DWA_07.4 Knowledge Check_DWA7

---

1. Which were the three best abstractions, and why?

   a) The css object:
   The CSS object is clear and concise. It defines color schemes for different themes, making it easy to understand. The object has a single responsibility, which means that it adheres to the SRP.
   The CSS object also adheres to the OCP (Open-Closed Principle) because it is open for extension but relatively closed for modification, by that I mean a developer can extend the object without modifying or changing the existing code should they wish to add more color themes.
   b) The createPreview function:
   The function that creates a book preview does not violate the SRP (Single Responsibility Principle), because the function only does one thing only.
   c) The handleShowMoreClick function:
   The function adheres to the SRP, it has a single responsibility, which is to handler the "show more" button on click events.
   The functions also align with the OCP, it does not require modification when the developer wants to trigger the rendering of more previews or change how it behaves in response to the "show more" button click event.

---

2. Which were the three worst abstractions, and why?

   a) The activeHandle function:
   The function violates the SRP, because the function has multiple responsibilities, it handles the activation of book preview and it also updates multiple HTML elements (title, images, just to name a few) based on the clicked book's information.
   The function also violates the OCP, in the sense that if a developer wishes to change how the book previews are activated or how the HTML elements are updated, then the developer would likely need to modify this function. So, the function is not designed to be open for extension, but closed for modification.
   The function also relies on a number of dependencies, these include html, books and authors, meaning that it violates the DIP.

b) The handleSearchFormSubmission:
The function has multiple responsibilities, such as handling form submission, processing form data, filtering books and also updates the book list. So it violates the SRP.
The function is not designed to be open for extension but closed for modification, meaning that if a developer wants to change the criteria for filtering books, they would need to modify the function. so it violates the OCP.

c) The updateBookList function:
The function violates SRP by handling multiple responsibilities, it has a responsibility to clear and update a book list, modifies the message display, handles pagination, resets the search form, and also updates the "show more" button.
The function also violates DIP, as it relies on a number of global variables and two other functions. This basically makes the function less flexible and harder to test.

---

3. How can The three worst abstractions be improved via SOLID principles.

a) The activeHandle function:
Since it violates SRP, it would be ideal to split the function into smaller functions, each with a single responsibility.
Since it relies on dependencies, this might be avoided by passing the dependencies as parameters instead of relying on global variables, this way the function would be more modular (independent) and easier to test.

b) The handleSearchFormSubmission:
Same as with the activeHandle function, the function should be split into smaller functions, each with a single responsibility.
For the function to adhere to OCP, it would be ideal to add a more flexible and extensible filtering mechanism (or criteria).

c) The updateBookListFunction:
Split it into smaller functions, each with a single responsibility, to adhere better to the SRP.
Also, explicitly passing dependencies, such as html, page, matches, BOOKS_PER_PAGE, form, overlay, updateShowMoreButton, and createBookPreview, as parameters to make the function more flexible, easier to test, and to adhere to the Dependency Inversion Principle (DIP).
