

fix the feature envy code smell using move method refactoring technique

```
public class RegistrySearch implements Search<String> { private SearchLog searchLog = new SearchLog("Registry Search"); public RegistrySearch() {}
```

```
    @Override
    public List<String> search(String text) {
        return handleRegistrySearch(text);
    }

    public SearchLog getSearchLog() {
        return searchLog;
    }

    private List<String> handleRegistrySearch(String text){
        List<String> results = new ArrayList<>();
        results.addAll(CardManager.getCardManager().searchInCards(text));
        results.addAll(HabitTracker.getHabitTracker().searchInHabits(text));
        results.addAll(TodoTracker.getInstance().searchInTodos(text));
        results.addAll(StudyTaskManager.getStudyTaskManager().searchInRegistries(text));
        this.searchLog.addSearchHistory(text);
        this.searchLog.setNumUsages(this.searchLog.getNumUsages() + 1);
        results.add("\nLogged in: " + this.searchLog.getLogName());
        return results;
    }
}
```

```
public class SearchLog { private final List<String> searchHistory; private final Map<String, Integer> searchCount; private boolean isLocked; private Integer numUsages; private String logName;
```

```
    public SearchLog(String logName) {
        if (logName == null || logName.trim().isEmpty()) {
            throw new IllegalArgumentException("Log name cannot be empty");
        }
        this.searchHistory = new ArrayList<>();
        this.searchCount = new HashMap<>();
        this.logName = logName;
        this.numUsages = 0;
        this.isLocked = false;
    }

    public void addSearchHistory(String searchTerm) {
        if (isLocked) {
            throw new IllegalStateException("Cannot add search to locked log");
        }
        if (searchTerm == null || searchTerm.trim().isEmpty()) {
            throw new IllegalArgumentException("Search term cannot be empty");
        }
        searchHistory.add(searchTerm);
        searchCount.merge(searchTerm, 1, Integer::sum);
    }
}
```

Let's fix the feature envy code smell by moving the logging-related operations from `RegistrySearch` to `SearchLog`.

#### Step-by-step plan:

1. Identify code in `RegistrySearch` that works primarily with `SearchLog` data
2. Move the logging logic to `SearchLog` class
3. Simplify `RegistrySearch` to only handle search operations
4. Update method calls

#### Implementation:

```
public class RegistrySearch implements Search<String> {
    private SearchLog searchLog = new SearchLog("Registry Search");

    public RegistrySearch() {}

    @Override
    public List<String> search(String text) {
        List<String> results = handleRegistrySearch(text);
        searchLog.logSearch(text, results);
        return results;
    }

    public SearchLog getSearchLog() {
        return searchLog;
    }

    private List<String> handleRegistrySearch(String text) {
        List<String> results = new ArrayList<>();
        results.addAll(CardManager.getCardManager().searchInCards(text));
        results.addAll(HabitTracker.getHabitTracker().searchInHabits(text));
        results.addAll(TodoTracker.getInstance().searchInTodos(text));
        results.addAll(StudyTaskManager.getStudyTaskManager().searchInRegistries(text));
        return results;
    }
}
```

```
public class SearchLog {
    // ...existing code...

    public void logSearch(String searchTerm, List<String> results) {
        if (isLocked) {
            throw new IllegalStateException("Cannot add search to locked log");
        }
        addSearchHistory(searchTerm);
        setNumUsages(getNumUsages() + 1);
        results.add("\nLogged in: " + getLogName());
    }

    // ...existing code...
}
```

#### The changes:

1. Moved logging responsibilities to `SearchLog.logSearch()`
2. Simplified `RegistrySearch.search()` to focus on search operations
3. Used `SearchLog.logSearch()` to handle all logging operations
4. Removed direct manipulation of `SearchLog` properties from `RegistrySearch`

This refactoring improves encapsulation and reduces feature envy by keeping logging-related operations within the `SearchLog` class.

