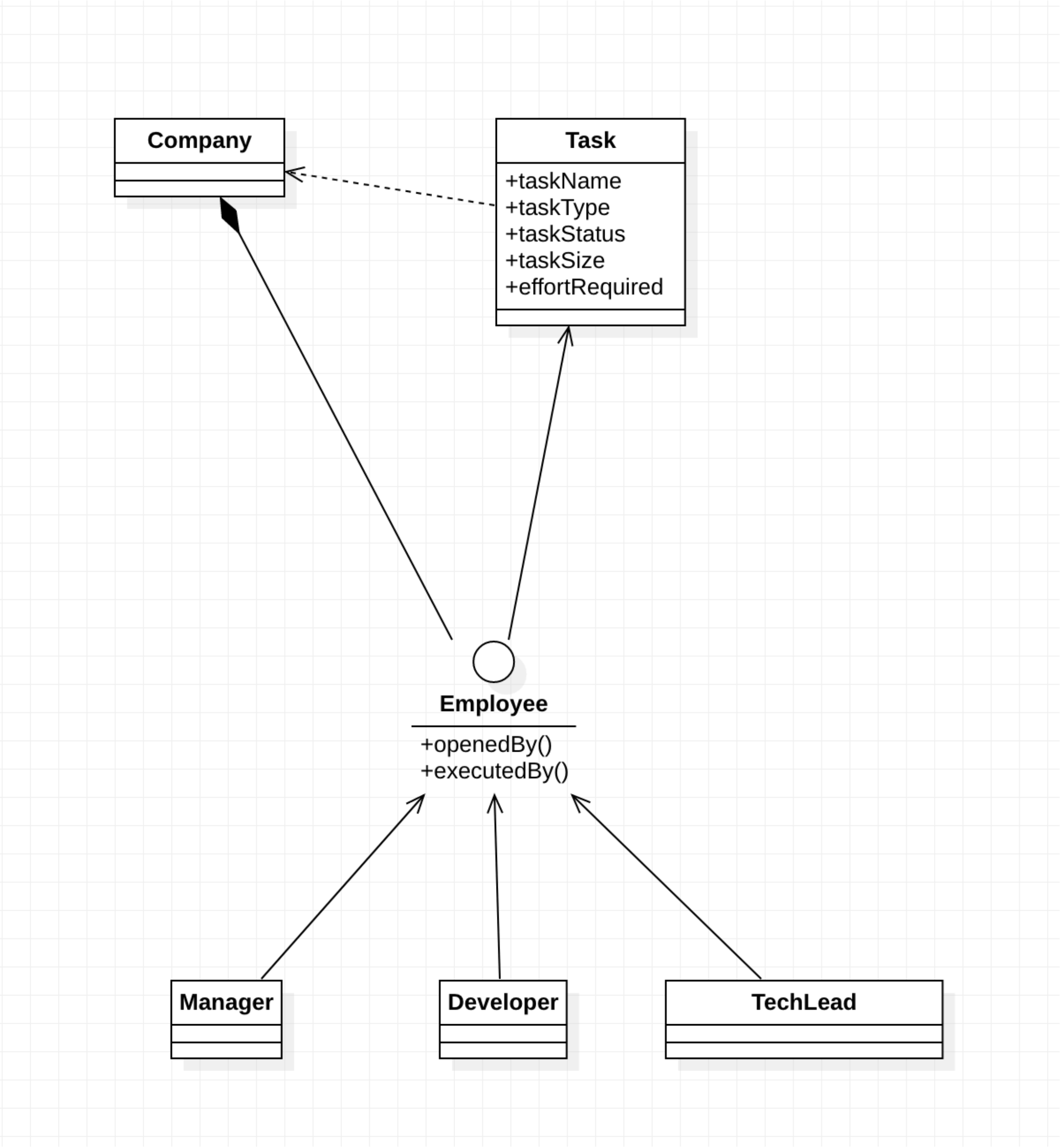
Name: Sammith S Bharadwaj	SRN: PES1UG19CS427	Date: 27/04/2022
	Section: G	OOADJ Assignment

CLASS DIAGRAM:



Design pattern information:

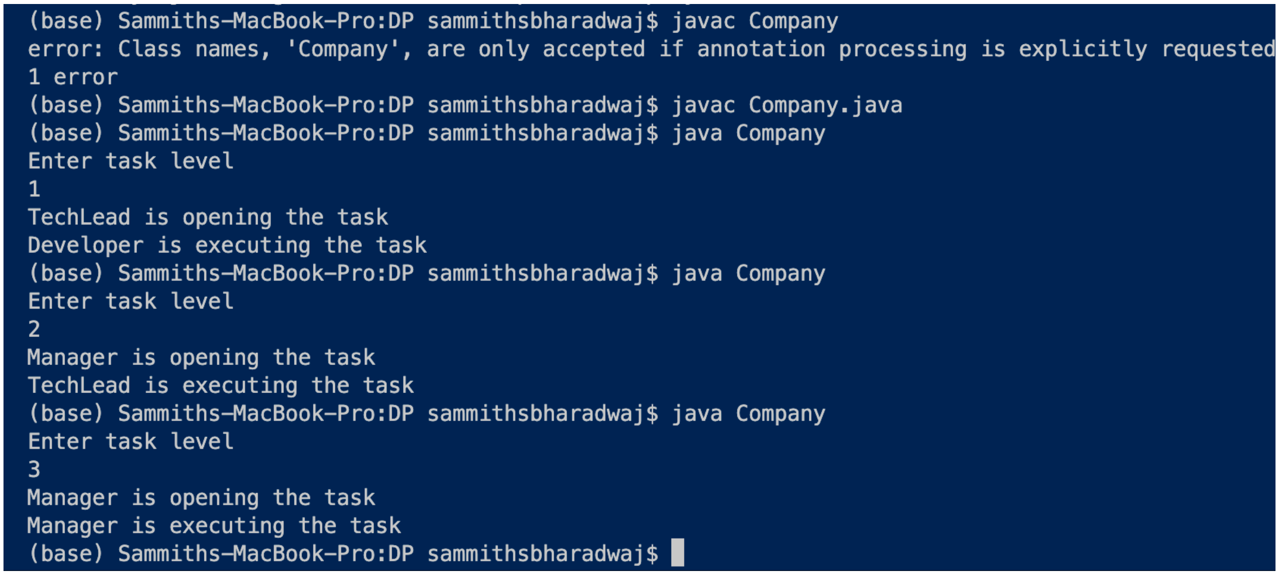
Creation pattern:

1) Factory – the employee interface provides a common ground which allows the company to add more types of employees apart from developer, techlead and manager without having to modify the base class.

Behavioral pattern:

1) Chain of responsibility – the request passes from the employee object to the task object
2) State – the state of the task object changes depending on what employee is assigned to the task.

Output Screen Shot:



```
(base) Sammiths-MacBook-Pro:DP sammithsbharadwaj$ javac Company
error: Class names, 'Company', are only accepted if annotation processing is explicitly requested
1 error
(base) Sammiths-MacBook-Pro:DP sammithsbharadwaj$ javac Company.java
(base) Sammiths-MacBook-Pro:DP sammithsbharadwaj$ java Company
Enter task level
1
TechLead is opening the task
Developer is executing the task
(base) Sammiths-MacBook-Pro:DP sammithsbharadwaj$ java Company
Enter task level
2
Manager is opening the task
TechLead is executing the task
(base) Sammiths-MacBook-Pro:DP sammithsbharadwaj$ java Company
Enter task level
3
Manager is opening the task
Manager is executing the task
(base) Sammiths-MacBook-Pro:DP sammithsbharadwaj$ 
```

Code:

1) Company.java

```
import java.util.Scanner;

public class Company {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        int level;
        System.out.println("Enter task level");
        level = scan.nextInt();
        Task task = new Task("BackDev",
        "Moderate",  "In Progress",level,
         "Medium", "2 days");
        Employee techlead = new TechLead();
        Employee developer = new Developer();
        Employee manager = new Manager();
        scan.close();
        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);
        }
    }
}
```

2) Task.java

```
public class Task {
    String taskName;
    String taskType;
    String taskStatus;
    int taskLevel;
    String taskSize;
    String effortRequired;
    public Task() {
        taskName = "Task1";
        taskType = "TaskType1";
        taskStatus = "TaskStatus1";
        taskLevel = 1;
        taskSize = "TaskSize1";
        effortRequired = "EffortRequired1";
    }

    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();
    }

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

3) Employee.java

```
public interface Employee {
    public void executedBy();

    public void openedBy();
}
```

4) Manager.java

```
public class Manager implements Employee {
    public void executedBy() {
        System.out.println("Manager is executing the task");
    }

    public void openedBy() {
        System.out.println("Manager is opening the task");
    }
}
```

5) Developer.java

```
public class Developer implements Employee {
    public void openedBy() {
        System.out.println("Developer is opening the task");
    }

    public void executedBy() {
        System.out.println("Developer is executing the task");
    }
}
```

6) TechLead.java

```
public class TechLead implements Employee {
    public void openedBy() {
        System.out.println("TechLead is opening the task");
    }

    public void executedBy() {
        System.out.println("TechLead is executing the task");
    }
}
```