Universidad de las Fuerzas Armadas ESPE

Object Oriented Programming

NRC: 14575

Homerwork #5

Topic: Project funding and object modeling

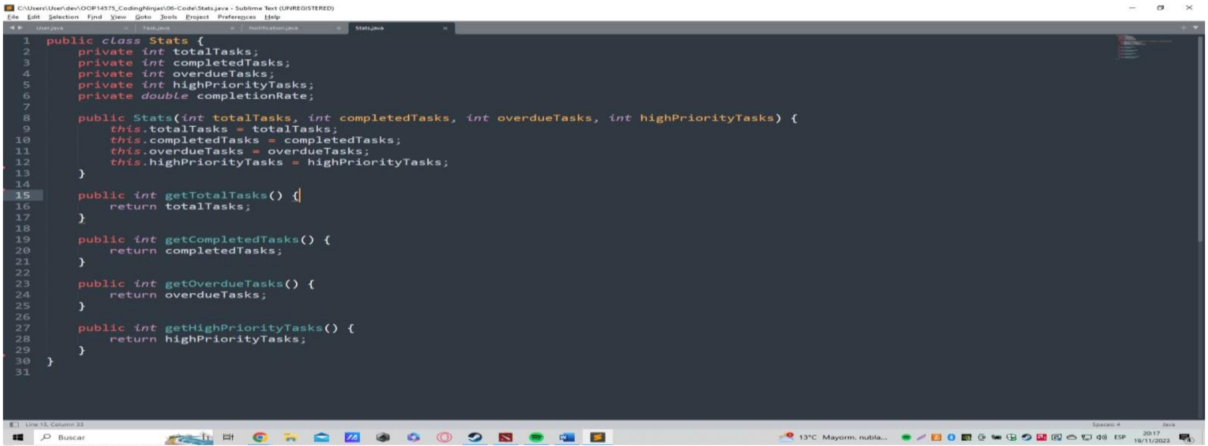
Teacher: Edison Lascano

Student: Antonio Revilla

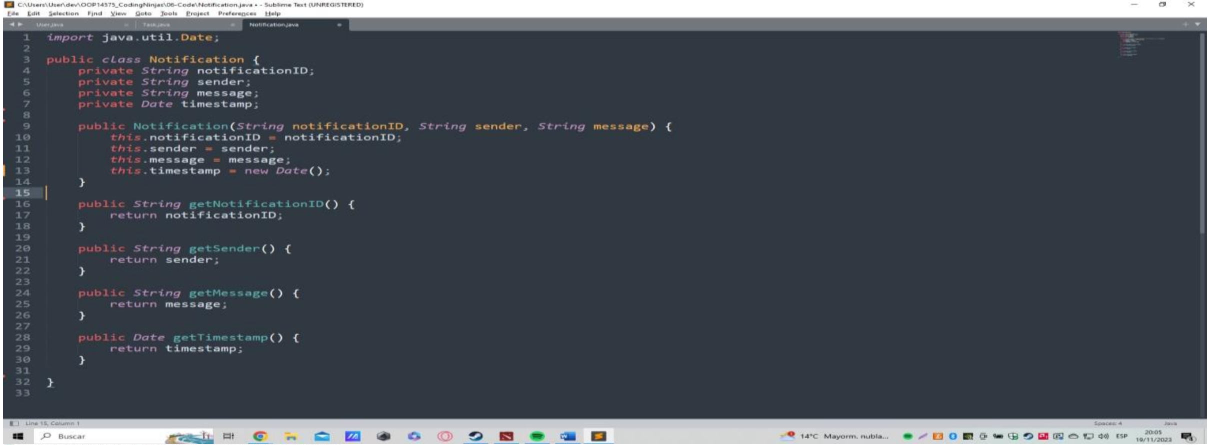
NAME	Selected (✓)	Not selected (✕)
1. Task	✓	
2. Users	✓	
3. Types of tasks	✓	
4. Reminders	✓	
5. Daily messages	✓	
6. Daily Agenda		✕
7. Calendar		✕
8. Reporting		✕
9. Graphs		✕
10. Challenges		✕
11. Notifications	✓	
12. System		✕
13. Creation Date		✕
14. Specific Projects		✕
15. Phases		✕
16. Priorities		✕
17. Workflow		✕
18. Projects	✓	
19. Stats	✓	
20. Final Task		✕
21. Task		✕
22. Integration		✕
23. Daily Status		✕
24. Methodologies		✕
25. Teams		✕

26. Roles	✓	
27. Individuals		✕
28. Organizations	✓	
29. Management applications		✕
30. Graph		✕

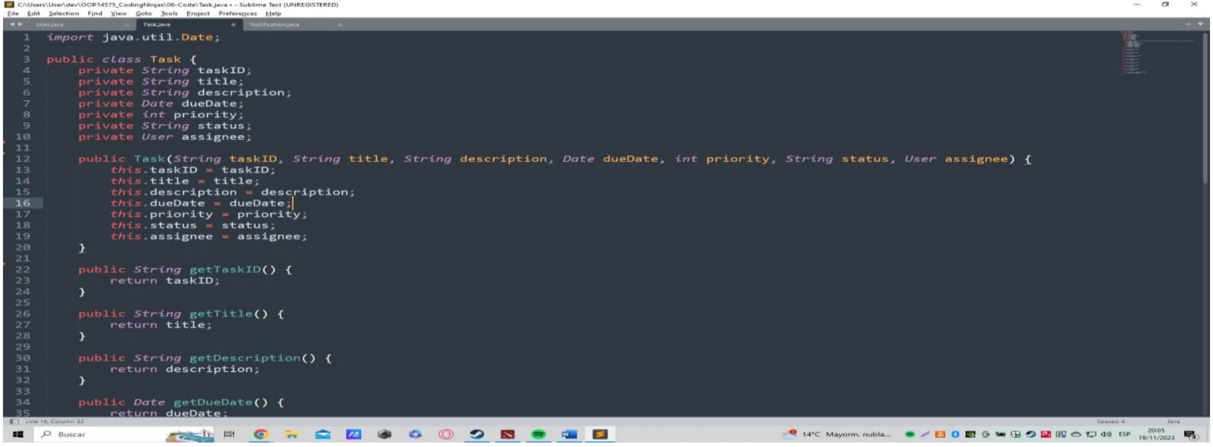
Code:


C:\Users\User\dev\OOP14575_CodingNinjas\06-Code\Stats.java - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
User.java
Task.java
Notification.java
Stats.java
Line 15, Column 33
Buscar

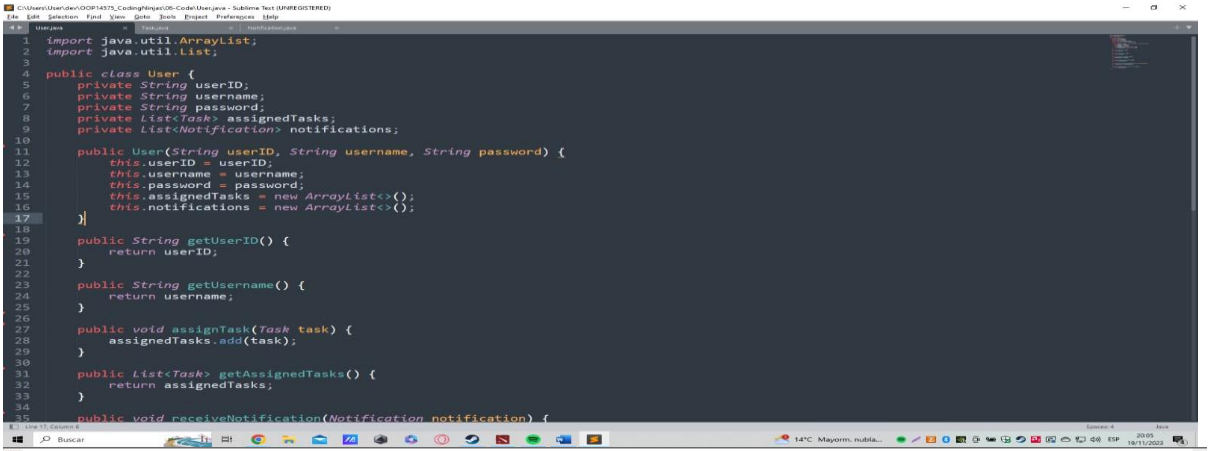
```java
public class Stats {
    private int totalTasks;
    private int completedTasks;
    private int overdueTasks;
    private int highPriorityTasks;
    private double completionRate;

    public Stats(int totalTasks, int completedTasks, int overdueTasks, int highPriorityTasks) {
        this.totalTasks = totalTasks;
        this.completedTasks = completedTasks;
        this.overdueTasks = overdueTasks;
        this.highPriorityTasks = highPriorityTasks;
    }

    public int getTotalTasks() {
        return totalTasks;
    }

    public int getCompletedTasks() {
        return completedTasks;
    }

    public int getOverdueTasks() {
        return overdueTasks;
    }

    public int getHighPriorityTasks() {
        return highPriorityTasks;
    }
}
```


C:\Users\User\dev\OOP14575_CodingNinjas\06-Code\Notification.java • - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
User.java
Task.java
Notification.java
Line 15, Column 1
Buscar

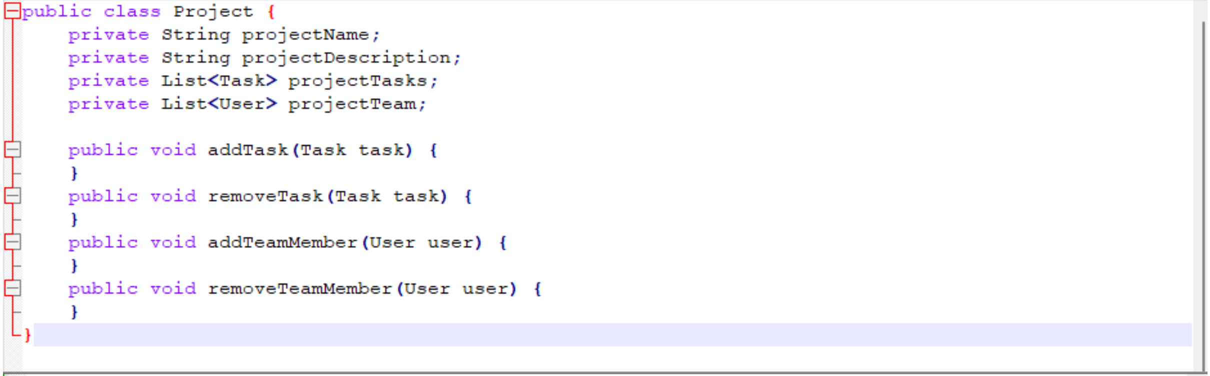
```java
import java.util.Date;

public class Notification {
    private String notificationID;
    private String sender;
    private String message;
    private Date timestamp;

    public Notification(String notificationID, String sender, String message) {
        this.notificationID = notificationID;
        this.sender = sender;
        this.message = message;
        this.timestamp = new Date();
    }

    public String getNotificationID() {
        return notificationID;
    }

    public String getSender() {
        return sender;
    }

    public String getMessage() {
        return message;
    }

    public Date getTimestamp() {
        return timestamp;
    }

}
```


C:\Users\User\dev\OOP14575_CodingNinjas\06-Code\Task.java • - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
User.java
Task.java
Notification.java
Line 16, Column 32
Buscar

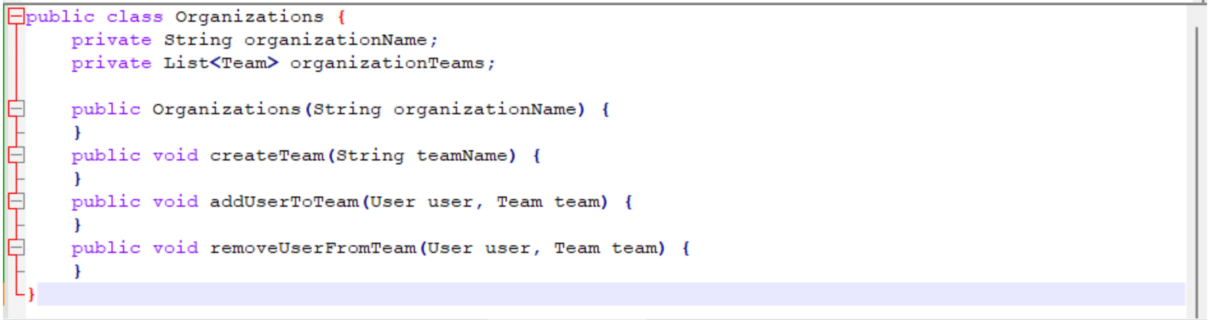
```java
import java.util.Date;

public class Task {
    private String taskID;
    private String title;
    private String description;
    private Date dueDate;
    private int priority;
    private String status;
    private User assignee;

    public Task(String taskID, String title, String description, Date dueDate, int priority, String status, User assignee) {
        this.taskID = taskID;
        this.title = title;
        this.description = description;
        this.dueDate = dueDate;
        this.priority = priority;
        this.status = status;
        this.assignee = assignee;
    }

    public String getTaskID() {
        return taskID;
    }

    public String getTitle() {
        return title;
    }

    public String getDescription() {
        return description;
    }

    public Date getDueDate() {
        return dueDate;
```


C:\Users\User\dev\OOP14575_CodingNinjas\06-Code\User.java - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
User.java
Task.java
Notification.java
Line 17, Column 6
Buscar
14°C Mayorm. nubla...
ESP
20:05
19/11/2023

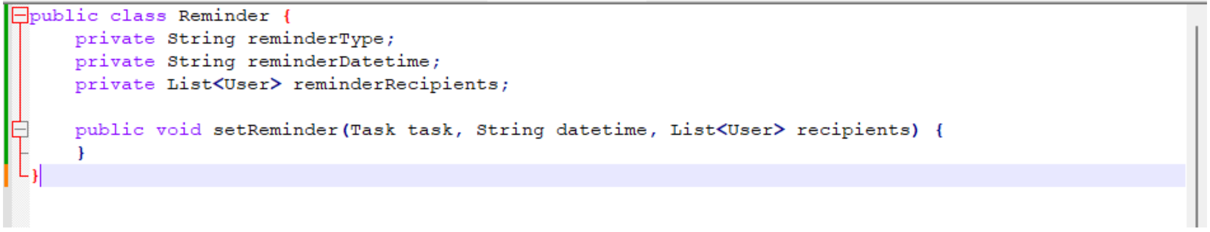
```java
import java.util.ArrayList;
import java.util.List;

public class User {
    private String userID;
    private String username;
    private String password;
    private List<Task> assignedTasks;
    private List<Notification> notifications;

    public User(String userID, String username, String password) {
        this.userID = userID;
        this.username = username;
        this.password = password;
        this.assignedTasks = new ArrayList<>();
        this.notifications = new ArrayList<>();
    }

    public String getUserID() {
        return userID;
    }

    public String getUsername() {
        return username;
    }

    public void assignTask(Task task) {
        assignedTasks.add(task);
    }

    public List<Task> getAssignedTasks() {
        return assignedTasks;
    }

    public void receiveNotification(Notification notification) {
```



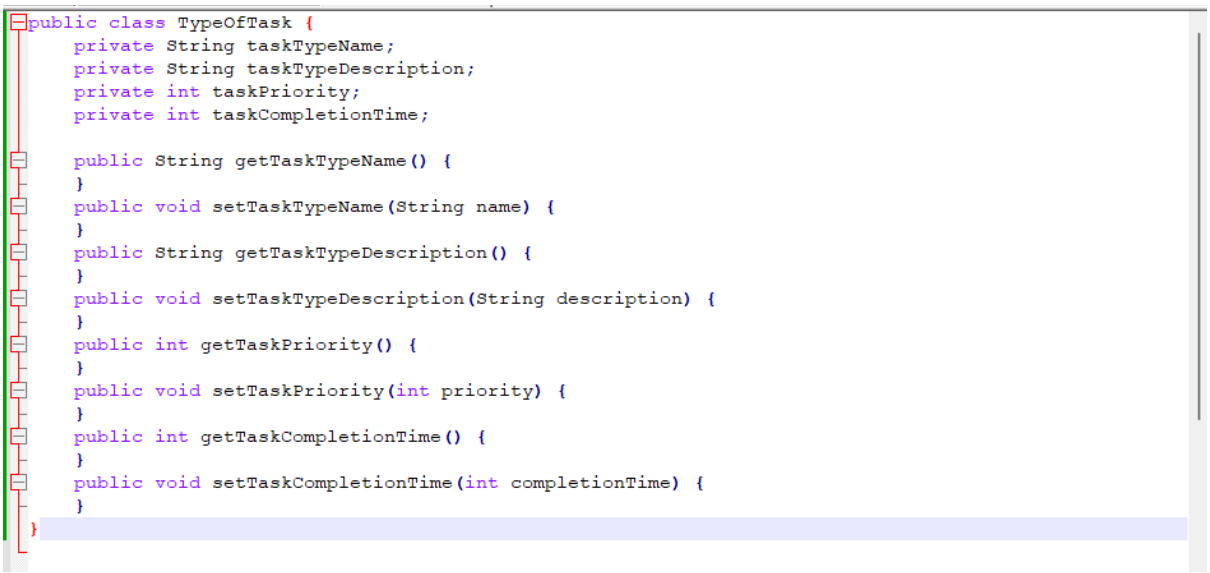
```java
public class Project {
    private String projectName;
    private String projectDescription;
    private List<Task> projectTasks;
    private List<User> projectTeam;

    public void addTask(Task task) {
    }
    public void removeTask(Task task) {
    }
    public void addTeamMember(User user) {
    }
    public void removeTeamMember(User user) {
    }
}
```



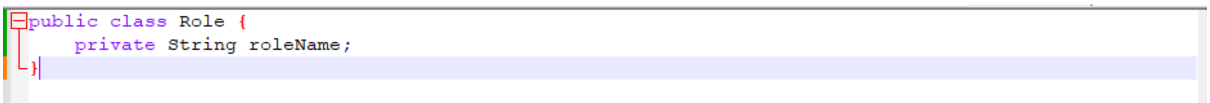
```java
public class Organizations {
    private String organizationName;
    private List<Team> organizationTeams;

    public Organizations(String organizationName) {
    }
    public void createTeam(String teamName) {
    }
    public void addUserToTeam(User user, Team team) {
    }
    public void removeUserFromTeam(User user, Team team) {
    }
}
```



```java
public class Reminder {
    private String reminderType;
    private String reminderDatetime;
    private List<User> reminderRecipients;

    public void setReminder(Task task, String datetime, List<User> recipients) {
    }
}
```

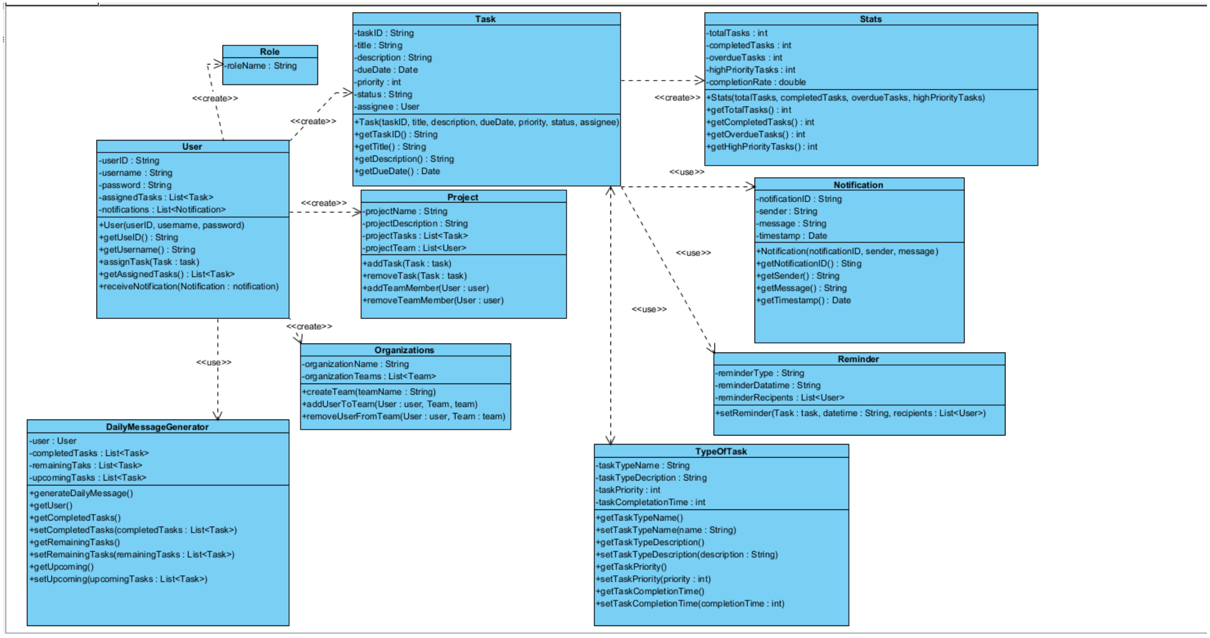


```java
public class TypeOfTask {
    private String taskTypeName;
    private String taskTypeDescription;
    private int taskPriority;
    private int taskCompletionTime;

    public String getTaskTypeName() {
    }
    public void setTaskTypeName(String name) {
    }
    public String getTaskTypeDescription() {
    }
    public void setTaskTypeDescription(String description) {
    }
    public int getTaskPriority() {
    }
    public void setTaskPriority(int priority) {
    }
    public int getTaskCompletionTime() {
    }
    public void setTaskCompletionTime(int completionTime) {
    }
}
```



```java
public class Role {
    private String roleName;
}
```

UML Diagram:


Role
-roleName : String
<<create>>
Task
-taskID : String
-title : String
-description : String
-dueDate : Date
-priority : int
-status : String
-assignee : User
+Task(taskID, title, description, dueDate, priority, status, assignee)
+getTaskID() : String
+getTitle() : String
+getDescription() : String
+getDueDate() : Date
<<create>>
Stats
-totalTasks : int
-completedTasks : int
-overdueTasks : int
-highPriorityTasks : int
-completionRate : double
+Stats(totalTasks, completedTasks, overdueTasks, highPriorityTasks)
+getTotalTasks() : int
+getCompletedTasks() : int
+getOverdueTasks() : int
+getHighPriorityTasks() : int
<<create>>
User
-userID : String
-username : String
-password : String
-assignedTasks : List<Task>
-notifications : List<Notification>
+User(userID, username, password)
+getUseID() : String
+getUsername() : String
+assignTask(Task : task)
+getAssignedTasks() : List<Task>
+receiveNotification(Notification : notification)
<<use>>
<<create>>
Project
-projectName : String
-projectDescription : String
-projectTasks : List<Task>
-projectTeam : List<User>
+addTask(Task : task)
+removeTask(Task : task)
+addTeamMember(User : user)
+removeTeamMember(User : user)
Notification
-notificationID : String
-sender : String
-message : String
-timestamp : Date
+Notification(notificationID, sender, message)
+getNotificationID() : Sting
+getSender() : String
+getMessage() : String
+getTimestamp() : Date
<<use>>
<<use>>
<<create>>
<<use>>
Organizations
-organizationName : String
-organizationTeams : List<Team>
+createTeam(teamName : String)
+addUserToTeam(User : user, Team, team)
+removeUserFromTeam(User : user, Team : team)
Reminder
-reminderType : String
-reminderDatatime : String
-reminderRecipents : List<User>
+setReminder(Task : task, datetime : String, recipients : List<User>)
DailyMessageGenerator
-user : User
-completedTasks : List<Task>
-remainingTaks : List<Task>
-upcomingTasks : List<Task>
+generateDailyMessage()
+getUser()
+getCompletedTasks()
+setCompletedTasks(completedTasks : List<Task>)
+getRemainingTasks()
+setRemainingTasks(remainingTasks : List<Task>)
+getUpcoming()
+setUpcoming(upcomingTasks : List<Task>)
TypeOfTask
-taskTypeName : String
-taskTypeDecription : String
-taskPriority : int
-taskCompletationTime : int
+getTaskTypeName()
+setTaskTypeName(name : String)
+getTaskTypeDescription()
+setTaskTypeDescription(description : String)
+getTaskPriority()
+setTaskPriority(priority : int)
+getTaskCompletionTime()
+setTaskCompletionTime(completionTime : int)