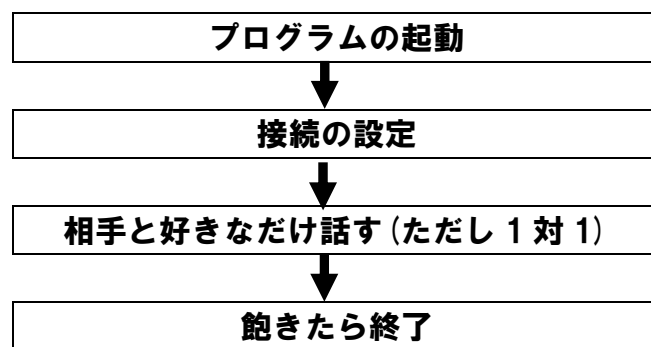


Java で作ろう！(簡単な)チャットプログラム

K06 庄子

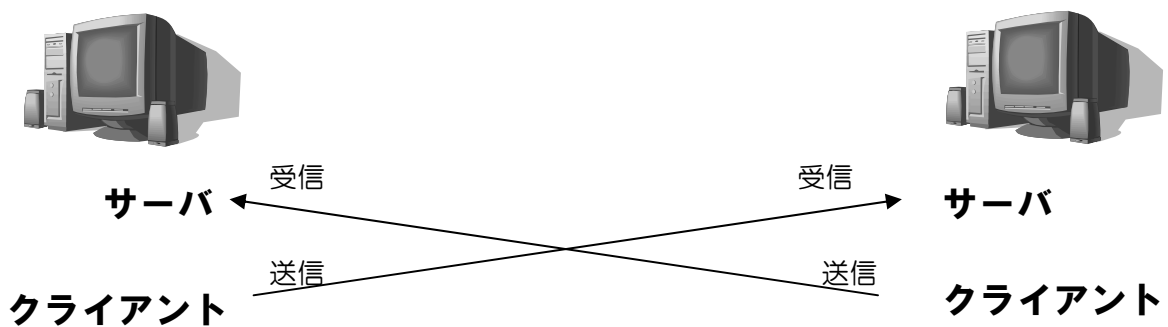
Java によるネットワークプログラミングと GUI の構築を一緒にやったら面白かろうと思い、勉強がてら簡単なチャットプログラムを作ってみました。

プログラムの流れ
大ざっぱに言えばこんな感じ。



処理の概要

1 つのアプリケーション内に、サーバとクライアントの両方を持たせることにより、双方向の送受信を実現しています。



各ソースファイルごとの説明

このプログラムは、以下に示す6つのソースファイルから成ります。

- ・ ProgMain.java

プログラムの開始部分です。自分が持つサーバのポート番号を入力させ、次のChat_Kernelを起動します。

- ・ Chat_Kernel.java

このプログラム全体の制御部分です。画面の作成、表示、HostSock, ServSock(後述)の作成などを担当します。また、メッセージの送信、受信したメッセージの表示もここを介して行います。

- ・ MainFrm.java

プログラムのメインとなる画面の設計を行います。

- ・ SettingFrame.java

接続設定を行う画面の設計を行います。

- ・ HostSock.java

クライアント機能を担当します。相手のアプリケーションが持つサーバ(ServSock)に対して接続を行い、メッセージの送信を行います。

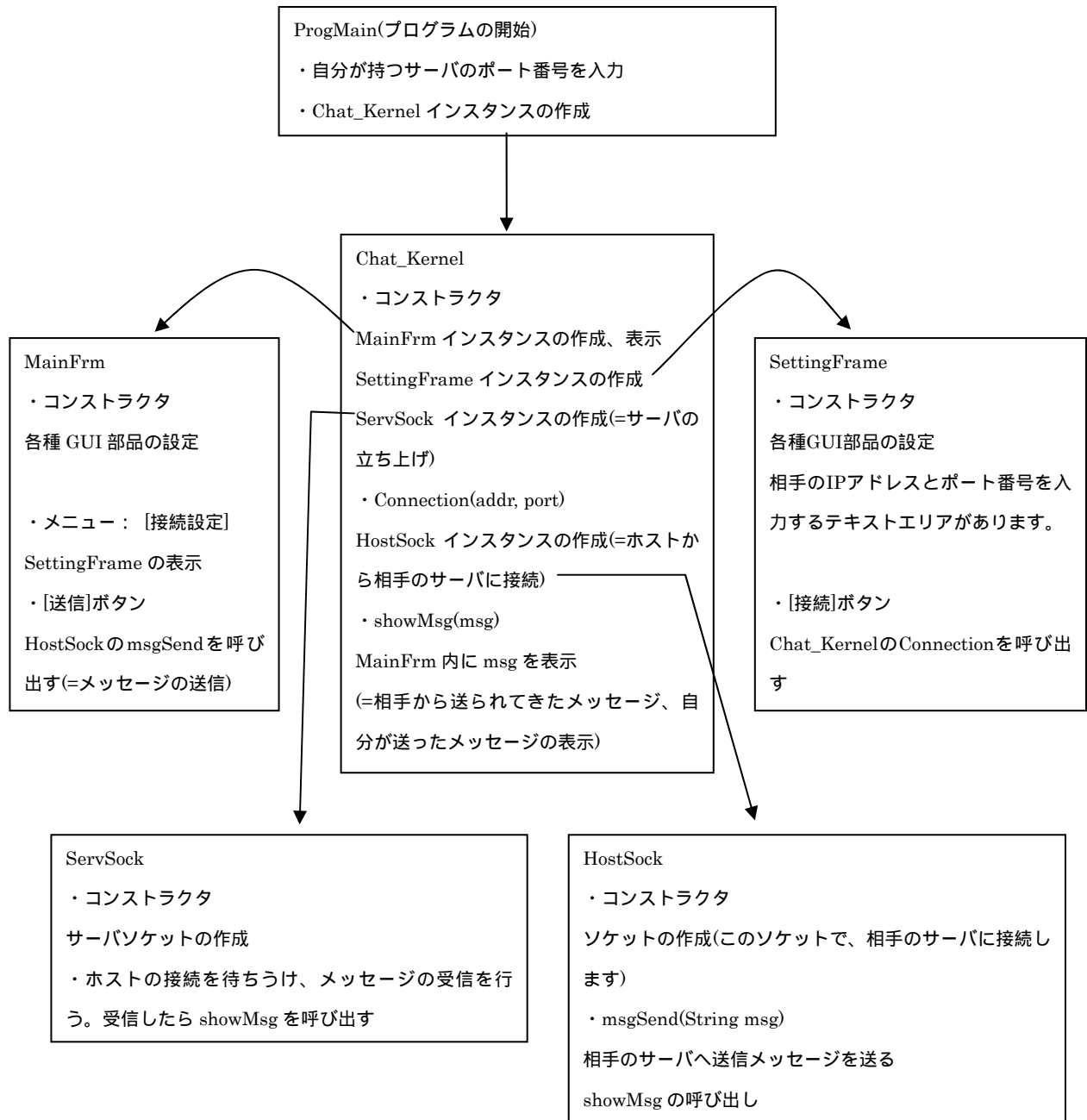
- ・ ServSock.java

サーバ機能を担当します。相手のクライアント(HostSock)からの接続を受けつけ、メッセージの受信を行います。

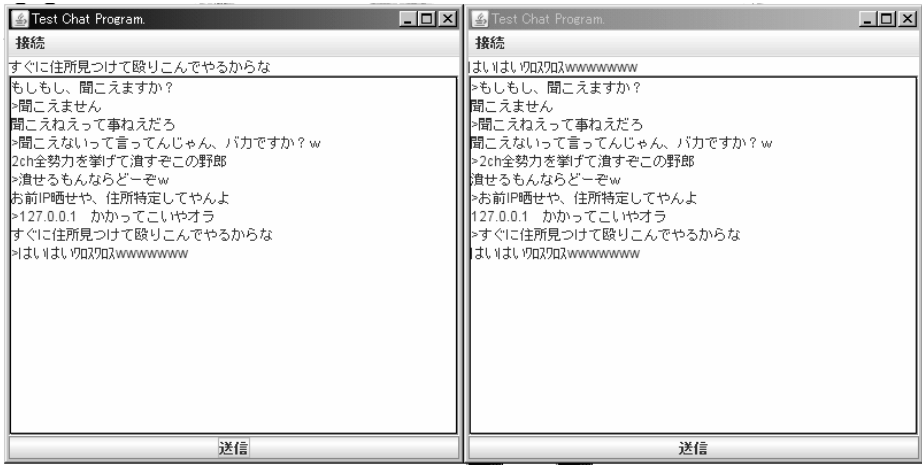
プログラムの状態遷移

物凄くゴチャゴチャしているので分かりづらいかもしれませんが....

矢印はインスタンスの生成を表しています。



実行画面(1人で2つ起動しています)



作成にあたって

Java のプログラムを作るときに、声を大にして言いたいこと。

「Java の GUI は面倒くさい！！！」

例えばVisualBasicなら、画面にボタンやチェックボックスをポンポン貼り付けて、コード書いて、それでアプリケーションが出来ます。でもJavaでGUIのアプリケーションを書こうとすると、それはそれは面倒くさい設定をチマチマ書かなければいけません。

詳細はソースコードを参照してもらうとして、このプログラムを作るのに費やした時間のうち、おそらく7割近くはGUIの設定につぎ込まれていると思います。

でもそれだけ沢山の時間を使ってまでGUIにしたかった理由というのも、結局は「GUIって楽しい」って所なんですね。目に見えて自分の作ったものが見えるし、いかにも「Windowsっぽい！」っていうものが作れるのは楽しいですよ。

こんな所から「プログラムって面白いな～」と感じて、のめりこんでいくのもアリかな...と思ったり。お粗末さまでした。