

CSE3OAD/CSE4OAD – Lab 7

Socket Programming

Provided Files:

Task 1

- SocketClient.java
- SocketServer.java
- SocketThread.java

Task 2

- EBACThread.java

Task 3

- HangMan.java
- HangManQuiz.java
- Helper.java
- HangManWords.txt

Task 1: Experimenting with the Socket Client/Server programs

1. Modify the port numbers of SocketClient.java and SocketServer.java. Run the programs and observe how they work. What happens if the SocketClient is run first? What happens if multiple SocketClient try to connect?
2. Modify SocketServer to perform some other operation with the number it receives.
3. Ask your friends for the address/port of their SocketServer and connect your Socket-Client to it.

Task 2: Blood Alcohol Estimator

Using the provided EBACThread.java class, you will write a simple client/server program to calculate the Estimated Blood Alcohol Concentration (EBAC) of the user.

The method to calculate this, calculateEBAC, is provided for you. You don't need to understand how it works and can skip to part 1 of this task, however the formula is described below if you're interested.

We can get an approximation of the amount of alcohol in a person's blood using the following formula:

$$EAB \left(in \frac{g}{dL} \right) = 0.9672 \times \frac{SD}{BW \times Wt} - MR \times DP$$

Where:

- SD is the number of Standard Drinks consumed
- BW is a body water constant (0.58 for men and 0.49 for women)
- Wt is the weight of the person
- MR is the metabolic rate (average of 0.015 for men and 0.017 for women)
- DP is the drinking period (in hours)

This information is taken from Wikipedia and cannot be guaranteed to be accurate.

1. Write a new program called EBACClient and open a new socket (ask your demonstrator what Address and Port to use, or do this question after completing parts 2 and 3 of this task). Ask the user for the following information and send it to the server:

- The number of standard drinks that the user has consumed (as an int).
- The user's weight (as a double).
- The user's gender (male or female); send this to the server as a boolean (using true for male and false for female).
- The number of hours since the user started drinking (as a double).

The server will return a double representing the user's EBAC and a String with a message (separated by a newline). Display this information to the user before finishing.

For example, given the input:

```
Enter the number of standard drinks consumed: 3
Enter your weight (in kg): 80
What is your gender (male/female)? male
How long ago did you start drinking (in hours)? 2.0
```

The server will return:

```
0.033
Be careful if you need to drive!
```

2. Implement the run method of the EBACThread class. This method should receive the following from the input stream:

- An int representing the number of standard drinks consumed.
- A double representing the weight of the user.
- A boolean indicating the users gender (true for male, false for female).
- A double representing the number of hours since the user started drinking.

These parameters should be passed to the calculateEBAC method, which will return a double representing the EBAC of the sender. For men, the parameter body_water_constant should be 0.58 and metabolism_rate should be 0.015. For women, the parameter body_water_constant should be 0.49 and metabolism_rate should be 0.017. Send the calculated EBAC to the client, followed by one of the following messages:

- If the sender's EBAC is over 0.05, send "If you drink and drive, you are ...".
- If the sender's EBAC is between 0.001 and 0.05, send "Be careful if you need to drive!"
- Otherwise, send "You haven't had any alcohol; it's safe for you to drive!"

3. Create a new class called EBACServer. This class should contain a main method, which creates a ServerSocket. When a connection is received, it should be handled by a new instance of the EBACThread class that you created in part 2.
4. Modify the EBACClient that you created in part 1 to connect to the EBACServer that you've just written. Test your programs to make sure that they work.

Task 3: A game of Hangman

Take a look at the program HangManQuiz and spend some time to understand how it works. Create a new Server which will allow a Client to play a single game of HangMan.

