Übung 10: Async IRC Bot

In dieser Übung implementieren Sie Teile eines asynchronen IRC Bot mittels RxScala. Ein IRC Bot ist ein Programm, das in einem IRC Channel autonom auf Nachrichten reagiert.

> IRC (Internet Relay Chat), bezeichnet ein rein textbasiertes Chat-System. Es ermöglicht Gesprächsrunden mit einer beliebigen Anzahl von Teilnehmern in sogenannten Channels (Gesprächskanälen). Um an einen Chat teilnehmen zu können wird ein Client benötigt.
>
> Es folgt eine Auswahl der bekanntesten Produkte:
> Online: http://webchat.freenode.net/?channels=conpr_4iab
> Mac: http://colloquy.info/
> Windows: http://www.mirc.com/
> Linux: http://xchat.org/

Auf dem AD finden Sie das 10_Scala_Conpr.zip Archiv. Importieren Sie es als Scala Projekt in Ihre IDE. Das Package `as.irc` beinhaltet nur ein einzelnes Source File. Darin ist das Singletonobjekt mit der `main` Methode und die Klasse `RxIrcBot` enthalten.

Bevor Sie aber die Applikation starten, sollten Sie den Spitznamen (`RxIrcBot.nick`) und den "echten" Namen (`RxIrcBot.real`) Ihres Bots anpassen. Diese Namen werden vom Bot verwendet, um sich zu identifizieren. Hier sehen Sie den relevanten Code:

```
class RxIrcBot(out: OutputStreamWriter, in: ConnectableObservable[String]) {
  import RxIrcBot._

  def init(): Unit = {
    sendCmd("NICK", nick)
    sendCmd("USER", s"$nick 0 * : $real")
    sendCmd("JOIN", chan)

    // Log all messages
    if (logMessages) in.subscribe(cmd => println("< " + cmd), exc => println(exc))

    // Hier kommt Ihre Funktionalität
    // a)
    // b)
    // c)

    // in.connect muss als letztes ausgeführt werden.
    // Informieren Sie sich über Hot Observables für Details.
    in.connect
  }
```

Aufgaben

a) Begrüssen Sie gleich nach dem Betreten des Raumes die anwesenden mit einer Willkommensnachricht. Verwenden Sie dazu die Methode `sendMsg`.

b) Ein IRC Server schickt regelmässig PING Messages, die Ihr Bot mit einem PONG beantworten muss. Sonst wird die Verbindung vom Server gekappt. Sie haben zwei vorbereitete Methoden, um diese Funktionalität zu implementieren: `isPing` und `sendPong`. Beide nehmen als Argument die ganze PING Nachricht, die vom Server kommt.

c) Implementieren Sie Ihre eigene Bot Funktionalität. Idealerweise implementieren Sie Ihr Bot so, dass er nicht auf jede Nachricht reagiert, sondern nur auf jene, die mit einem bestimmten Muster beginnen. Sonst gibt es ziemlich schnell viel Traffic in einem Channel wo sich mehrere Bots tummeln!
Sie können also eine einkommende Nachricht untersuchen, ob sie z.B. mit ":rot13 " startet und den Rest der Nachricht als Rot13 codiert zurück schicken.
Mit der Methode `isPrivMsg` finden Sie heraus, ob es sich bei einem IRC Command um eine Nachricht eines Benutzers handelt und nicht etwa um ein Systemkommando.
Mit der Methode `senderMsg` können Sie die tatsächliche Textnachricht extrahieren und `senderNick` kann verwendet werden, um den Spitznamen des Senders einer Nachricht zu extrahieren.

Ideen für IRC Services:
- :time soll die aktuelle Uhrzeit zurückgeben
- Ein Psychotherapeut a la Eliza (http://en.wikipedia.org/wiki/ELIZA)
- Zapfen Sie irgendwelche Web-APIs an um interessante Dienste anzubieten:
  - Wikipedia, News, Wetter, Taschenrechner

Da Sie wissen, wer eine Nachricht geschickt hat, können Sie auch individuell Zustand für einzelne Benutzer speichern. Lassen Sie sich was einfallen!