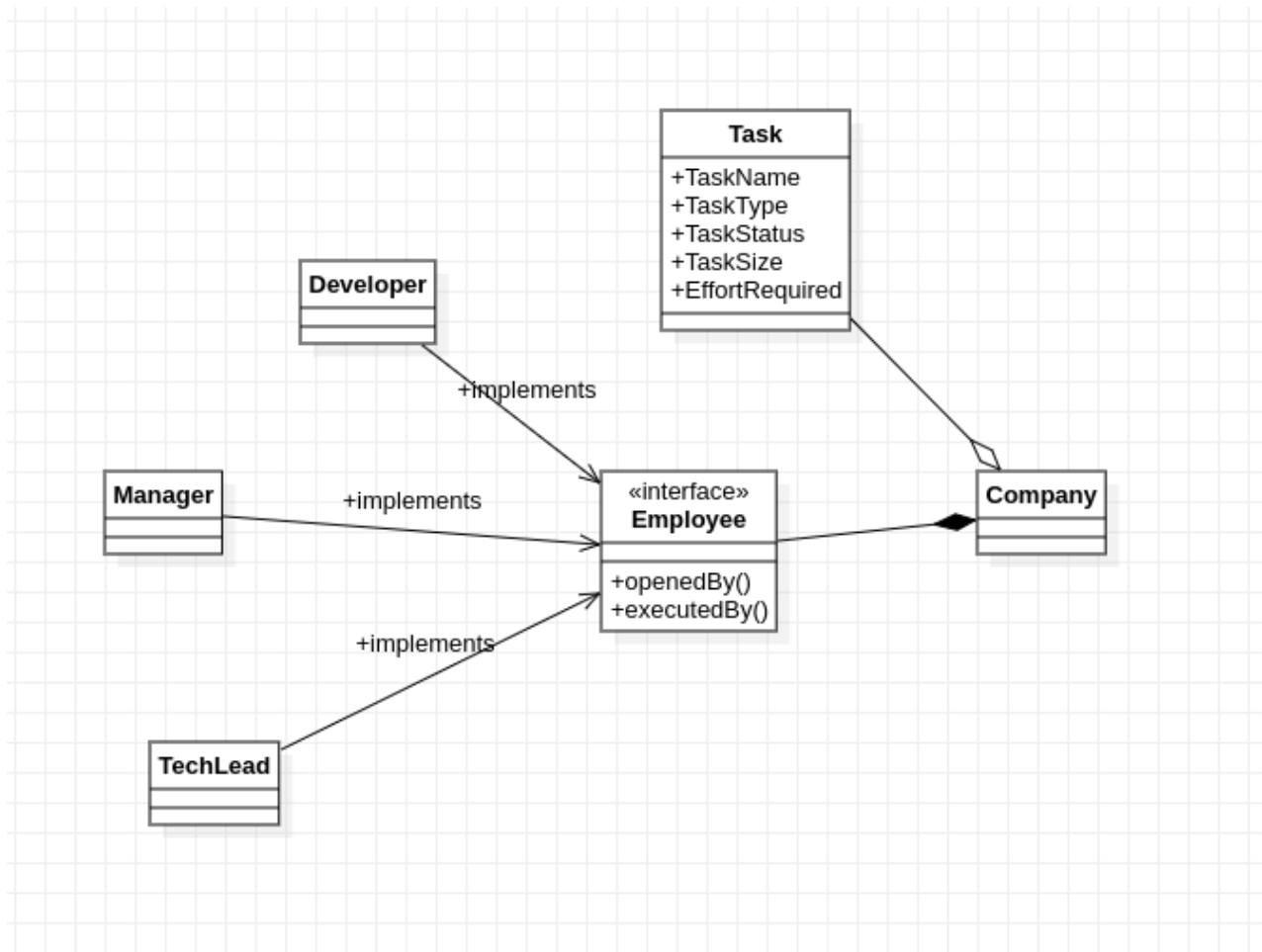


Assignment on Design Patterns

Name: Sumukh Raju Bhat	SRN: PES1UG19CS519	Date: 28/04/2022
	Section: H	OOAD Assignment

CLASS DIAGRAM:



Assignment on Design Patterns

Design pattern information:

Creation pattern:

1) Factory :

The employee interface serves as a common ground, allowing the firm to add more categories of employees in addition to developer, techlead, and manager without having to change the base class.

Behavioral pattern:

1) Chain of responsibility :

The request is passed from the employee to the task object.

2) State:

Depending on which employee is allocated to the task, the status of the task object varies.

Output Screen Shot:

```
sumukhbhat2701@pop-os:week8$ javac Company.java
sumukhbhat2701@pop-os:week8$ java Company
SRN: PES1UG19CS519
Enter the task level for the task:
1
TechLead's open()
Developer's execute()
sumukhbhat2701@pop-os:week8$ java Company
SRN: PES1UG19CS519
Enter the task level for the task:
2
Manager's open()
TechLead's execute()
sumukhbhat2701@pop-os:week8$ java Company
SRN: PES1UG19CS519
Enter the task level for the task:
3
Manager's open()
Manager's execute()
sumukhbhat2701@pop-os:week8$ java Company
SRN: PES1UG19CS519
Enter the task level for the task:
4
4 not recognized as a valid level. Valid levels are 1, 2 and 3.
sumukhbhat2701@pop-os:week8$
```

Assignment on Design Patterns

Code:

1) Company.java

```
// SRN: PES1UG19CS519
import java.util.Scanner;

public class Company{
    public static void main(String[] args)
    {

        Scanner sc = new Scanner(System.in);
        int level;
        System.out.println("SRN: PES1UG19CS519");
        System.out.println("Enter the task level for the task:");
        level = sc.nextInt();
        sc.close();
        Task task = new Task("Tester", "High", "Started", level, "Easy", "1
day");

        Employee techLead = new TechLead();
        Employee developer = new Developer();
        Employee manager = new Manager();

        if(task.taskLevel==1)
        {
            task.openedBy(techLead);
            task.executedBy(developer);
        }
        else if(task.taskLevel==2)
        {
            task.openedBy(manager);
            task.executedBy(techLead);
        }
        else if(task.taskLevel==3)
        {
            task.openedBy(manager);
            task.executedBy(manager);
        }
        else
        {

```

Assignment on Design Patterns

```
        System.out.println(level + " not recognized as a valid level.  
Valid levels are 1, 2 and 3.");  
    }  
}  
}
```

2) Task.java

```
// SRN: PES1UG19CS519  
public class Task {  
    String taskName;  
    String taskType;  
    String taskStatus;  
    int taskLevel;  
    String taskSize;  
    String effortRequired;  
  
    public Task()  
    {  
        taskName = "default-task";  
        taskType = "default-task-type";  
        taskStatus = "default-task-status";  
        taskLevel = 1;  
        taskSize = "default-task-size";  
        effortRequired = "0 day";  
    }  
  
    public Task(String taskName, String taskType, String taskStatus, int  
taskLevel, String taskSize, String effortRequired) {  
        this.taskName = taskName;  
        this.taskType = taskType;  
        this.taskStatus = taskStatus;  
        this.taskLevel = taskLevel;  
        this.taskSize = taskSize;  
        this.effortRequired = effortRequired;  
    }  
  
    public void executedBy(Employee employee) {  
        employee.executedBy();  
    }  
}
```

Assignment on Design Patterns

```
public void openedBy(Employee employee) {  
    employee.openedBy();  
}  
}
```

3) Employee.java

```
// SRN: PES1UG19CS519  
public interface Employee{  
    public void openedBy();  
    public void executedBy();  
}
```

4) Manager.java

```
// SRN: PES1UG19CS519  
public class Manager implements Employee{  
    public void openedBy() {  
        System.out.println("Manager's open()");  
    }  
    public void executedBy() {  
        System.out.println("Manager's execute()");  
    }  
}
```

5) Developer.java

```
// SRN: PES1UG19CS519  
public class Developer implements Employee {  
    public void openedBy() {  
        System.out.println("Developer's open()");  
    }  
    public void executedBy() {  
        System.out.println("Developer's execute()");  
    }  
}
```

6) TechLead.java

Assignment on Design Patterns

```
// SRN: PES1UG19CS519
public class TechLead implements Employee {
    public void openedBy() {
        System.out.println("TechLead's open()");
    }
    public void executedBy() {
        System.out.println("TechLead's execute()");
    }
}
```