SPRINT PLANNER
DESCRIPTION
DESIGN A SPRINT PLANNER (SIMILAR TO JIRA). A SPRINT PLANNER IS USED TO ASSIGN THE TASK TO THE TEAM MEMBER AND CHECK THE PROGRESS OF THE PROJECT OF EACH TEAMMATE WHICH IS ASSIGNED TO HIM/HER.
● SPRINT: THIS IS A TIME PERIOD IN WHICH THE COMPANY ASSIGNS A TASK TO THE EMPLOYEE/TEAM TO COMPLETE A SET AMOUNT OF WORK.
● TASK: THIS COMES INSIDE THE SPRINT, WHERE ANY NEW FEATURE OF THE PROJECT IS BROKEN DOWN INTO A TASK, WHICH IS ASSIGNED TO THE TEAM MEMBER.
FUNCTIONALITIES OF THE SPRINT PLANNER:
1. USERS SHOULD BE ABLE TO CREATE ANY SPRINT. SHOULD BE ABLE TO ADD ANY TASK TO THE SPRINT AND REMOVE FROM IT.++
2. USERS SHOULD BE ABLE TO CREATE A TASK OF TYPE STORY, FEATURE, BUG. EACH CAN HAVE THEIR OWN STATUS (TODO, INPROGRESS, DONE).
3. USERS SHOULD BE ABLE TO CHANGE THE STATUS OF A TASK.
4. SHOW TASKS ASSIGNED TO A USER IN THE SPRINT.
5. SHOW TYPES TASKS IN THE SPRINT.
CONSTRAINTS:
● USERS CAN HAVE MAX 2 TASKS IN INPROGRESS STATUS IN A SPRINT.

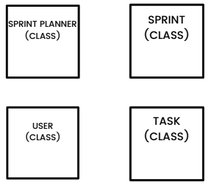
SPRINT CAN HAVE MAX 20 TASKS.
● FOLLOWING STATUS CHANGES ARE ALLOWED (TODO -> INPROGRESS, INPROGRESS->TODO, INPROGRESS-> DONE).
EXPECTATIONS:
1. MAKE SURE THAT YOU HAVE A WORKING AND DEMONSTRABLE CODE
2. MAKE SURE THAT THE CODE IS FUNCTIONALLY CORRECT
3. MAKE SURE THAT THE CODE IS READABLE AND MODULAR (CODE QUALITY) 4. MAKE SURE THAT THE CODE IS THREAD SAFE

SKETCH OUT

CLASS
PROPERTIES
ENUM

```
import java.util.ArrayList;
import java.util.List;

enum TaskType {
    STORY,
    FEATURE,
    BUG
}

enum TaskStatus {
    TODO,
    INPROGRESS,
    DONE
}

public class Main {
    public static void main(String[] args) {
        Sprint sprint1 = new Sprint("Sprint 1");

        Task task1 = new Task("Implement login feature", TaskType.FEATURE);
        Task task2 = new Task("Fix authentication bug", TaskType.BUG);
        Task task3 = new Task("Write user stories", TaskType.STORY);

        sprint1.addTask(task1);
        sprint1.addTask(task2);
        sprint1.addTask(task3);

        User user1 = new User("Alice");
        user1.assignTask(task1);
        user1.assignTask(task2);
        user1.assignTask(task3);// This should print "Maximum number of tasks in progress reached for this user"
    }
}

class Task {
    private String name;
    private TaskType type;
    private TaskStatus status;

    public Task(String name, TaskType type) {
        this.name = name;
        this.type = type;
        this.status = TaskStatus.TODO;
    }

    public void changeStatus(TaskStatus newStatus) {
        // Implement logic to check allowed status transitions
        if ((status == TaskStatus.TODO && newStatus == TaskStatus.INPROGRESS) ||
            (status == TaskStatus.INPROGRESS && (newStatus == TaskStatus.TODO || newStatus == TaskStatus.DONE)) ||
            (status == TaskStatus.DONE && newStatus == TaskStatus.INPROGRESS)) {
            status = newStatus;
        } else {
            System.out.println("Invalid status transition");
        }
    }

    public String getName() {
        return name;
    }

    public TaskType getType() {
        return type;
    }hk

    public TaskStatus getStatus() {
        return status;
    }
}

class Sprint {
    private String name;
    private List<Task> tasks;

    public Sprint(String name) {
        this.name = name;
        this.tasks = new ArrayList<>();
    }

    public void addTask(Task task) {
        // Check if the maximum number of tasks allowed in the sprint is reached
        if (tasks.size() < 20) {
            tasks.add(task);
        } else {
            System.out.println("Maximum number of tasks reached for this Sprint");
        }
    }

    // Other Sprint functionalities can be implemented here
}

class User {
    private String name;
    private List<Task> tasksinProgress;

    public User(String name) {
        this.name = name;
        this.tasksinProgress = new ArrayList<>();
    }

    public void assignTask(Task task) {
        // Check if the maximum number of tasks in progress for the user is reached
        if (tasksinProgress.size() < 2) {
            tasksinProgress.add(task);
            task.changeStatus(TaskStatus.INPROGRESS);
        } else {
            System.out.println("Maximum number of tasks in progress reached for this user");
        }
    }

    // Other User functionalities can be implemented here
}
```

STEP 2 (DEFINE CLASSES)



STEP 2 (DEFINE FUNCTIONS)