

# Rapport du TP ERO

---

*Alarme*

**Réalisé par :**

**EL ANNAOUI IMAD**

**ELMIMOUNI MOHAMED**

**Encadré par :**

**Mme BOUCHIKHI**

## L'application "Alarme"

Est une application client/serveur CORBA qui permet d'afficher l'heure local et elle émet un son à une heure prédéterminée.































Organisation des fichiers de l'application Utilisant l'approche par délégation :

1. Alarm.idl : description IDL de l'interface.
2. AlarmDelegate.java : implémentation Java du servant.
3. DelegationServeur.java : application côté serveur.
4. GUIClient.java : application côté client

## Les étapes de l'exécution :

**1-Compiler le fichier .idl avec la commande : Idlj -fall Alarm.idl**

Après il se génère plusieurs fichiers .java :

 <b>_AlarmStub</b> Fichier JAVA 4.17 Ko	 <b>Alarm.class</b> Fichier CLASS 182 octet(s)	 <b>Alarm.idl</b> Interface Definition Language... 172 octet(s)	 <b>Alarm</b> Fichier JAVA 283 octet(s)
 <b>AlarmDelegate\$1.class</b> Fichier CLASS 1.36 Ko	 <b>AlarmDelegate\$2.class</b> Fichier CLASS 608 octet(s)	 <b>AlarmDelegate\$3.class</b> Fichier CLASS 630 octet(s)	 <b>AlarmDelegate\$4\$1.class</b> Fichier CLASS 790 octet(s)
 <b>AlarmDelegate\$4.class</b> Fichier CLASS 2.06 Ko	 <b>AlarmDelegate.class</b> Fichier CLASS 3.49 Ko	 <b>AlarmDelegate</b> Fichier JAVA 5.59 Ko	 <b>AlarmHelper.class</b> Fichier CLASS 2.29 Ko
 <b>AlarmHelper</b> Fichier JAVA 2.12 Ko	 <b>AlarmHolder.class</b> Fichier CLASS 814 octet(s)	 <b>AlarmHolder</b> Fichier JAVA 696 octet(s)	 <b>AlarmOperations.class</b> Fichier CLASS 202 octet(s)
 <b>AlarmOperations</b> Fichier JAVA 318 octet(s)	 <b>AlarmPOA.class</b> Fichier CLASS 2.30 Ko	 <b>AlarmPOA</b> Fichier JAVA 2.54 Ko	 <b>AlarmPOATie.class</b> Fichier CLASS 1.00 Ko
 <b>AlarmPOATie</b> Fichier JAVA 1.18 Ko	 <b>DelegationServer.class</b> Fichier CLASS 2.53 Ko	 <b>DelegationServer</b> Fichier JAVA 1.70 Ko	 <b>GUIClient\$1.class</b> Fichier CLASS 610 octet(s)
 <b>GUIClient\$2.class</b> Fichier CLASS 596 octet(s)	 <b>GUIClient\$3.class</b> Fichier CLASS 906 octet(s)	 <b>GUIClient.class</b> Fichier CLASS 3.98 Ko	 <b>GUIClient</b> Fichier JAVA 4.21 Ko

## Quelques captures du code :

## DelegationServer :

```
public class DelegationServer {
    private ORB orb;
    private POA rootPOA;
    private void initializeORB(String[] args) {
        Properties props = getProperties();
        orb = ORB.init(args, props);
        try {
            rootPOA = POAHelper.narrow(orb.
                resolve_initial_references("RootPOA"));
        } catch (org.omg.CORBA.ORBPackage.InvalidName ex) { }
    }
    public DelegationServer(String[] args) {
        try {
            AlarmDelegate cd= new AlarmDelegate();
            initializeORB(args);
            NamingContextExt nc = NamingContextExtHelper.narrow(
                orb.resolve_initial_references("NameService"));
            AlarmPOATie c_impl = new AlarmPOATie(cd);
            Alarm c = c_impl._this(orb);
            //ici
            nc.unbind(nc.to_name("Home.Object"));
            NameComponent[] name = nc.to_name("Home.Object");
            NamingContext nc1 = nc.bind_new_context(name);
            NameComponent[] name1 = nc.to_name("Alarm.Fonction");
            nc1.rebind(name1,c);

            out.println("Server started. Stop: Ctrl-C");
            rootPOA.the_POAManager().activate();
            orb.run();
        } catch (Exception ex) {
            out.println("Exception: " + ex.getMessage());
            exit(1);
        }
    }
    public static void main(String args[]) {
        new DelegationServer(args);
    }
}
```

## GUIClient

```
public void createGUI(String[] args) {
    try {
        initializeORB(args);
        NamingContextExt nc = NamingContextExtHelper.narrow(
            orb.resolve_initial_references("NameService"));

        NameComponent[] name = nc.to_name("Home.Object/Alarm.Fonction");

        org.omg.CORBA.Object obj = nc.resolve(name);
        this.c = AlarmHelper.narrow(obj);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 295, 178);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        GridBagLayout gbl_contentPane = new GridBagLayout();
        gbl_contentPane.columnWidths = new int[]{0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
        gbl_contentPane.rowHeights = new int[]{0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
        gbl_contentPane.columnWeights = new double[]{0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, Double.MIN_VALUE};
        gbl_contentPane.rowWeights = new double[]{0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, Double.MIN_VALUE};
        contentPane.setLayout(gbl_contentPane);

        JSpinner spinner = new JSpinner();
        spinner.setModel(new SpinnerNumberModel(0, 0, 11, 1));
        spinner.setFont(new Font("OCR A Extended", Font.PLAIN, 50));
        GridBagConstraints gbc_spinner = new GridBagConstraints();
        gbc_spinner.insets = new Insets(0, 0, 5, 5);
        gbc_spinner.gridx = 4;
        gbc_spinner.gridy = 5;
        contentPane.add(spinner, gbc_spinner);

        JButton btnNewButton = new JButton("Time Now");
        btnNewButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent arg0) {
                c.LocalTime ();
            }
        });
    }
}
```

```

JSpinner spinner_1 = new JSpinner();
spinner_1.setFont(new Font("OCR A Extended", Font.PLAIN, 50));
spinner_1.setModel(new SpinnerNumberModel(0, 0, 59, 1));
GridBagConstraints gbc_spinner_1 = new GridBagConstraints();
gbc_spinner_1.insets = new Insets(0, 0, 5, 5);
gbc_spinner_1.gridx = 5;
gbc_spinner_1.gridy = 5;
contentPane.add(spinner_1, gbc_spinner_1);
GridBagConstraints gbc_btnNewButton = new GridBagConstraints();
gbc_btnNewButton.insets = new Insets(0, 0, 0, 5);
gbc_btnNewButton.gridx = 4;
gbc_btnNewButton.gridy = 8;
contentPane.add(btnNewButton, gbc_btnNewButton);

JButton btnNewButton_1 = new JButton("Set alarme");

btnNewButton_1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        int i = (int) spinner.getValue();
        int j = (int) spinner_1.getValue();
        c.setAlarm(i,j);
    }
});

GridBagConstraints gbc_btnNewButton_1 = new GridBagConstraints();
gbc_btnNewButton_1.insets = new Insets(0, 0, 0, 5);
gbc_btnNewButton_1.gridx = 5;
gbc_btnNewButton_1.gridy = 8;
contentPane.add(btnNewButton_1, gbc_btnNewButton_1);

}

catch (BAD_PARAM ex) {
    out.println("Narrowing failed");
    exit(3);
} catch (Exception ex) {
    out.println("Exception: " + ex.getMessage());
    exit(1);
}

```

2-On compile les fichiers qu'on a écrit à savoir l'AlarmOperations, DelegationServer et enfin le GUIClient.

3- Usage de l'outil orbd du JDK et invocation avec spécification du numéro de port :

**orbd -ORBInitialPort 777**

4-Exécution du Serveur :

**Java DelegationServer -ORBInitRef  
NameService=corbaloc::localhost:777/NameService**

5-Exécution du Client :

**Java GUIClient -ORBInitRef  
NameService=corbaloc::localhost:777/NameService**

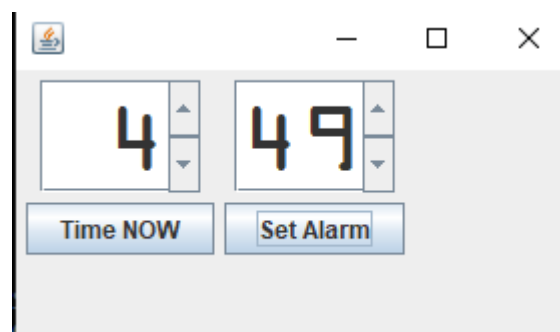
### Exceptions :

On a trouvé une exception lors de la compilation et on a résolu par l'ajout de **unbind** dans le fichier DelegationServer :

```
nc.unbind(nc.to_name("Home.Object"));
```

### Les Interfaces :

Après l'exécution correcte de GUIClient une première interface s'affiche :



Le chiffre à gauche signifie l'heure et l'autre la minute de l'alarme voulu.

Il y a deux boutons Time NOW pour afficher l'heure actuelle et Set Alarm pour

Mettre une alarme avec l'heure et la minute qu'on veut. (**Voir la vidéo de l'exécution**).

Après la clique sur le bouton Time NOW une autre interface s'affiche (l'heure actuelle)



Quand l'heure de l'alarme vient une interface s'affiche avec une sonnerie :



**FIN**