Tarea Java #A: El Chat

* Nombre de Archivo Java: ***Chat.java***

* Ejemplo Nombre Archivo Comprimido:

GutierrezAvilesLuisJATSOAD0413B.zip

Descripción

Implementar una aplicación gráfica Chat.

Requerimientos Generales

1. Todas las clases tengan atributos privados.
2. Mostrar un área de texto
3. Mostrar una etiqueta “IP” seguida de un campo de texto a la derecha
4. Mostrar una etiqueta “mensaje” seguida de un campo de texto a la derecha
5. Mostrar un botón con la etiqueta “enviar”
6. Programar un Thread (como clase interna ó externa a la clase Chat)
7. El hilo anterior se quede bloqueado a la espera de mensajes (como en Receptor.java)
8. El hilo imprima en el área de texto la IP de donde provenga el mensaje
9. Para la IP, se imprima sin la "/" usando el método getHostAddress() en lugar de toString())
10. El hilo imprima en el área de texto el contenido del mensaje que llegue
11. Realizar los puntos 7 al 10 para todo mensaje que llegue
12. Se puede probar la parte receptora de su chat si escucha al mismo puerto que el ejemplo Receptor.java y se usa al Emisor.java para enviarle mensaje a su máquina.
    No se necesita conexión a red si se usa la dirección destino localhost "127.0.0.1" al correr al Emisor
13. Tras presionar el botón se obtenga contenido del "campo IP", del "campo mensaje"
14. Después del paso anterior se inicien las instrucciones para emisión de mensaje por la red (como en Emisor.java) usando los datos obtenidos del punto anterior. Quizás sea más fácil empezar por la parte de emisión de su chat y probarla con Receptor.java corriendo en otra consola, luego prueban a su chat como receptor usando al Emisor.java en otra consola

15. El chat se pueda enviar un mensaje a sí mismo si se usa la IP localhost

Requerimientos y Restricciones Complementarios

A. Todos los identificadores de variables y constantes tengan nombres claros y significativos de modo que sea fácil saber qué dato almacenarán
B. Todo identificador debe ser necesario para la operación del programa
C. Cumplir con las convenciones para nomenclatura de variables y constantes explicados en clase según convenciones Java
D. Todo valor asignado a un identificador debe ser necesario para la operación de alguna parte del programa
E. A excepción de las constantes 0 y 1, los cómputos utilicen únicamente variables y constantes simbólicas
F. A excepción de que un requerimiento lo solicite, no incluir la redacción de los requerimientos en el código fuente
G. Las líneas de código fuente sean máximo de 100 caracteres
H. Código fuente indicando como comentario en la parte superior de cada archivo: nombre del(la) alumn@, sección y no. de tarea
I. Cumplir con las convenciones sobre indentación de código fuente
J. Código fuente libre de instrucciones anuladas mediante comentarios
K. El código fuente libre de errores de compilación y advertencias
L. Código fuente libre de comentarios o impresiones de pantalla que describan el funcionamiento del código
M. Evitar comparar banderas contra constante numérica o simbólica
N. Toda instrucción ejecutada sea necesaria según los datos de entrada; por ejemplo, en un programa que validara una fecha en base a día, mes y año, si se trata del mes de Enero, sería inapropiado comprobar si se trata de un año bisiesto, pues esto solo sería necesario para caso del mes de Febrero.
O. Evitar el uso del operador relacional == de una misma variable contra 2 ó más tipos de datos enteros siempre que en su lugar se pueda utilizar la estructura de control “switch”
P. Programa en ejecución libre de desbordamiento de arreglos
Q. Todos los métodos tengan nombres claros y significativos de modo que sea fácil saber cuál es su función en el programa.

R. Ejercer la programación modular siempre que sea posible la reutilización de código
S. Evitar declarar atributos cuyos datos no formen parte de las propiedades que describan la clase en la que se declaren o bien no representen el estado del programa, y que por tanto sea posible una implementación utilizando variables locales y paso de parámetros.
T. Todas las clases (y sus atributos) cuenten con un nombre (identificador) claro y significativo a modo que a la lectura sea intuitivo saber para qué se utiliza.
U. Toda instrucción del programa en ejecución sea ejecutable sin necesidad de recompilar el programa
V. Cumplir con las convenciones Java (para clases, métodos, etc.).
W. Todo atributo siempre sea privado, a menos que exista una implementación que por eficiencia justifique un menor ocultamiento de información, pero que igualmente el diseño cumpla con la propiedad de encapsulamiento.
X. Si incluye comentarios, estos sean solo para Javadoc; en sus métodos, estos sean breves y referentes solo a los parámetros y resultados a obtener, y de las clases solo su propósito; los anteriores en términos de caja negra.
Y. El Código fuente sea libre de instrucciones para suprimir advertencias
Z. Evitar el uso del modificador static, a menos que sea indispensable el uso de atributos de clase y métodos de clase, justificando su uso con comentarios en código

Criterios de Evaluación

- Cumplir con las “Reglas de Operación y Evaluación del Taller de Sistemas Operativos Avanzados”.
- Fecha de asignación: 11 de Septiembre del 2013
- Fecha planeación de entrega: 18 de Septiembre del 2013
- Fecha límite de entrega (Sólo 55% del total): 09 de Octubre del 2013
- Observación: _Ninguna_____
- Calificación en base a cobertura de requerimientos y fecha de entrega en horas clase