ALGONQUIN COLLEGE

Assignment #2

In this assignment, you are required to implement different design patterns for the development of a Library Management System. The application should connect with a database named **bookvault** and support fundamental CRUD (Create, Read, Update, Delete) operations on **events** table within the database. Although the database consists of only one table, it is important to design your solution with scalability and extensibility in mind, anticipating the potential need for system modifications to accommodate additional functionalities in the future.

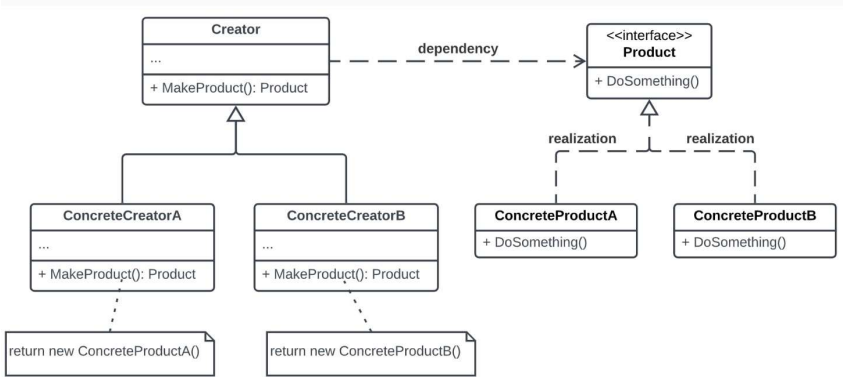
Part 1 - Design Your Solution

1. Develop a solution incorporating the Factory Method design pattern to create various types of Events based on the Library type. The system should accommodate two distinct Library types:
    a. Public Library
    b. Academic Library

2. Each library should provide two distinct Event types, each associated with a specific cost:
    a. Academic Library – Workshop and Book Launch
    b. Public Library - Movie Night and Kids Story Time

3. Implement **calculateAdmissionFee()** by using the **rate** and **duration** defined into **Constants**

4. Reference - Implement your solution based on the following UML diagram.



ALGONQUIN COLLEGE

Assignment #2

5. Implement the class named **DBConnection** utilizing the Singleton Design Pattern. This class should manage the creation of a single instance of the database connection.

6. Implement the class named **DBOperations** that is used for creating, retrieving, updating, and deleting a specific Event object generated by the Client class (You may create a separate class to handle the interaction between Client and Factory.).

7. Finally implement the class called **LMSLogger** responsible for managing application logging. Utilize the singleton design pattern to ensure the initialization of only one instance of the logger. This class should utilize various log levels (such as INFO, ERROR, WARNING, etc.) defined into the **LogLevel** class and used across the entire solution. Make sure to log the exception thrown by your program as well.

Part 2 - JUnit

1. Write test cases for **DBConnection** and **LMSLogger** class to test that there is only one instance of each object is created throughout the execution of your program.

2. Write test case for the each **calculateAdmissionFee()** method of the concrete implementation of each Event class.

Deliverables

- This task constitutes 7.5% of your overall grade and due by **June 14 (Friday), 11:59 PM.**
- Deliver a comprehensive coding solution.
- You code should include commenting and coding practice according to Java standards.
- You must **demo** your solution during the lab session and submit your code on **Brightspace**.
- Violating academic integrity or missing the deadline will result in a **grade of 0** for your submission.

ALGONQUIN COLLEGE

Assignment #2

Individual Assignment

You may verbally discuss the general approach to solving this individual assignment with other students and that is the only extent of collaboration allowed for this assignment. You are not allowed to work together and you are not allowed to share or read each other's code/deliverables. If your code or any other deliverables for this assignment resemble with those of another student, you will be reported to the Academic Integrity Office for cheating/plagiarism investigation. Please refer to the Academic Integrity policy document (AA48) of the college at https://www.algonquincollege.com/policies/files/2021/09/AA48.pdf .

Statement on Generative Artificial Intelligence (AI)

You may consult any tools and information available external to the course but you must quote the reference in your submission. Failing to do so will result in Academic Integrity investigation. Furthermore, any content taken directly from these tools/information base and submitted will result in proportional reduction in grade. Any code obtained from generative AI tools such as ChatGPT cannot be submitted as your work and will be considered as plagiarized.