

Examen PSP.

Información general:

JorgeDuenasLerin/docencia-23-24/blob/main/examenes/README.md

Tiempo de examen: 2:45 horas

Ejercicio 1

“Cogemos una par de B29 y os pacificamos en pequeños trocitos”.

La estanquera de Saigon. LCDM.

HTTP y Observer. Servido HTTP monothread.

Tienes una estructura de datos con ciudades, capitales en mayúsculas y una información de si han sido pacificadas o no.

Implementa un servicio para consultar si han sido pacificadas, el puerto de escucha se recibe por parámetro.

```
/consulta/Madrid/  
/consulta/Albacete/  
/consulta/Boston/
```

Cada vez que se haga una consulta el sistema responderá con HTTP indicando el estado de esa ciudad. Pueden suceder 3 cosas:

- 200 - Pacificada
- 200 - Salvaje
- 404 - No encontrado

Este servidor implementa un observer. Este observer notificará a sus observadores cuando se consulte una ciudad que es capital y que sí ha sido pacificada. Este observer escribirá por la consola “Ha sido consultada una capital pacificada: *Nombredecapital*”

Puntos:

- 0.5 puntos. carga de datos
- 0.5 puntos. inicio socket
- 1.0 puntos. respuesta HTTP 200
- 1.0 puntos. respuesta no encontrado
- 1.0 puntos. observer
- 1.0 puntos. escribir de capitales pacificadas

Ejercicio 2

“Los medios nos empujan a poner alarmas”.

El país del miedo. LCDM.

Observer y UDP Broadcast.

Estás integrando un sistema de alarmas para tu casa. Las alarmas emiten caracteres con los sucesos que pasan y detectan cada minuto.

Los códigos son los siguientes:

- N - Nada
- B - Banco
- O - Okupa

El programa recibe dos puertos al ejecutarse. Uno de recepción y otro de envío.

En el puerto de recepción te llegan reportes de sucesos cada 5 minutos: es decir tramas con 5 caracteres de 'N', 'B', 'O'.

El receptor implementa el patrón observer. Cada vez que llegue una B o una O notificará el suceso a los observadores.

Implementa un observador. El observador recibirá las notificaciones. Cuando le llegue una notificación de 'B' u 'O', enviará a Broadcast el texto "¡Alarma!". Cada vez que llegue una 'B' registrará el suceso con la hora en un fichero bancos.txt con la fecha

```
String timeStamp = new SimpleDateFormat("yyyy.MM.dd HH:mm:ss").format(new Date());
```

Puntos:

- 0.5 puntos: recibir parámetros
- 0.5 puntos: escuchar en socket
- 1.0 puntos: procesar trama
- 1.0 puntos: entregar mediante observer
- 1.0 puntos: envío broadcast
- 1.0 puntos: registrar banco

Ejercicio 3

“No lo recordaba, un montón de gili%#\$as sintiéndose ricos”.

Fiebre del Viernes Noche. LCDM.

Estás haciendo el sistema de entradas para los bares más exclusivos de Ponzano. Cuando inicies el programa leerá el fichero entradas.csv

El programa recibe por parámetro un puerto de escucha y espera recibir códigos de entrada.

El programa recibirá por UDP texto de entradas. Escribirá por pantalla:

- “VÁLIDA” si está en el fichero y no está consumida. valor a 0.
- “CANJEADA” si está pero ya ha sido canjeada.
- “NOENCONTRADA” si no está.

Cuando una entrada sea procesada como válida cambiará la información en la estructura de memoria. Solo se lee el fichero en el inicio del programa.

Este validador de entradas tendrá un observer. Publicará cada entrada válida.

Implementa un observador VIP, cuando la suma de los números de la entrada sea primo escribirá por pantalla “EntradaVIP - *entrada*” con el código de la entrada. Esta salida se añada a la salida descrita anteriormente.

Puntos:

- 0.5 puntos: cargar datos
- 0.5 puntos: gestionar puerto de escucha
- 1.0 puntos: mostrar información
- 1.0 puntos: actualizar información
- 1.0 puntos: observer
- 1.0 puntos: implementar VIP