Assignment 2 - Part 1: Initial Implementation

Overview

In Part 1 of this assignment, I have created three classes: “Person”, “BlogPost”, and “Blog”. These classes are designed according to the specified requirements, with proper input validation, a builder pattern, and deserialization support. Additionally, I have written unit tests to validate the constructors and the proper functionality of the “Person” and “BlogPost” classes.

---

Class Implementation

“Person” Class

- Variables:

- “String id”

- “String firstName”

- “String lastName”

- “Integer age”

- “String gender”

- Key Features:

- Input validation is included to ensure:

- “id” is not “null”.

- “firstName” and “lastName” are not “null” or blank.

- “age” is not negative.

- The class uses a private constructor and a builder pattern via “@Builder”.

- The class is Jacksonized for deserialization using the “@Jacksonized” annotation.

“BlogPost” Class

- Variables:

- “String id”

- “String authorId” (which equals the “personId” of the “Person” class)

- “String postContent”

- Key Features:

- Input validation ensures:

- “id” and “authorId” are not “null”.

- The class also uses a private constructor and a builder pattern via “@Builder”.

- The class is Jacksonized for deserialization using the “@Jacksonized” annotation.

“Blog” Class

- Variables:

- “List<BlogPost> posts”

- “List<Person> contributors”

- Key Features:

- The class contains lists of “BlogPost” and “Person” objects.

- Implemented “toString()” and “equals()” methods for proper object comparison and string representation.

---

Unit Tests

“PersonTest”

- Tests implemented:

- Validates constructor exceptions when invalid input is provided.

- Validates successful object creation when valid input is provided.

-------------------------------------------------------

T E S T S

-------------------------------------------------------

Running com.assignment.BlogPostTest

Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.136 s -- in com.assignment.BlogPostTest

Results:

Tests run: 2, Failures: 0, Errors: 0, Skipped: 0

BUILD SUCCESS

“BlogPostTest”

- Tests implemented:

- Validates constructor exceptions for invalid input (null “id” or “authorId”).

- Validates successful object creation when valid input is provided.

-------------------------------------------------------

T E S T S

-------------------------------------------------------

Running com.assignment.PersonTest

Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.143 s -- in com.assignment.PersonTest

Results:

Tests run: 3, Failures: 0, Errors: 0, Skipped: 0

BUILD SUCCESS

Both tests ran successfully with no failures, errors, or skipped tests.

---

**Output**

Person 1 Details:

ID: P001

Name: John Doe

Age: 30

Gender: Male

Person 2 Details:

ID: P002

Name: Jane Smith

Age: 25

Gender: Female

Blog Posts:

Post ID: B001

Author ID: P001

Content: This is the first blog post content.

Post ID: B002

Author ID: P002

Content: This is the second blog post content.

Contributors:

John

Jane

Thank you 😊