The University of the West Indies, St. Augustine
COMP 2603 Object Oriented Programming 1
Assignment 1 (5%)
2023/2024 Semester 2

Due Date: February 07, 2024 at 10:00 p.m.
Lockout Date: February 09, 2024 at 10:00 p.m.

Overview:
An object-oriented application is required for simulating a chatbot platform that allows a user to create and interact with different types of chatbots through message prompts up to a daily limit.

UML Diagram of Domain Classes

Figure 1 shows a simplified UML diagram of the domain classes along with a main class called ChatBotSimulation which manages a ChatBotPlatform object. A ChatBotPlatform object store a collection of ChatBot objects which are generated with data from a ChatBotGenerator class based on a client supplied code.

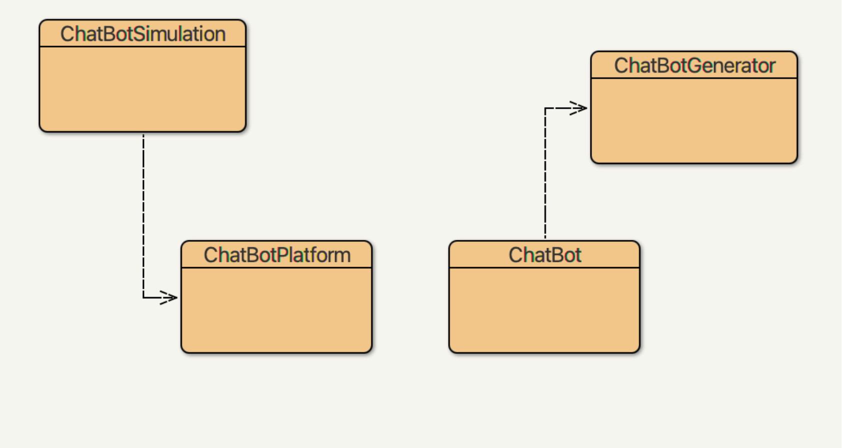

Figure 1: UML Diagram of the Domain Classes

Important Notes:

- Accessors must be provided for all attributes in all of the domain classes, mutators where appropriate/specified.
- Information hiding principles must be adhered to (use of access modifiers as appropriate for classes, attributes and methods).
- All methods are publicly accessible unless indicated.
- There are additional methods specified for the classes in the pages that follow. These methods are necessary for auto-testing of your program. Therefore your solution may or may not use these methods directly, however, the methods must be present in your code.
- The arrows on the diagram may not match your own version exactly as the visualisation may vary based on the BlueJ version.

ChatBot Class

The **ChatBot** class models chatbot that accepts a user message and generates a response up to a maximum daily limit. It has the following attributes and methods. Accessors are required for all attributes.

Attribute	Type	Purpose
chatBotName	String	Stores the name of LLM that powers the chatbot
numResponsesGenerated	int	Keeps track of the number of responses generated by the chatbot.
messageLimit	int	Stores the **fixed** total number of messages that can be generated overall by **all** chatbots. Initialised to 10.
messageNumber	int	Keeps track of the number of messages generated overall by **all** chatbots. Initialised to 0.

Method Signature	Return Type	Purpose
ChatBot( )		Constructor. Creates a ChatBot object with the default LLM set to ChatGPT-3.5 represented by the chatBotName, using the ChatBotGenerator class method.
ChatBot(int LLMCode)		Overloaded Constructor. Creates a ChatBot object with a specific LLM, represented by the chatBotName, using the ChatBotGenerator class method.
getChatBotName( )	String	Accessor for chatBotName
getNumResponsesGenerated( )	int	Accessor for numResponsesGenerated
getTotalNumResponsesGenerated()	int	Accessor for messageNumber. This is a **class** method.
getTotalNumMessagesRemaining()	int	Calculates the difference between the messages generated overall and the limit allowed. This is a **class** method.
limitReached( )	boolean	Checks whether the total limit of messages has been reached across all chatbots and return true, otherwise false. This is a **class** method.
generateResponse( )	String	Generates and returns a String containing a unique message number and the response from the chatbot with its name. The method also increments the count of the messages generated by the chatbot and the count of the messages generated overall by all chatbots. This method is **private.** *See below for sample output.

Method Signature	Return Type	Purpose
prompt(String requestMessage)	String	Processes the incoming message if the limit on the total number of messages has not be reached and returns the response generated by the ChatBot. Otherwise, it returns the following message: “Daily Limit Reached. Wait 24 hours to resume chatbot usage”
toString( )	String	Returns a String representation of a ChatBot object. *See below for sample output.

- Format for the generateResponse( ) output for a ChatBot powered by Mistral7B:

```
(Message# 4) Response from Mistral7B        >> generatedTextHere
```

- Format for the toString( ) output for a ChatBot powered by Mistral7B:

```
ChatBot Name: Mistral7B       Number Messages Used: 1
```

ChatBotPlatform Class

The **ChatBotPlatform** class manages one or more ChatBot objects within a bots collection. Accessors are required for all attributes.

Attribute	Type	Purpose
bots	ArrayList<ChatBot>	Stores multiple ChatBot objects.

Method Signature	Return Type	Purpose
ChatBotPlatform ( )		Constructor. Initialises the bots collection.
addChatBot(int LLMcode)	boolean	Creates and adds a new ChatBot object to the bots collection based on the LLM code supplied. The method returns true if successful, false otherwise. It first checks whether the limit has been reached for the number of messages that can be sent by the platform. If not, then a new ChatBot object is created and added to the bots collection using the ArrayList method.
getChatBotList()	String	Returns a String containing formatted information about all of the chatbots managed by the ChatBotPlatform object together with summary usage statistics. *See below for details.

Method Signature	Return Type	Purpose
interactWithBot (int botNumber, String message)	String	Passes a message to a given chatbot and returns its response. The botNumber represents the index of a chatbot in the bots collection. If an incorrect botNumber is supplied, the method returns an informative message with indicating the wrong number. Example: “Incorrect Bot Number (7) Selected. Try again”

*Sample output for the getChatBotList( ) method

```
----------------------
 Your ChatBots
Bot Number: 0 ChatBot Name: ChatGPT-3.5  Number Messages Used: 4
Bot Number: 1 ChatBot Name: LLaMa        Number Messages Used: 0
Bot Number: 2 ChatBot Name: Mistral7B    Number Messages Used: 1
Bot Number: 3 ChatBot Name: Bard         Number Messages Used: 0
Bot Number: 4 ChatBot Name: Claude       Number Messages Used: 1
Bot Number: 5 ChatBot Name: Solar        Number Messages Used: 3
Bot Number: 6 ChatBot Name: ChatGPT-3.5  Number Messages Used: 1
Total Messages Used: 10
Total Messages Remaining: 0
----------------------
```

ChatBotGenerator Class

The **ChatBotGenerator** class models the generation of various LLMs. No actual LLMs are used in this simulation.

Method Signature	Return Type	Purpose
generateChatBotLLM (int LLMCodeNumber)	String	A class method that returns the name of an LLM based on a supplied integer code. 1: “LLaMa” 2: “Mistral7B” 3: “Bard” 4: “Claude” 5: “Solar” Default: “ChatGPT-3.5”

ChatBotSimulation

This class contains the main method for your solution. It should do the following programmatically:

1. Begin with a Hello World! printed to the screen.
2. Declare and initialise a ChatBotPlatform object.
3. Add several ChatBot objects (at least one of each kind) to the ChatBotPlatform.
4. Print out on the screen the list of all ChatBots managed by the ChatBotPlatform with their summary statistics.
5. Interact up to 15 times with random ChatBots by sending messages, printing out their responses

to the screen

6. Print out on the screen a final list of all ChatBots managed by the ChatBotPlatform with their summary statistics.

See sample on the next page.

```
Hello World!
----------------------
 Your ChatBots
Bot Number: 0 ChatBot Name: ChatGPT-3.5         Number Messages Used: 0
Bot Number: 1 ChatBot Name: LLaMa       Number Messages Used: 0
Bot Number: 2 ChatBot Name: Mistral7B   Number Messages Used: 0
Bot Number: 3 ChatBot Name: Bard        Number Messages Used: 0
Bot Number: 4 ChatBot Name: Claude      Number Messages Used: 0
Bot Number: 5 ChatBot Name: Solar       Number Messages Used: 0
Bot Number: 6 ChatBot Name: ChatGPT-3.5         Number Messages Used: 0
Total Messages Used: 0
Total Messages Remaining: 10
----------------------
(Message# 1) Response from Solar        >> generatedTextHere
(Message# 2) Response from ChatGPT-3.5          >> generatedTextHere
(Message# 3) Response from ChatGPT-3.5          >> generatedTextHere
(Message# 4) Response from Mistral7B    >> generatedTextHere
(Message# 5) Response from ChatGPT-3.5          >> generatedTextHere
(Message# 6) Response from ChatGPT-3.5          >> generatedTextHere
(Message# 7) Response from ChatGPT-3.5          >> generatedTextHere
(Message# 8) Response from Claude       >> generatedTextHere
(Message# 9) Response from Solar        >> generatedTextHere
(Message# 10) Response from Solar       >> generatedTextHere
Incorrect Bot Number (7) Selected. Try again
Daily Limit Reached. Wait 24 hours to resume chatbot usage
Daily Limit Reached. Wait 24 hours to resume chatbot usage
Daily Limit Reached. Wait 24 hours to resume chatbot usage
Daily Limit Reached. Wait 24 hours to resume chatbot usage
----------------------
 Your ChatBots
Bot Number: 0 ChatBot Name: ChatGPT-3.5         Number Messages Used: 4
Bot Number: 1 ChatBot Name: LLaMa       Number Messages Used: 0
Bot Number: 2 ChatBot Name: Mistral7B   Number Messages Used: 1
Bot Number: 3 ChatBot Name: Bard        Number Messages Used: 0
Bot Number: 4 ChatBot Name: Claude      Number Messages Used: 1
Bot Number: 5 ChatBot Name: Solar       Number Messages Used: 3
Bot Number: 6 ChatBot Name: ChatGPT-3.5         Number Messages Used: 1
Total Messages Used: 10
Total Messages Remaining: 0
----------------------
```

Submission Instructions

- Write the Java code for each class in the application using BlueJ.
- Document your student ID at the top of each file within a comment block.
- Upload a ZIP file of your compiled project source and class files to the myElearning course page by the deadline. **Submissions that do not compile or which are empty will receive 0 marks.**
- Name your ZIP file as follows: **FirstName_LastName_ID_A1.zip. Marks will be deducted for submissions that do not conform to this naming convention.**
- Accept the University Plagiarism declaration statement (tick box) upon submission on myeLearning confirming that you are submitting your own original work and that you have not copied or collaborated with other students.
- Early submission points (5 marks) can be earned if you submit before the due date. No penalties will be applied but no early points will be earned if you submit by the lockout date.

Important Information Regarding Academic Conduct

- This is an individual assignment. You must attempt this assignment by yourself without any help from others (Humans and Generative AI Tools).
- You may use legitimate resources on the Internet, in books, or from the course notes to assist (unless prohibited by a question). Copying or modifying such content does not make it yours. **Indicate on your submission any or all sources used in your answers within comments at the end of your files. Identify code snippets that you did not write yourself but reused from some source.**
- You are not allowed to communicate, share or disclose any aspect of your solutions/work in any form to anyone except for the course lecturer, markers, examiners and tutors.
- You are not allowed to assist others with this assignment.
- University plagiarism and academic misconduct policies apply fully.
- No part of your submission should be made publicly available even after the due date without permission from the course lecturer.
- The use of Chegg.com, CourseHero.com or any tutoring or homework assistance services, online or not, are prohibited.
- The use of Generative AI Tools to produce code solutions is prohibited.
- If you are faced with extenuating circumstances, contact your lecturer right away. Concessions may be arranged on a case-by-case basis. Be honest with your claims and provide evidence where possible.