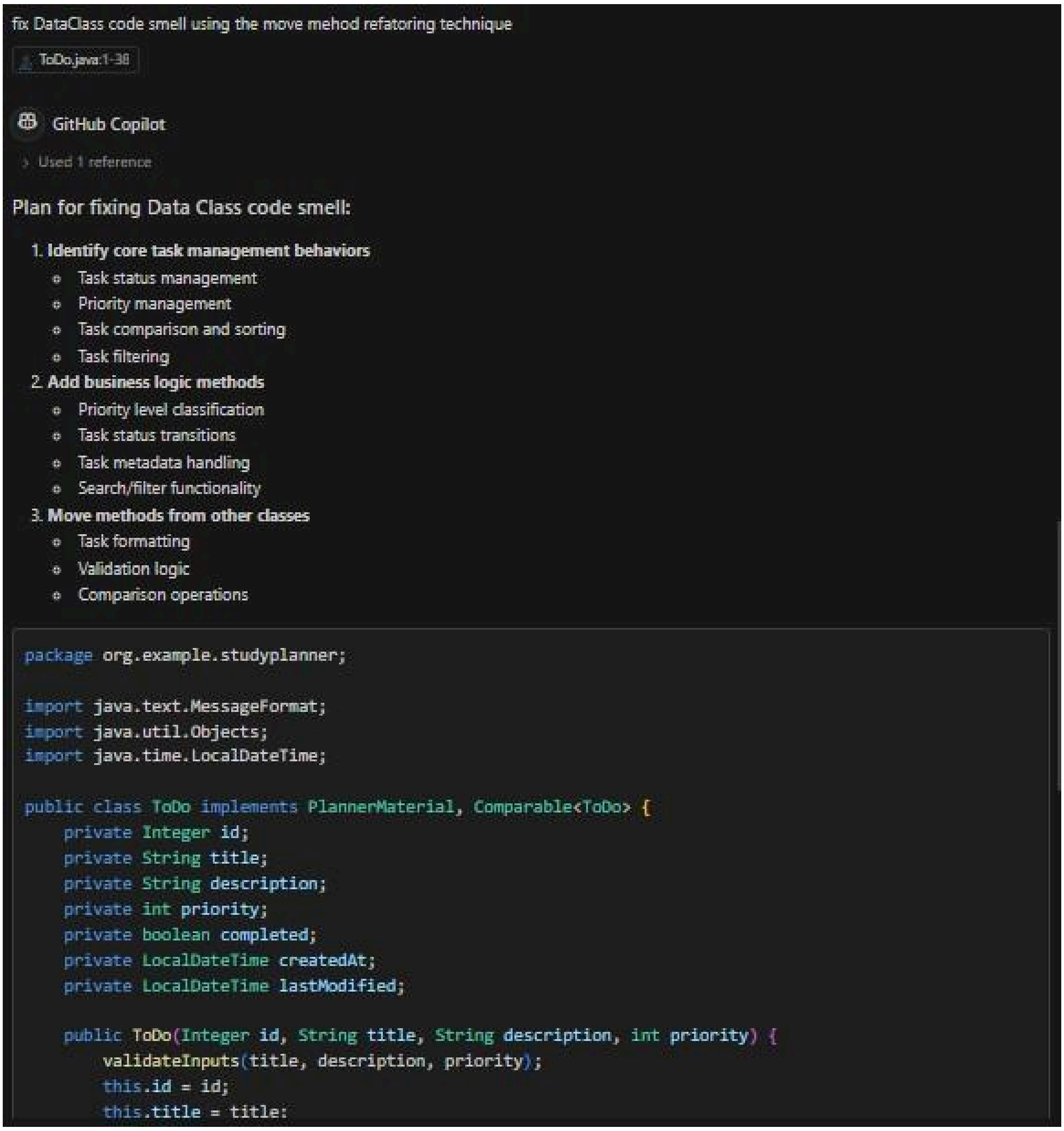
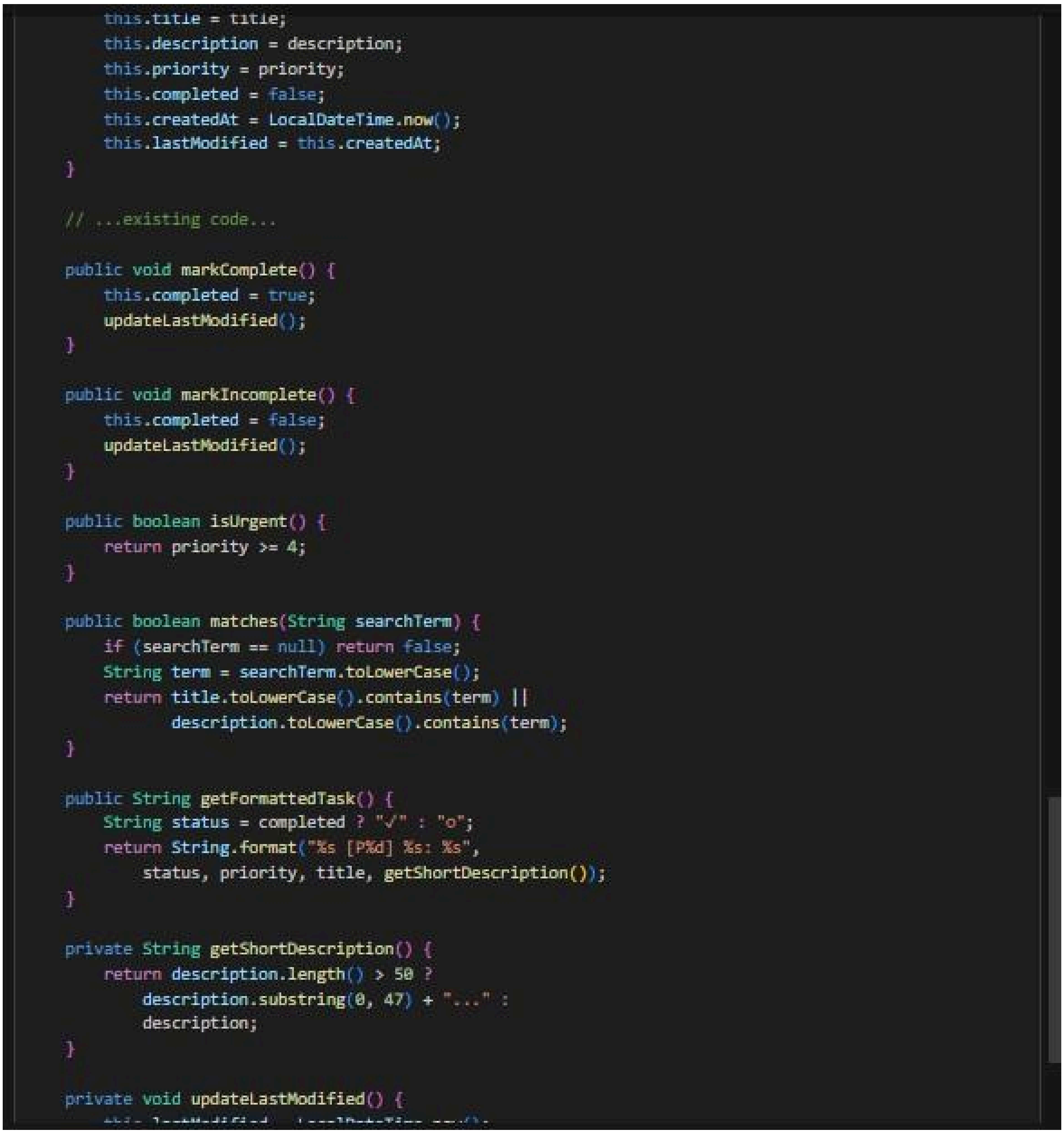


fix DataClass code smell using the move mehod refatoring technique

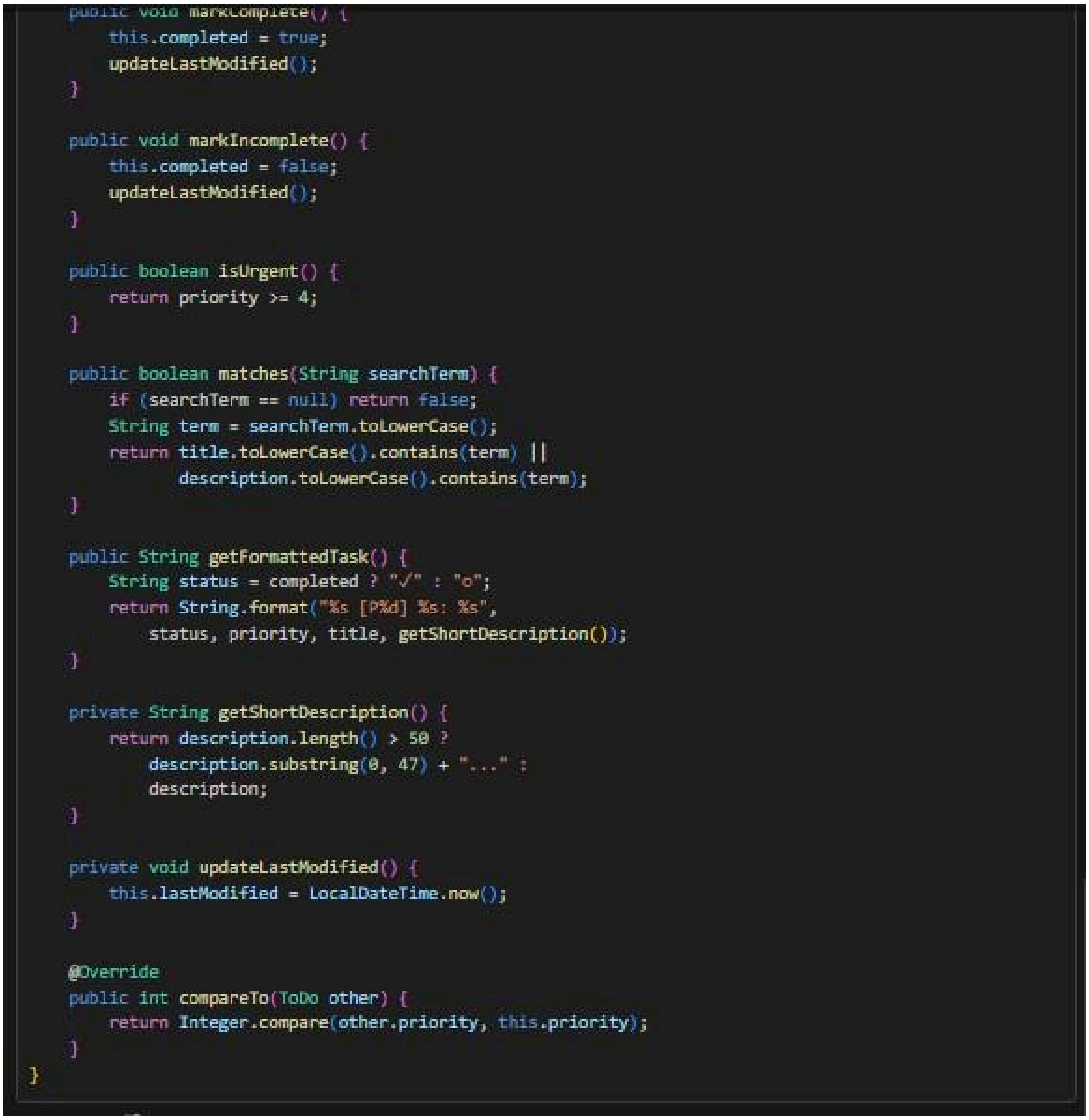
Plan for fixing Data Class code smell:

1. **Identify core task management behaviors**
   - Task status management
   - Priority management
   - Task comparison and sorting
   - Task filtering
2. **Add business logic methods**
   - Priority level classification
   - Task status transitions
   - Task metadata handling
   - Search/filter functionality
3. **Move methods from other classes**
   - Task formatting
   - Validation logic
   - Comparison operations

```java
package org.example.studyplanner;

import java.text.MessageFormat;
import java.util.Objects;
import java.time.LocalDateTime;

public class ToDo implements PlannerMaterial, Comparable<ToDo> {
    private Integer id;
    private String title;
    private String description;
    private int priority;
    private boolean completed;
    private LocalDateTime createdAt;
    private LocalDateTime lastModified;

    public ToDo(Integer id, String title, String description, int priority) {
        validateInputs(title, description, priority);
        this.id = id;
        this.title = title;
```



```
        this.title = title;
        this.description = description;
        this.priority = priority;
        this.completed = false;
        this.createdAt = LocalDateTime.now();
        this.lastModified = this.createdAt;
    }

    // ...existing code...

    public void markComplete() {
        this.completed = true;
        updateLastModified();
    }

    public void markIncomplete() {
        this.completed = false;
        updateLastModified();
    }

    public boolean isUrgent() {
        return priority >= 4;
    }

    public boolean matches(String searchTerm) {
        if (searchTerm == null) return false;
        String term = searchTerm.toLowerCase();
        return title.toLowerCase().contains(term) ||
               description.toLowerCase().contains(term);
    }

    public String getFormattedTask() {
        String status = completed ? "✓" : "o";
        return String.format("%s [P%d] %s: %s",
            status, priority, title, getShortDescription());
    }

    private String getShortDescription() {
        return description.length() > 50 ?
            description.substring(0, 47) + "..." :
            description;
    }

    private void updateLastModified() {
```



```java
    public void markComplete() {
        this.completed = true;
        updateLastModified();
    }

    public void markIncomplete() {
        this.completed = false;
        updateLastModified();
    }

    public boolean isUrgent() {
        return priority >= 4;
    }

    public boolean matches(String searchTerm) {
        if (searchTerm == null) return false;
        String term = searchTerm.toLowerCase();
        return title.toLowerCase().contains(term) ||
               description.toLowerCase().contains(term);
    }

    public String getFormattedTask() {
        String status = completed ? "✓" : "o";
        return String.format("%s [P%d] %s: %s",
            status, priority, title, getShortDescription());
    }

    private String getShortDescription() {
        return description.length() > 50 ?
            description.substring(0, 47) + "..." :
            description;
    }

    private void updateLastModified() {
        this.lastModified = LocalDateTime.now();
    }

    @Override
    public int compareTo(ToDo other) {
        return Integer.compare(other.priority, this.priority);
    }
}
```