



TD09 Correction

Matière : DEVELOPPEMENT MOBILE AVANCE

Classes : SEM31

"JFDomotique"

```
private void lancerServeur() {  
    Runnable r = new Runnable() {  
        @Override  
        public void run() {  
            demmarrerServeur();  
        }  
    };  
    Thread th = new Thread(r);  
    th.start();  
}  
  
private void demmarrerServeur() {  
    try {  
        ss = new ServerSocket(PORT);  
        s = ss.accept();  
        br = new BufferedReader(new InputStreamReader(s.getInputStream()));  
        while (true) {  
            String cmd;  
            cmd = br.readLine();  
            execterCommande(cmd);  
        }  
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
}  
  
private void execterCommande(String commande) {  
    String[] t = commande.split(":");  
    if (t.length > 1) {  
        int type = Integer.parseInt(t[0]);  
        int valeur = Integer.parseInt(t[1]);  
        switch (type) {  
            case 0:  
                if (t[1] == 0)  
                    eteindreLampe();  
                else  
                    allumerLampe();  
                break;  
            case 1:  
                if (t[1] == 0)  
                    eteindreClimatiseur();  
                else if (t[1] == 1)  
                    allumerClimatiseur();  
                else if (t.length > 2)  
                    reglerTemperature(t[2]);  
                break;  
        }  
    }  
}
```

```

"MainActivity"
private void lancerThreadClient() {
    Runnable r = new Runnable() {
        @Override
        public void run() {
            demmarrerClient();
        }
    };
    Thread th = new Thread(r);
    th.start();
}
protected void demmarrerClient() {
    try {
        Inet4Address i = (Inet4Address) Inet4Address.getByName(edAdresse.getText().toString());
        s = new Socket(i, Integer.parseInt(edPort.getText().toString()));
        pw = new PrintWriter(new BufferedWriter(new OutputStreamWriter
                (s.getOutputStream())), true);
    } catch (UnknownHostException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
private void envoyer(final String cmd) {
    Runnable r = new Runnable() {
        @Override
        public void run() {
            if (pw != null) {
                pw.println(cmd);
            }
        }
    };
    Thread th = new Thread(r);
    th.start();
}
private void allumerLampe() {
    envoyer("0:1");
}
protected void eteindreLampe() {
    envoyer("0:0");
}
protected void allumerClim() {
    envoyer("1:1");
}
protected void eteindreClim() {
    envoyer("1:0");
}
protected void reglerTemperatureClim() {
    envoyer("1:2:" + seekTemp.getProgress());
}
}

```