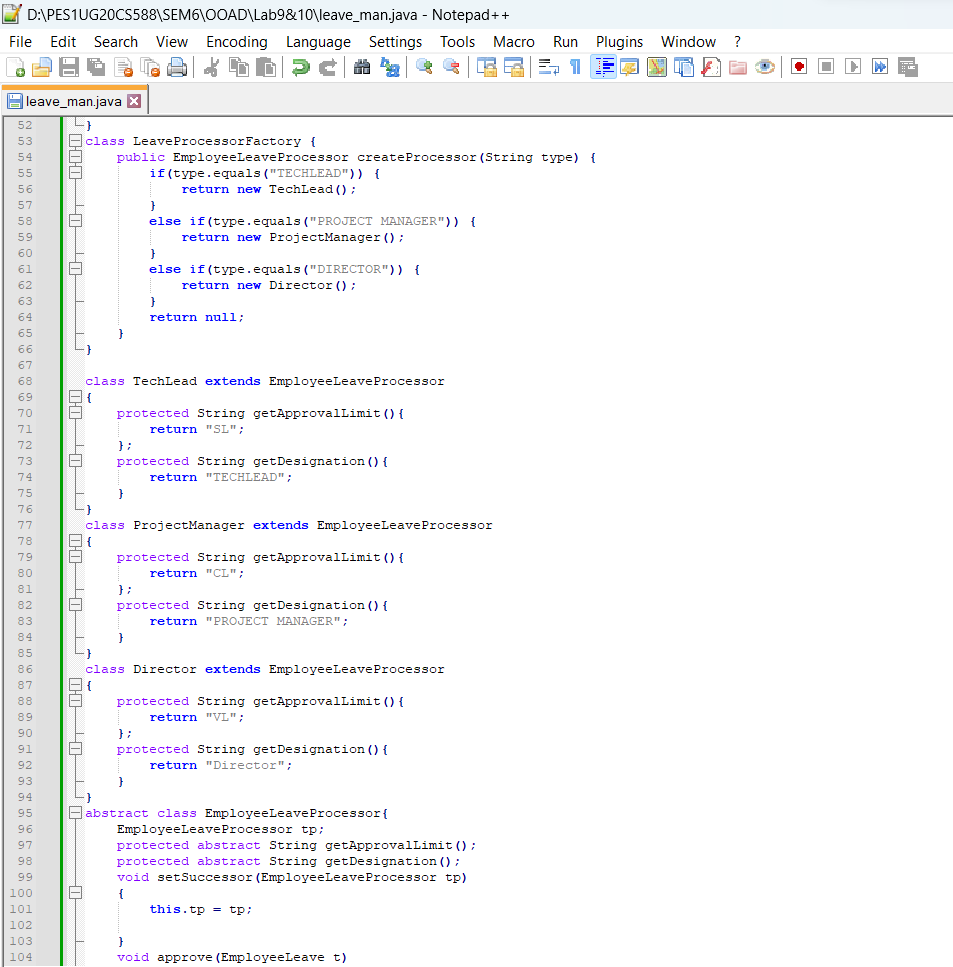
**Chain of Responsibility:** The design pattern used in this code is the Chain of Responsibility pattern. The EmployeeLeaveProcessor abstract class and its concrete subclasses TechLead, ProjectManager, and Director implement the chain of responsibility pattern, where each object in the chain has a reference to the next object in the chain. The Client (leave\_man class) sends a request to the first object in the chain (TechLead), and the object decides whether it can handle the request or not. If it can handle the request, it processes it; otherwise, it passes the request to the next object in the chain (ProjectManager), and the process continues until the request is handled or there is no more object in the chain.

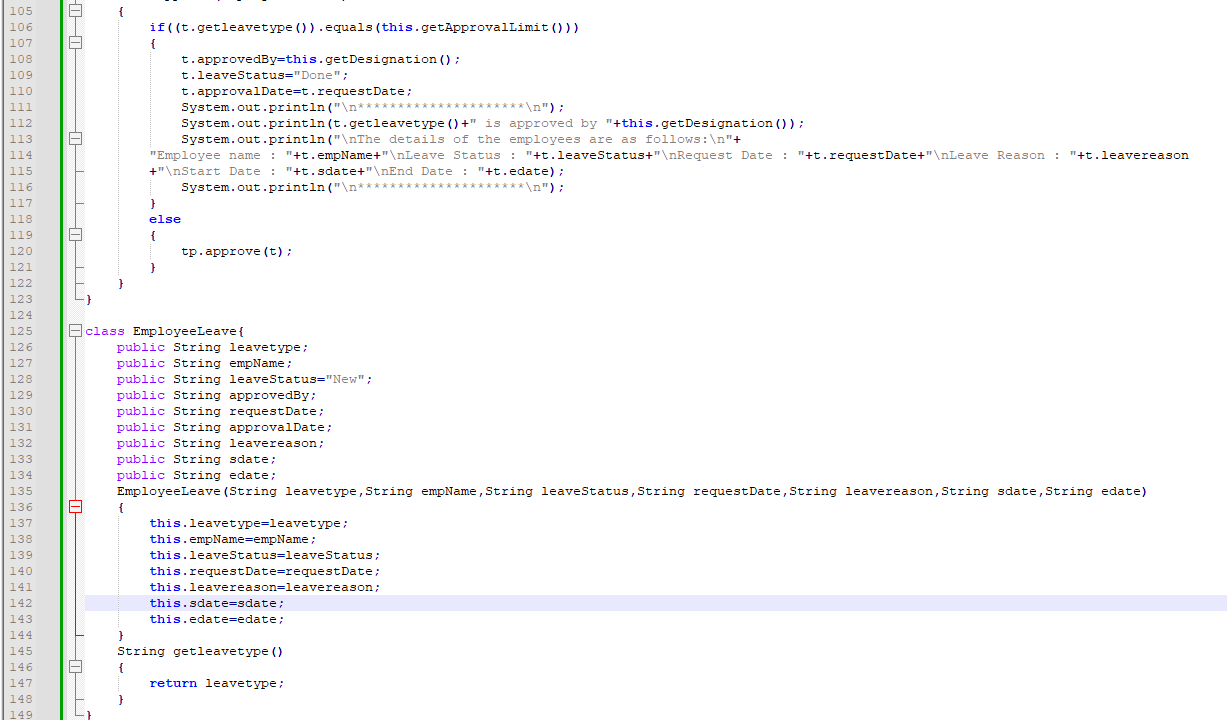
**4. UML Class Model**



1. **Code**







1. **Input and Output Screenshots for all types of leaves**

