

Chapter 6

ネットワークプログラミング

以下の指示に従って、クラス、オブジェクトを作成・実行せよ。なお、文中では個別に指示しないが、変更を行ったら適宜コンパイルすること。

6.1 Eclipseを使おう

今回から統合開発環境としてよく使われるEclipseを使おう。

学情のコンピューターにはすでにEclipseがインストールされている。これを起動して次のことをやってみよう。

- Workspaceの場所の確認（基本はデフォルトでOk）
- Javaプロジェクトの作成
- Javaクラスの作成
- コードの実装

詳細は、例えば下記のWebサイトなどを参考にするとよい。

- <http://qiita.com/Kazunori-Kimura/items/aad2366531502a1da618>

6.2 クライアントプログラムの作成

- 次のTCP_Clientクラスを作成せよ。（次以降の手順ではあらかじめコンパイルされている必要がある）

```

import java.io.*;
import java.net.*;

class TCP_Client{

    public static void main(String[] args) {

        int port=10007;
        try {
            Socket skt=new Socket("127.0.0.1",port);

            (      A      ) oSW
            =new (      A      )(skt.getOutputStream());
            PrintWriter outS=new PrintWriter(oSW,true);

            (      B      ) iSR
            =new (      B      )(skt.getInputStream());
            BufferedReader inS= new BufferedReader(iSR);

            if(args.length==0){
                outS.println("");
            }else{
                outS.println(args[0]);
            }

            String tmpString="";
            while((tmpString=inS.readLine())!=null){
                System.out.println(tmpString);
            }
            inS.close();
            outS.close();

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

```

- 括弧Aと括弧Bにはクラス名が入る。適切なものを入れてプログラムを完成させよ。
- このクラスが行うのはどんな処理か。また、127.0.0.1とは何を表しているか。

6.3 サーバープログラムの作成

次のTCP_Serverクラスを作成せよ。(次以降の手順ではあらかじめコンパイルされている必要がある) ただし、XXXXXXXには何が入るか。適切な値を入れてプログラムを完成させよ。

```
import java.io.*;
import java.net.*;
import java.util.*;

class TCP_Server {

    public static void main(String[] args) {

        int port=XXXXXXX;
        try {
            (      C      ) srvSock=new (      C      )(port);
            boolean flg=true;
            while(flg==true){
                Socket skt=srvSock.(      D      )();

                InputStreamReader iSR
                    =new InputStreamReader(skt.getInputStream());
                BufferedReader inS= new BufferedReader(iSR);

                OutputStreamWriter oSW
                    =new OutputStreamWriter(skt.getOutputStream());
                PrintWriter outS=new PrintWriter(oSW,true);

                Calendar cal=Calendar.getInstance();

                String msg=inS.readLine();
```

```

        if(msg.equals("")){ flg=false; }

        String tmpString="Recieved:"+msg;
        tmpString+=" (at time "+cal.get(cal.HOUR);
        tmpString+=":"+cal.get(cal.MINUTE)+")";

        System.out.println(tmpString);
        outS.println(tmpString);

        inS.close();
        outS.close();
        skt.close();
    }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

```

- さらに、プログラムの処理について考察し、上記クライアントプログラム・サーバープログラムを動かし、クライアントプログラムから「こんにちは」と入力したときの結果を予想せよ。
- また、実際に動かしてみて動作を確認せよ。
- 各自のコンピュータのIPアドレスを調べて、クライアントプログラムから別のコンピュータ（隣の人と組んでやってもよい）で実行されていると上記サーバープログラムへアクセスしてみよ。

6.4 HTTP通信プログラムを作ってみる

<http://orion.data.ise.shibaura-it.ac.jp/> にアクセスし、ソースを取得するプログラムをSocketオブジェクトを使って（URLオブジェクトは使わないこと）作れ。