

```
/*
 * 作成日 : 2008/12/18
 *
 * TODO この生成されたファイルのテンプレートを変更するには次を参照。
 * ウィンドウ > 設定 > Java > コード・スタイル > コード・テンプレート
 */

/**
 * @author itou
 *
 * TODO この生成された型コメントのテンプレートを変更するには次を参照。
 * ウィンドウ > 設定 > Java > コード・スタイル > コード・テンプレート
 */
import java.io.*;
import java.net.*;
import java.util.regex.*;

class Act {
    String fruit = "";
    int num = 0;
    String action = "";
    boolean stock = false;

    void set(String str) {
        action = "";

        Pattern p = Pattern.compile(":");
        String s[] = p.split(str, 2);
        if (s[0].equals("fruit")) {
            fruit = s[1];
            System.out.println(fruit);
            if (fruit.equals("mikan") ||
                fruit.equals("apple") ||
                fruit.equals("grape") ||
                fruit.equals("kaki"))
```

```
        stock = true;
    else
        stock = false;
}
else if (s[0].equals("num"))
    num = Integer.parseInt(s[1]);
else
    action = s[1];
}
}
```

```
public class SocketTest {

    public static void main(String[] args) {
        try{
            // ソケットを生成
            Socket socket = new Socket("localhost", 10500);

            // 出力ストリームを取得
            PrintWriter out = new PrintWriter(socket.getOutputStream(), true);

            // 入力ストリームを取得
            BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));
            BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));

            out.println("PAUSE");

            while(true) {
                // julianに送るコマンドを入力
                String a = reader.readLine();
                System.out.println(a);
                out.println(a);

                // 読み込んだデータを表示
```

```
String b;
int state = 0;
    Act act = new Act();
// 音声認識のループ
while ((b = in.readLine()) != null) {
//
//
//
    System.out.println(b);
    // 正規表現による文のチェック
    switch (state) {
        case 0:
            boolean c1 = patterncheck(b, ".*RECOGOUT.*");
            if (!c1)
                continue;
            state = 1;
            continue;
        case 1:
            boolean c2 = patterncheck(b, ".*RECOGOUT.*");
            if (c2) {
                state = 2;
                continue;
            }
            boolean c3 = patterncheck(b, ".*<WHYPO WORD=. *");
            if (!c3)
                continue;
            String w = extword(b);
            if (w.equals("_"))
                continue;
            act.set(w);
            break;
        case 2:
            if (act.action.equals("confirm")) {
                if (act.stock)
                    System.out.println("ありますよ。いくつにしますか?");
                else
                    System.out.println("申し訳ありません。今、きれってます。");
            }
            state = 0;
            continue;
    }
}
```

```
        } else {
            System.out.println("1000円になります。");
            state = 0;
            continue;
        }
    }
}

}
} catch(IOException e) {
    e.printStackTrace();
}
}

public static String extword(String str) {
    Pattern p = Pattern.compile("¥");
    String s[] = p.split(str, 3);

    return s[1];
}

// 文法パターンのチェック
public static boolean patterncheck(String sent, String word) {
    Pattern pattern = Pattern.compile(word);
    Matcher matcher = pattern.matcher(sent);
    boolean c = matcher.matches();
    return c;
}
}
```