

FEATURE ENVY

MaterialSearch

```
package org.example.studysearch;
```

```
import org.example.studyregistry.StudyMaterial;
```

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

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

```
public class MaterialSearch implements Search<String>{
```

```
    private SearchLog searchLog = new
```

```
    SearchLog("Material Search");
```

```
    public MaterialSearch() {}
```

```
    @Override
```

```
    public List<String> search(String text) {
```

```
        return handleMaterialSearch(text);
```

```
    }
```

```
    public SearchLog getSearchLog() {
```

```
        return searchLog;
```

```
    }
```

```
    private List<String> handleMaterialSearch(String text) {
```

```
        List<String> results = new ArrayList<>();
```

```
        results.addAll(StudyMaterial.getStudyMaterial().searchInMaterials(text));
```

```
        this.searchLog.addSearchHistory(text);
```

```
        this.searchLog.setNumUsages(this.searchLog.getNumUsages() + 1);
```

```
        results.add("\nLogged in: " +
```

```
        this.searchLog.getLogName());
```

```
        return results;
```

```
    }
```

```
}
```

```
package org.example.studysearch;
```

```
import org.example.studycards.CardManager;
```

```
import org.example.studyplanner.HabitTracker;
```

```
import org.example.studyplanner.TodoTracker;
```

```
import org.example.studyregistry.StudyMaterial;
```

```
import org.example.studyregistry.StudyTaskManager;
```

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

```
import java.util.HashMap;
```

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

```
import java.util.Map;
```

```
public class SearchLog {
```

```
    private List<String> searchHistory;
```

```
    private Map<String, Integer> searchCount;
```

```
    private boolean isLocked;
```

```
    private Integer numUsages;
```

```
    private String logName;
```

```
    public SearchLog(String logName) {
```

```
        searchHistory = new ArrayList<>();
```

```
        searchCount = new HashMap<>();
```

```
        this.logName = logName;
```

```
        numUsages = 0;
```

```
        isLocked = false;
```

```
    }
```

```
    public List<String> search(String text) {
```

```
        List<String> results = new ArrayList<>();
```

```
        results.addAll(CardManager.getCardManager().searchInCards(text));
```

```
        results.addAll(HabitTracker.getHabitTracker().searchInHabits(text));
```

```
        results.addAll(TodoTracker.getTodoTracker().searchInTodos(text));
```

```
        results.addAll(StudyMaterial.getStudyMaterial().searchInMaterials(text));
```

```
        results.addAll(StudyTaskManager.getStudyTaskManager().searchInRegistries(text));
```

```
        addSearchHistory(text);
```

```
        setNumUsages(getNumUsages() + 1);
```

```
        results.add("\nLogged in: " + getLogName());
```

```
        return results;
```

```
    }
```

```
// Moved method from MaterialSearch class
```

```
    public List<String> handleMaterialSearch(String text) {
```

```
        List<String> results = new ArrayList<>();
```

```
        results.addAll(StudyMaterial.getStudyMaterial().searchInMaterials(text));
```

```
        addSearchHistory(text);
```

```
        setNumUsages(getNumUsages() + 1);
```

```
        results.add("\nLogged in: " + getLogName());
```

```
        return results;
```

```
    }
```

```
    public void addSearchHistory(String searchHistory) {
```

```
        this.searchHistory.add(searchHistory);
```

```
    }
```

```
    public List<String> getSearchHistory() {
```

```
        return searchHistory;
```

```
    }
```

```
    public Map<String, Integer> getSearchCount() {
```

```
        return searchCount;
```

```
    }
```

```
    public boolean isLocked() {
```

```
        return isLocked;
```

```
    }
```

```
    public void setLocked(boolean locked) {
```

```
        isLocked = locked;
```

```
    }
```

```
    public Integer getNumUsages() {
```

```
        return numUsages;
```

```
    }
```

```
    public void setNumUsages(Integer numUsages) {
```

```
        this.numUsages = numUsages;
```

```
    }
```

```
    public String getLogName() {
```

```
        return logName;
```

```
    }
```

```
    public void setLogName(String logName) {
```

```
        this.logName = logName;
```

```
    }
```

```
fix the feature envy code smell using the move method
```

```
refactoring technique
```

◆ The MaterialSearch class can now be modified to delegate the search functionality to the SearchLog class:

```
print("```java")
```

```
print("package org.example.studysearch;")
```

```
import org.example.studyregistry.StudyMaterial;
```

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

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

```
public class MaterialSearch implements Search<String>{
```

```
    private SearchLog searchLog = new SearchLog("Material Search");
```

```
    public MaterialSearch() {}
```

```
    @Override
```

```
    public List<String> search(String text) {
```

```
        // Delegate search to Searchlog class
```

```
        return searchLog.handleMaterialSearch(text);
```

There seems to be a syntax error in the generated code. I'll fix the indentation and provide the refactored code again.

Sure, the Feature Envy code smell can be refactored using the Move Method refactoring technique as follows:

The move method refactoring technique involves identifying a method that belongs more to another class and moving it to that class.

In the given code, the handleMaterialSearch method seems to have feature envy for the SearchLog class. This method uses several methods of the SearchLog class (addSearchHistory, setNumUsages) and doesn't seem to be tightly coupled with the MaterialSearch class.

We can move the handleMaterialSearch method to the SearchLog class as follows:

```
Java
```

```
package org.example.studysearch;
```

```
import org.example.studyregistry.StudyMaterial;
```

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

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

```
public class MaterialSearch implements Search<String>{
```

```
    private SearchLog searchLog = new SearchLog("Material Search");
```

```
    public MaterialSearch() {}
```

```
    @Override
```

```
    public List<String> search(String text) {
```

```
        // Delegate search to Searchlog class
```

```
        return searchLog.handleMaterialSearch(text);
```

```
    }
```

```
    public void addSearchHistory(String searchHistory) {
```

```
        this.searchHistory.add(searchHistory);
```

```
    }
```

```
    public List<String> getSearchHistory() {
```

```
        return searchHistory;
```

```
    }
```

```
    public Map<String, Integer> getSearchCount() {
```

```
        return searchCount;
```

```
    }
```

```
    public boolean isLocked() {
```

```
        return isLocked;
```

```
    }
```

```
    public void setLocked(boolean locked) {
```

```
        isLocked = locked;
```

```
    }
```

```
    public Integer getNumUsages() {
```

```
        return numUsages;
```

```
    }
```

```
    public void setNumUsages(Integer numUsages) {
```

```
        this.numUsages = numUsages;
```

```
    }
```

```
    public String getLogName() {
```

```
        return logName;
```

```
    }
```

```
    public void setLogName(String logName) {
```

```
        this.logName = logName;
```

```
    }
```

By moving the handleMaterialSearch method to the SearchLog class, we improve the cohesion of the SearchLog class and reduce the coupling between the MaterialSearch class and the SearchLog class. This makes the code easier to understand and maintain.

◆ ◇ ◈ :