/\*\*

\* The place where we create ChatRoom and User and get the users to broadcast messages

\*/

class ChatApplication {

public static void main(String[] args) {

ChatRoom cr = new ChatRoom(); // an object of type Chatroom called "cr" is created. How? (inside ChatRoom.java)

User u1 = new User(cr); // an object of type User called "u1" created. How? (inside User.java)

User u2 = new User(cr); // another object of type User called "u2" is created.

int converationLength = 10; // What happens next?

for (int i = 0; i < 10; i++) {

u1.writeMessage(); // ?

u2.writeMessage(); // ?

}

cr.printLog(); // chat room starts the log printing behavior

}

}

class ChatRoom {

private static int MAX\_CHAT\_LOGS = 1000; // The size of the log list is decided to be 1000.

private String []chatLogs = new String[MAX\_CHAT\_LOGS]; //a 1000-sized array of String objects "chatLogs" is created.

private int totalChats;

// Our ChatRoom has no defining attributes for now

public void ChatRoom() {

}

// Store the user chat

public void receiveUserChat(String message) {

// Fill out the method - explain your code (how do you make a ChatRoom object store the message?)

}

// Retreive the chat logs

public void printLog() {

// Fill out the method - explain your code (how do you make this object retrieve the log?)

}

}

import java.util.Random;

class User {

static int autoId = 0; // Class variable for assigning ID

private int userId; // User Identification

public static final int MAX\_MESSAGE\_LENGTH = 100; // Maximum message length

private ChatRoom assignedChatroom; // ChatRoom object reference associated with the User object

public User(ChatRoom cr) {

autoId++;

userId = autoId;

assignedChatroom = cr;

}

// Broadcasts a random message to the ChatRoom

public void writeMessage() {

String alphabet = "abcdefghijklmnopqrstuvwxyz";// a String object "abcdefghijklmnopqrstuvwxyz" is created.

String message = "["+userId+"] "; // What happens here?

Random messageRandomization = new Random(); // What happens here?

int messageLength = 1+messageRandomization.nextInt(MAX\_MESSAGE\_LENGTH); // the Random object "messageRandomization" finds the next integer in the range [0, 100).

for (int i = 0; i < messageLength; i++) { // What is the range of values possible to be assigned for messageLength?

message += alphabet.charAt(messageRandomization.nextInt(alphabet.length())); // What happens here?

}

// Fill out - you need to send the message to the ChatRoom object

}

}

/\*\*

\* The place where we create ChatRoom and User and get the users to broadcast messages

\*/

public class ChatApplication {

public static void main(String[] args) {

ChatRoom cr = new ChatRoom();

User u1 = new User(cr);

User u2 = new User(cr);

int converationLength = 10;

for (int i = 0; i < 10; i++) {

u1.writeMessage();

u2.writeMessage();

}

cr.printLog();

}

}

public class ChatRoom {

public static final int MAX\_CHAT\_LOGS = 1000;

private String []chatLogs = new String[MAX\_CHAT\_LOGS];

private int totalChats;

// Our ChatRoom has no defining properties for now

public void ChatRoom() {

}

// Store the user chat

public void receiveUserChat(String message) {

chatLogs[totalChats] = message;

totalChats++;

}

// Retreive the chat logs

public void printLog() {

for (int j = 0; j < totalChats; j++) {

System.out.println(chatLogs[j]);

}

}

}

import java.util.Random;

public class User {

static int autoId = 0; // Class variable for assigning ID

private int userId; // User Identification

public static final int MAX\_MESSAGE\_LENGTH = 100; // Maximum message length

private ChatRoom assignedChatroom; // ChatRoom object reference associated with the User object

public User(ChatRoom cr) {

autoId++;

userId = autoId;

assignedChatroom = cr;

}

// Broadcasts a random message to the ChatRoom

public void writeMessage() {

String alphabet = "abcdefghijklmnopqrstuvwxyz";

String message = "["+userId+"] ";

Random messageRandomization = new Random();

int messageLength = 1+messageRandomization.nextInt(MAX\_MESSAGE\_LENGTH);

for (int i = 0; i < messageLength; i++) {

message += alphabet.charAt(messageRandomization.nextInt(alphabet.length()));

}

assignedChatroom.receiveUserChat(message);

}

}