

**TD09**

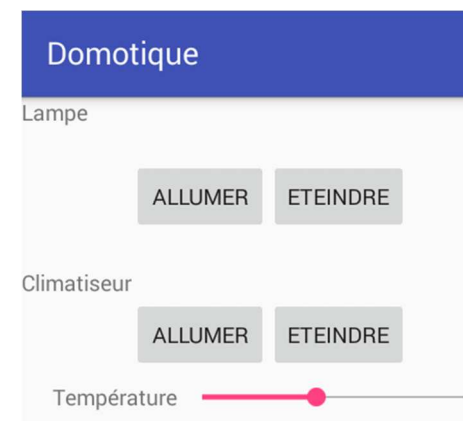
**Matière : DEVELOPPEMENT MOBILE AVANCE**

**Classes : SEM31**

## Exercice

On désire développer une application domotique qui permet de contrôler deux équipements d'un salon (une lampe (0) et un climatiseur (1)). Cette application utilise les sockets. Le programme serveur est installé sur un ordinateur d'adresse "192.86.10.11" connecté aux deux équipements pour les commander. Le programme client est installé sur un Smartphone android pour envoyer les commandes de l'utilisateur au serveur. Les opérations qu'on peut effectuer sur ces deux équipements et les messages envoyés par le socket clients sont résumées dans le tableau suivant :

Equipements	Opération	Paramètre	Message envoyé
Lampe (0)	eteindre	aucun	0:0
	allumer	aucun	0:1
Climatiseur (1)	eteindre	aucun	1:0
	allumer	aucun	1:1
	reglerTemperature	temperature	1:2:temperature



L'interface de l'application client android est la suivante :

- 1- Donner le code des méthodes "lancerThreadServeur()", "demarrerServeur()" et "executerCommande(...)" de la classe "JFDomotique" (coté serveur).
- 2- Donner le code des méthodes "lancerThreadClient()", "demarrerClient()", "allumerLampe()", "eteindreLampe()", "allumerClim()", "eteindreClim()" et "reglerTemperatureClim()" de la classe "MainActivity" (coté client).

```
public class JFDomotique extends JFrame {
    private static final long serialVersionUID = 1L;
    private static final int port = 6025;
    private ServerSocket ss;
    private Socket s;
    private BufferedReader br;
    public static void main(String[] args) {
        JFDomotique f = new JFDomotique();
        f.setVisible(true);
    }
    public JFDomotique() {
        setSize(300, 200);
    }
}
```

```

        setLocation(500, 200);
        setTitle("Serveur Domotique");
        lancerThreadServeur();
    }
    protected void eteindreLampe()    { // Eteint la lampe}
    protected void allumerLampe()    { // Allume la lampe}
    protected void eteindreClimatiseur() { // Eteint le climatiseur}
    protected void allumerClimatiseur () { // Allume le climatiseur}
    protected void reglerTemperature(int temp) { //Règle la température du climatiseur à temp}

```

```

    private void lancerThreadServeur() {
    }
    private void demarrerServeur() {
    }
    private void executerCommande(String commande) {
    }
}

```

```

package com.domotique;
//imports
public class MainActivity extends Activity {
    private Button btnAllumerLampe;
    private Button btnEteindreLampe;
    private Button btnAllumerClim;
    private Button btnEteindreClim;
    private SeekBar seekTemp;
    private static final String ADR_SERVEUR = "192.86.10.11";
    private static final int PORT = 6025;
    private Socket s;
    private PrintWriter pw;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        init();
    }
    private void init() {
        // Les findViewById
        ajouterEcouteur();
        lancerClient();
    }
    private void ajouterEcouteur() {
        // btnAllumerLampe.setOnClickListener(...) -> allumerLampe();
        // btnEteindreLampe.setOnClickListener(...) -> eteindreLampe();
        // btnAllumerClim.setOnClickListener(...) -> allumerClim();
        // btnEteindreClim.setOnClickListener(...) -> eteindreClim();
        // seekTemp.setOnSeekBarChangeListener(...) -> reglerTemperatureClim();
    }
}

```

```

    private void lancerThreadClient()    {    }
    private void demarrerClient()    {    }
    private void allumerLampe()    {    }
    protected void eteindreLampe()    {    }
    protected void allumerClim()    {    }
    protected void eteindreClim()    {    }
    protected void reglerTemperatureClim() {    }
}

```