
SOFTWARE D'APLICACIONS DISTRIBUÏDES

Bon dia Francesc,

Som el Esteve i el Hamza i t'adjuntem aquí un petit índex de totes les pràctiques i les diferents parts opcionals que has anat proposant. Cada part feta la trobaràs o bé al git del Hamza o bé al de l'Esteve. Per si a cas, et passarem un zip de totes les pràctiques també.

Al mateix temps t'afegim uns quants documents teòrics que hem anat recollint per si els futurs alumnes de l'assignatura per si alguns volen més necessiten informació.

URL GIT ESTEVE:

<https://github.com/Evm7>

URL GIT HAMZA:

<https://github.com/haerba/SAD>

PRÀCTICA 1

Al git del Hamza trobaràs:

(Exercicis 1, 2, 3, 4, 7 i 10)

Versió bàsica (no MVC) del EditableBufferedReader que implementa:

- ☞ Console communication implemented using **Runtime**
- ☞ Line modification uses a **simple String** for all the modifications
- ☞ Class **Line** has **Column Counter**
- ☞ The **VIEW** part is implemented on the `EditableBufferedReader.readLine()`

Versió (MVC) EditableBufferedReader:

- ☞ Using pattern MVC
- ☞ Line modification uses an **array of Chars** for all the modifications
- ☞ Instead of Observer, it uses the **class PropertyChangeSupport**
- ☞ Transform the **List of Chars** using **StringBuilder**

Versió (MVC + ratolí) EditableBufferedReader:

- ☞ We implement some movements of the functionality **of the mouse**.
- ☞ Transform the **List of Chars** using the **vector length**

Al git del Esteve trobaràs:

Exercicis 1, 2, 3, 5, 6 i 9

MVC EditableBufferedReader - editablebufferedReader_Line:

- ☞ Console communication implemented using **ProcessBuilder**
- ☞ Line modification using our **own** implemented class **MyStringBuffer**
- ☞ Class EditableBufferedReader has **Column Counter**
- ☞ The VIEW console implemented by using **Observer/Observable**

MVC EditableBufferedReader - editablebufferedReader_EDITOR:

- ☞ Same as the **last MVC** with its characteristics but it has the possibility of editing a **large line that has a bigger size** than the columns of the terminal.
- ☞ **Multiline Editor**.

PRÀCTICA 2

Al git del Hamza trobaràs:

Versió del xat implementat amb Java, usant un servidor MultiThreading amb capacitat de múltiples clients:

- ☞ We implemented two classes based on Socket and ServerSocket which only make try-catch statements **in order not to throw exceptions**.
- ☞ It is a socket **connection via TCP**
- ☞ One worker, **sends to rest of the workers one by one**.
- ☞ Colours to help interaction and visualize better who has transmit the message .

Al git del Esteve trobaràs:

ChatBash és un xat entre múltiples clients basat en JAVA i un servidor MultiThreading:

- ☞ We implemented two classes based on Socket and ServerSocket which only make try-catch statements **in order not to throw exceptions**.
- ☞ It is a socket **connection via TCP**
- ☞ Server is based on **Reactor patern and Selector NIO**.
- ☞ En the visualitzation enviroment Bash of the Chat we are able to visualize the use of **Scape Sequences** in order to ease the reading comprehension:
 - Erase the written line as we send the message. In console there are only messages definetly send.
 - Colours to help interaction and visualize better who has transmit the message (Admin use White).
 - Permits the personalitzation of the Port and IPAdress to which Server/Client are Connected to and use; by attaching this information as paràmetres in the inicialitzations. It also notificates errors in connection.

PRÀCTICA 3

Al git del Esteve trobaràs:

Versió del xat entre clients múltiples basada en Swing amb un servidor multi-threading:

- ▣ Server is based on Selector **NIO and Reactor pattern**.
- ▣ Client is **multithreading**: main thread **works as the collector information** from the user, whereas the **ClientThread communicates with the server and also process the information**.
- ▣ As we can see, **ChatClient uses Swing to ease the interaction of user**.
- ▣ We implemented **two classes based on Socket and ServerSocket** which only make try-catch statements in order **not to throw exceptions**.

Al git del Hamza trobaràs:

Versió bàsica usant MVC del xat entre clients múltiples basada en Swing amb un servidor multi-threading:

- ▣ Username demanded **via terminal**
- ▣ Users **does not know when the other users leave**
- ▣ Everyone sees the **same conversation**
- ▣ **No color** modifications
- ▣ Sending a text is **activated via enter o click** to the button send

Versió avançada 1 usant MVC del xat entre clients múltiples basada en Swing amb un servidor multi-threading:

- ▣ Username is asked for via an **auxiliar display**
- ▣ Users are **aware of who leaves and enters the chat**, via text and via user list.
- ▣ Everyone sees the **same conversation**
- ▣ Sending a text is **activated via enter o click** to the button send
- ▣ We modify the **background colors** of the text insertion zone and we differentiate between the actual user (**right alignment**) and the rest of the users (**left alignment**).

Versió avançada 2 usant MVC del xat entre clients múltiples basada en Swing amb un servidor multi-threading: (NOT FINISHED).

- ▣ Username is asked for via an **auxiliar display**
- ▣ Users are **aware of who leaves and enters the chat**, via text and via user list.
- ▣ Everyone sees the **same conversation**
- ▣ Sending a text is **activated via enter o click** to the button send
- ▣ We modify the **background colors** of the text insertion zone and we differentiate between the actual user (**right alignment**) and the rest of the users (**left alignment**).
- ▣ We modify the form of where the text is inserted and we **use chat bubbles**.

PRÀCTICA extra

Al git del Hamza trobaràs:

Versió del HangMan per a consola amb el ús del llenguatge PYTHON