



fix the code smell Feature envy using the move method

file LeitnerSystem.java:...

✦ Similarly, we can move the logic of building the response string

to a new method called **buildResponse**.

print("2. Move the logic of building the response string to a new method called buildResponse:")

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

```
class LeitnerSystem extends StudyMethod{
```

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

```
private String buildResponse(Integer randomCard, Card card) {
    String response = "[" + randomCard + "] ";
    response += "The random question was: " + card.getQuestion() + " | ";
    response += "The answer is: " + card.getAnswer();
    return response;
}
```

```
// ... other methods ...
```

```
}
```

```
# By applying these changes, we can improve the code readability and
# maintainability of the LeitnerSystem class. The getRandomCard method
# becomes more focused on its core responsibility of retrieving a random
# card from the provided boxes, and the responsibility of fetching cards
# from CardManager and building the response string is delegated to
# separate methods.
```

There seems to be

Você interrompeu esta resposta

file LeitnerSystem.java:...



The `buildResponse` method constructs the response message based on the card information. While it currently resides in the `LeitnerSystem` class, it does not inherently rely on any functionalities specific to `LeitnerSystem`. This suggests that `buildResponse` can be a method within the `Card` class itself.

However, directly moving the method to `Card.java` would break the current functionality. The `LeitnerSystem` class retrieves the card information (question and answer) from the `CardManager` class.

Here's how we can refactor the code to achieve the desired outcome:

Move the logic of building the response string to a new method named `buildResponseString` within the `Card` class:

Java



```
file Card.java
package org.example.studycards;

public class Card {
    private String question;
    private String answer;

    public Card(String question, String answer) {
        this.question = question;
        this.answer = answer;
    }

    public String getQuestion() {
        return question;
    }

    public void setQuestion(String question) {
        this.question = question;
    }

    public String getAnswer() {
        return answer;
    }

    public void setAnswer(String answer) {
        this.answer = answer;
    }

    public void edit(String question, String answer) {
        setQuestion(question);
        setAnswer(answer);
    }

    // New method to build the response string
    public String buildResponseString(int cardId) {
        return "[" + cardId + "] " + "The random question was: " + question + " | " + "The answ
```

 Gemini ▼[Testar o Gemini Advanced](#)

1.5 Flash

Modify the `LeitnerSystem` class to use the `buildResponseString` method of the `Card` object it retrieves from the `CardManager` :

Java



```
file LeitnerSystem.java
package org.example.studycards;

public class LeitnerSystem extends StudyMethod{
    // ... existing code ...

    private String buildResponse(Integer randomCard, Card card) {
        // Utilize the card's buildResponseString method
        return card.buildResponseString(randomCard);
    }

    // ... other methods ...
}
```

By following these steps, the `buildResponseString` method resides within the `Card` class, and the `LeitnerSystem` class can access it through the retrieved card object, improving code maintainability.



Powered by Gemini All Chat Downloader