

SD | Pràctica 1 client-servidor.

L'objectiu docent de la pràctica és aprendre a utilitzar els mecanismes de programació Client/Servidor en JAVA. Concretament és necessari que aprengueu com programar amb:

- Sockets amb JAVA (utilitzant l'API Socket de Java.net)
- Servidor multi-petició amb i sense threads (JAVA)

Set i mig

El **Joc del Set i mig** és un famós joc de cartes similar al blackjack que es juga en els casinos. En aquesta pràctica implementarem una versió client/servidor del joc. En aquesta versió un sol client jugarà contra el servidor, però el servidor podrà servir múltiples partides alhora.

Tasques a realitzar.

- Programar un client i un servidor que implementin el protocol per al set i mig segons la nostra [especificació](#).
 - El client ha de tenir un mode manual (menú per pantalla) i un mode automàtic (juga automàticament segons els [paràmetres introduïts](#))
 - S'han de fer dues implementacions del servidor, una ha de ser multi-thread i l'altre usant Selector.
 - El servidor ha d'escriure un log de l'interacció amb el client a fitxer seguint el [format especificat](#).
- Fer proves de robustesa i d'estrès del sistema.
- Realitzar memòria de la pràctica.

Calendari

Sessió	Tasca
12/02/2015	a) Creació de grups. b) Preparació del GitHub. c) Extedre la classe ComUtils.java .
19/02/2015	Protocol
26/02/2015	Implementació client.
05/03/2015	Implementació servidor multi-thread.
12/03/2015	Implementació servidor amb Selector.
19/03/2015	Sessió de Test creuat.
01/04/2015	Entrega codi, memòria i log execució a CV.

Especificació

[Aquí](#) podeu trobar el RFC per a la nostre aplicació client/servidor pel joc del 7 i mig. Llegiu-lo atentament i seguiu-lo escrupulosament.

Notes importants

- No es demana que s'implementi cap interfície gràfica.
- En cada sessió de pràctiques es comprovarà si s'ha arribat a l'objectiu fixat en el calendari per la sessió anterior.
- Feu un bon ús de la programació Orientada a Objectes, es penalitzarà si no se'n fa d'un bon ús d'interfícies i separació de responsabilitats.

Execució

```
servidor> java Server -h
us: java Server -p <port> -b <starting_bet> -f <deckfile>

client> java Client -h
us: java Client -s <maquina_servidora> -p <port> [-a topcard]
```

Starting_bet és un sencer que indica l'aposta mínima que el servidor imposa al jugador.

Deckfile és la ruta a un arxiu de definició de la baralla com [aquest](#).

Topcard és un paràmetre opcional que s'usa quan es juga amb el client automàtic i defineix fins a quin valor entre 1 i 7.5 seguirà demanant cartes el client.

Sortida demanada

En el fitxer [ServerThread-0.log](#) podeu trobar un exemple de log de sortida per [aquest](#) fitxer de cartes i la següent execució del client:

```
client> java Client -s localhost -p 1212 -a 6
```

Només cal guardar el fitxer de log en el Servidor. Heu de guardar només el contingut de la comunicació per socket, tant del que es rep com el que s'està enviant. En cas de que hi hagi un error també heu de guardar el missatge que s'envii o que es rebi pel socket.

En el cas del servidor multithread el nom del fitxer de log l'heu de construir de la següent forma:

```
"Server"+Thread.currentThread().getName()+".log"
```

En el cas del servidor monothread, digueu-li serverGame-0.log augmentant en u l'índex per cada partida nova que enguegui el servidor.

Amb aquest [notebook](#) o [script python](#) podeu comprovar si l'execució i la sortida demanada que feu és correcte. Sobretot feu una carpeta src per cada aplicació on les classes principals es diguin cLiEnti server respectivament. Comproveu que el format de sortida és **EXACTAMENT** el mateix que el que teniu en el fitxer [ServerThread-0.log](#) ja que s'usa la

comanda `diff`.

Entregues

- Actualitzar codi a Github abans de cada sessió.
- Sessió Presencial de Test obligatòria.
- A CampusVirtual: 1/04/2015 23.55h.

Avaluació

En cas de que el codi no compleixi les especificacions determinades o no segueixi el protocol especificat la pràctica estarà suspesa. El fitxer de log del servidor ha de seguir **estrictament** el [format indicat](#), ja que es compararan totes les entregues de forma automàtica amb diferents jocs de proves.

En cas de que es superi tot el conjunt de proves satisfactòriament es ponderarà de la següent forma:

- 35% Memòria.
- 65% Estil de codi.