

## tp5

Lectures préalables :	Thèmes du TP :
<ul style="list-style-type: none"> <li>Le pattern <a href="#">Chaîne de responsabilités</a> (note 25).</li> </ul>	<ul style="list-style-type: none"> <li>Les threads</li> <li>Les requêtes HTTP</li> <li>Le pattern Chaîne de responsabilités</li> </ul>

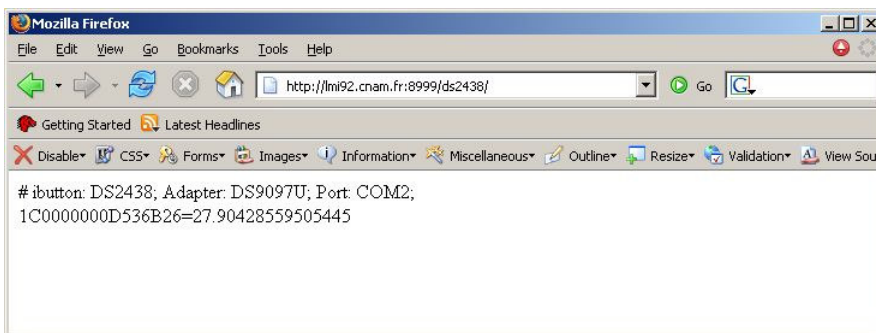
- Visualisez le sujet en ouvrant `index.html` du répertoire qui a été créé à l'ouverture de `tp5.jar` par BlueJ; vous aurez ainsi accès aux différents liens qui sont proposés pour vous aider, et aux applettes.
- Soumettez chaque question à l'outil d'évaluation `jnews`.



## Acquisition cyclique

Il s'agit de lire les valeurs d'un capteur accessible sur internet, en protocole HTTP.

- Ce capteur est un [DS2438](#) qui délivre le taux d'[HumiditéRelative](#) d'un bureau au CNAM/Paris 75003.
- Il est relié au serveur `lmi92.cnam.fr` par un adaptateur port série <--> bus [1-Wire](#) de [Dallas Semiconductor/MAXIM](#).
- Il est accessible à l'URL : <http://lmi92.cnam.fr:8999/ds2438/>
- Un exemple de requête depuis un navigateur :

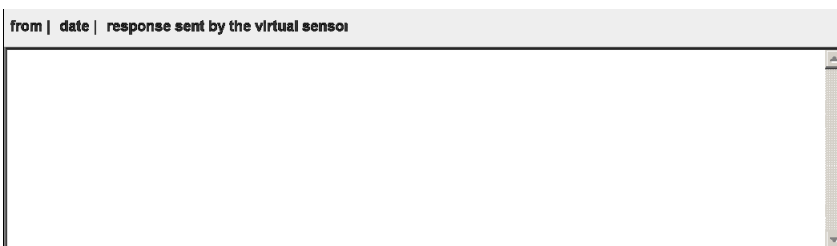


le source de cette page :

```
# ibutton: DS2438; Adapter: DS9097U; Port: COM2;<br>1C000000D536B26=27.90428559505445
```

(1C000000D536B26 représente le numéro du composant, 27.90428559505445 le taux d'humidité relative)

- Notez que si ce service est inaccessible, ( coupure réseau, capteur en panne, etc ... ) l'applette ci-dessous simule (et réagit comme) le capteur; son url est <http://localhost:8999/ds2438/> Vous pouvez cliquer ! (cette applette contient un serveur web, installée sur votre poste de travail; observez les traces dans le cadre ci-dessous)
  - Attention !** Dans un navigateur à l'ESIEE, si le lien ci-dessus ne fonctionne pas, `localhost` doit être remplacé par le nom de votre PC (`pc5103b`, par exemple).
  - Si cette applette ne fonctionne pas (probablement à cause d'une mauvaise configuration du navigateur), vous pouvez exécuter cette commande depuis le répertoire du tp :  
`repertoire_tp4> java -cp AppliSimulateurDS2438.jar question1.AppliSimulateurDS2438`  
 (l'url est ensuite identique <http://localhost:8999/ds2438/> )



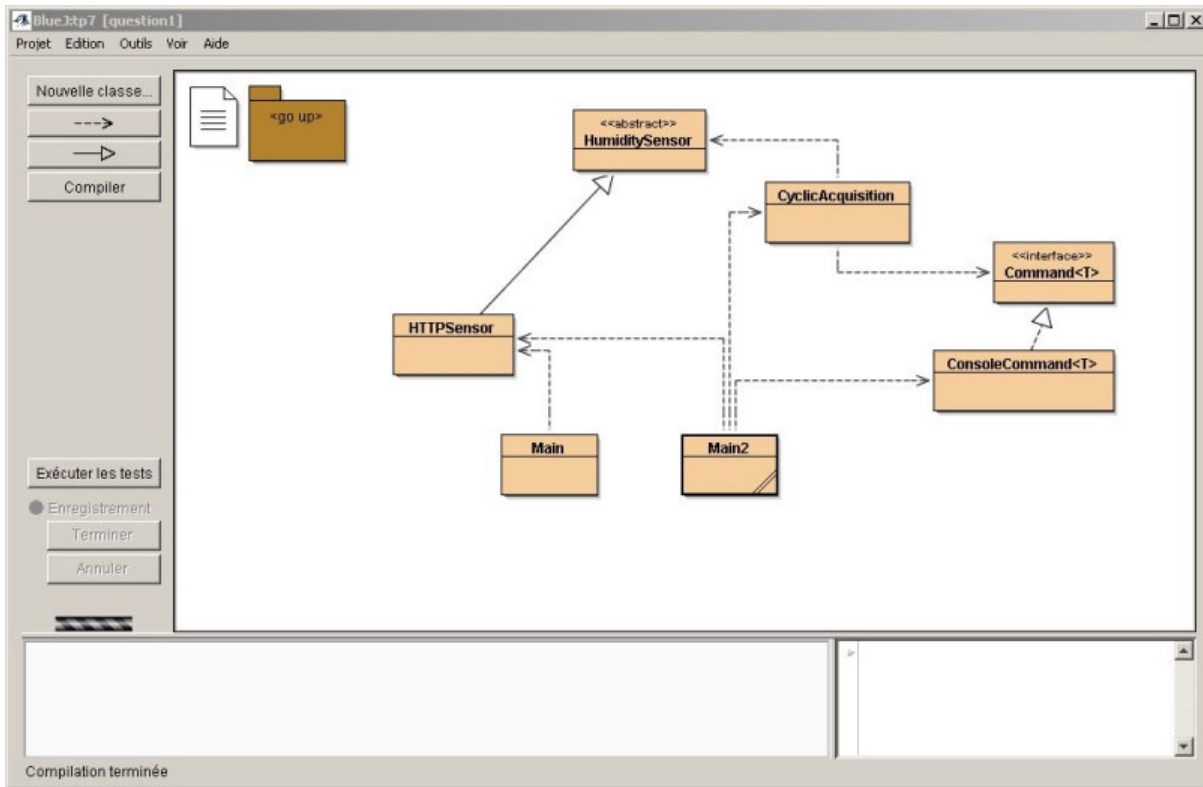
## Simulateur du capteur d'humidité sur le Web

Cette applette contient un serveur Web sur le port 8999, par défaut inaccessible depuis un autre poste.

**from** : le site client, localhost ou 127.0.0.1

**date** de la requête

**response** : la réponse retournée au client HTTP



Architecture présentée :

**HumiditySensor** reflète ce que l'on attend d'un capteur : une valeur, et la période minimale de lecture

**HTTPSensor** le capteur au protocole HTTP, d'autres types capteurs pourraient être installés

**Main** et **Main2** sont des applications Java, qui doivent s'exécuter dans une fenêtre DOS

**CyclicAcquisition** effectue une acquisition périodique de la valeur du capteur et transmet cette information

**Command<T>** est une interface qui propose le traitement d'une information de type T, ici une valeur du capteur

**ConsoleCommand<T>** se contente d'afficher T et l'heure courante

question1 .1) Complétez la classe **Main**, en affichant le résultat de 2 requêtes,

en mode "réel" ou sinon, en mode simulé.

**ouvrez une fenêtre DOS>**

Si DOS ne trouve pas la commande java, tapez :

```
SET PATH="C:\Program Files\java\jdk...\bin";%PATH%
```

où jdk... est le répertoire de la dernière version du JDK. (pas du JRE !)

1. Classes préalablement compilées avec BlueJ
2. Placez-vous dans le répertoire du TP
3. exécutez la commande suivante

```
java -cp . question1.Main http://localhost:8999/ds2438/ en mode simulé
```

- ou bien `java -cp . question1.Main http://lmi92.cnam.fr:8999/ds2438/`

**Aide : si vous êtes en mode simulé, n'oubliez pas de laisser votre navigateur ouvert avec cet énoncé qui contient l'applette/simulateur...**

**Aide : Erreur 503 ==> mauvais nom de machine**

question1 .2) Complétez la classe **HTTPSensor** en implémentant la méthode `value()` ;

Cette méthode a la signature suivante :

```
public float value() throws Exception;
```

**La valeur retournée aura un seul chiffre après la virgule (non par arrondi, mais par troncature).**

Pour répondre à cette question, vous pouvez utiliser la classe [java.util.StringTokenizer](#)

(Ne pas utiliser split. Quel délimiteur choisir ? Quel token contient alors l'information désirée ?).

Complétez de nouveau la classe **Main**, en remplaçant le résultat des 2 requêtes par seulement les 2 valeurs.

question1 .3) Complétez la classe **CyclicAcquisition**, contenant un **Thread** local.

Ce Thread permet l'acquisition cyclique du taux d'humidité.

Remarquez le constructeur qui accepte en paramètres un capteur, une période, et une commande.

La période d'acquisition doit être respectée en utilisant la méthode `minimalPeriod()` de la classe `HumiditySensor`, cette version néglige le temps de communication : Requête HTTP et analyse du résultat.

Si une exception survient pendant la requête, retourner la valeur -1.

L'information reçue est affichée sur la console par une implémentation de l'interface `Command<T>`, ici `ConsoleCommand<T>`.

Utilisez la classe `Main2` pour tester `CyclicAcquisition`; le programme doit s'arrêter au bout de 2 secondes.

**ouvrez une fenêtre DOS>**

1. placez-vous dans le répertoire du tp
2. exécutez les commandes suivantes
  1. Classes préalablement compilées avec `bluej`
  2. `java -cp . question1.Main2 http://localhost:8999/ds2438/` en mode simulé
    - ou bien `java -cp . question1.Main2 http://lmi92.cnam.fr:8999/ds2438/`

**note : si vous êtes en mode simulé, n'oubliez pas de laisser votre navigateur ouvert avec cet énoncé qui contient l'applette/simulateur...**

Un exemple de traces possible, ici en utilisant le simulateur (la période minimale est de 500ms)

```

C:\WINNT\system32\cmd.exe
G:\progAvancee\tp7_correction>java -cp . question1.Main2 http://localhost:8999/ds2438/
[09:50:02] ds2438.value : 12.8
[09:50:02] ds2438.value : 95.8
[09:50:03] ds2438.value : 0.9
[09:50:03] ds2438.value : 86.7
[09:50:04] ds2438.value : 13.8
[09:50:04] ds2438.value : 61.7
[09:50:05] ds2438.value : 9.0
[09:50:05] ds2438.value : 20.4
[09:50:06] ds2438.value : 1.9
[09:50:07] ds2438.value : 7.7
[09:50:07] ds2438.value : 40.2
[09:50:08] ds2438.value : 13.9
[09:50:08] ds2438.value : 23.0
[09:50:09] ds2438.value : 2.6
[09:50:09] ds2438.value : 99.6
[09:50:10] ds2438.value : 22.7
[09:50:10] ds2438.value : 70.0
[09:50:11] ds2438.value : 41.6
[09:50:11] ds2438.value : 86.7
  
```

Le [simulateur](#) inclus dans cet énoncé doit afficher :

question2

## Chaîne de responsabilités

L'information reçue, le taux d'humidité, est maintenant transmise à différents consommateurs (responsables...). Les consommateurs sont chaînés entre eux. Ce type de conception est issu du pattern "[chaîne de responsabilités](#)". Le principe est de transmettre l'information à une chaîne de consommateurs, chaque consommateur décide s'il doit laisser passer l'information vers son successeur ou bien arrêter sa propagation (*l'ordre a donc de l'importance* ...).

question2

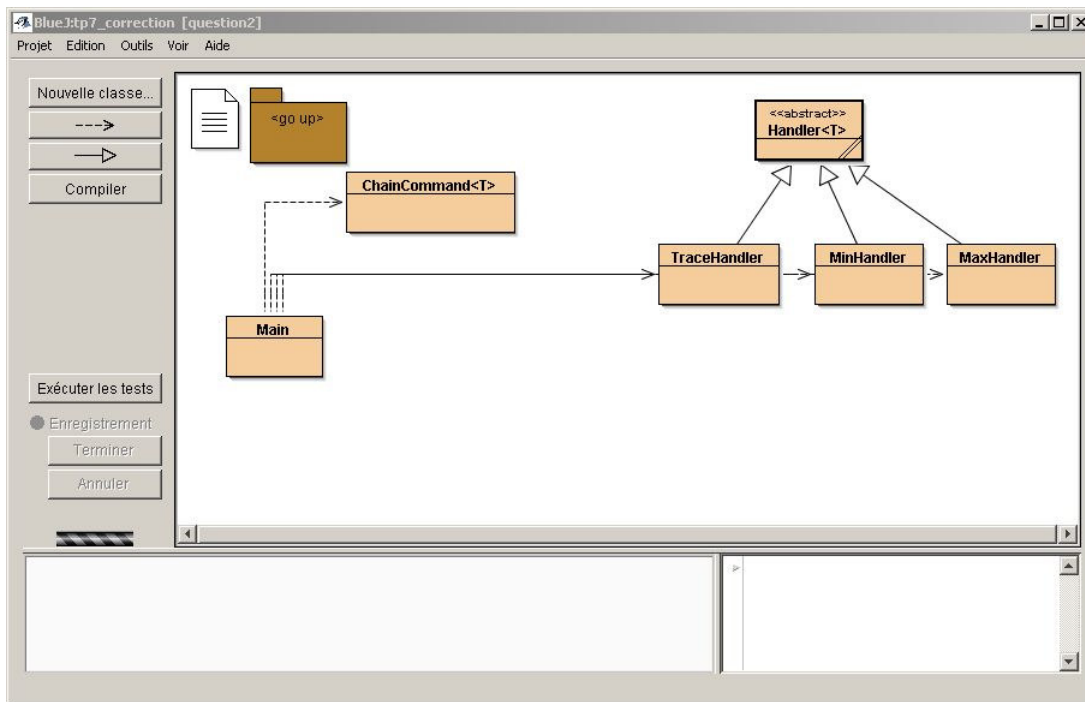
### .1) Proposez une première "Chaîne de responsabilités" constituée de 3 "Handlers"

liée à notre application de lecture cyclique du taux d'humidité et complétez les classes `Main`, `MinHandler` et `MaxHandler`.

- **TraceHandler** : affiche la date et l'heure courante ainsi que la valeur du taux d'humidité relative sur la console et transmet l'information à son successeur),

- **MinHandler** : détecte et affiche une valeur d'humidité relative strictement inférieure à 35%
- **MaxHandler** : détecte une valeur d'humidité (non détectée par MinHandler) strictement inférieure à 100% (100% : c'est le point de rosée !!!)

**Attention !** respectez les formats d'affichage montrés dans la fenêtre DOS ci-dessous.



**Architecture retenue :**

- Handler<T>**, **TraceHandler**, **MinHandler**, **MaxHandler** les gestionnaires attendus
- ChainCommand<T>** convertit la "commande" vers une chaîne de responsabilités
- Main** est une application Java, qui doit s'exécuter dans une fenêtre DOS

**ouvrez une fenêtre DOS>**

1. placez-vous dans le répertoire du tp
2. exécutez les commandes suivantes
  1. Classes préalablement compilées avec bluej
  2. java -cp . question1.**Main**2 http://localhost:8999/ds2438/ en mode simulé

## Une Trace d'exécution possible

```

C:\WINNT\system32\cmd.exe
G:\prog\avancees\tp7_correction>java -cp . question2.Main http://localhost:8999/ds2438/
[05/07/05-15:33] ds2438.value : 7.3
minimum ... ds2438.value : 7.3
[05/07/05-15:33] ds2438.value : 55.3
maximum .....ds2438.value : 55.3
[05/07/05-15:33] ds2438.value : 5.0
minimum ... ds2438.value : 5.0
[05/07/05-15:33] ds2438.value : 17.4
minimum ... ds2438.value : 17.4
[05/07/05-15:33] ds2438.value : 35.9
maximum .....ds2438.value : 35.9
[05/07/05-15:33] ds2438.value : 92.1
maximum .....ds2438.value : 92.1
[05/07/05-15:33] ds2438.value : 26.6
minimum ... ds2438.value : 26.6
[05/07/05-15:33] ds2438.value : 80.6
  
```

- Pour information, ces quelques lignes Java fournissent la date et heure courante :

```

Calendar c = Calendar.getInstance();
DateFormat df = DateFormat.getDateInstance(DateFormat.SHORT, Locale.FRANCE);
DateFormat dt = DateFormat.getTimeInstance(DateFormat.SHORT, Locale.FRANCE);
String date = df.format(c.getTime()) + "-" + dt.format(c.getTime());
  
```

avec `import java.util.Calendar;`  
`import java.text.DateFormat;`  
`import java.util.Locale;`

- Pour plus de détails, voir la classe [Calendar](#).

**question2 .2) Ajoutez le maillon FileHandler dans la chaîne et au bon endroit.**

Ce 'handler' est chargé de la sauvegarde sur fichier des mesures obtenues.

- **FileHandler** enregistre sur fichier les couples (date, valeur) ; il engendre un fichier à la syntaxe HTML. Ce fichier est sauvegardé par exemple toutes les X mesures. Votre fichier de mesures [mesures.html](#) (Afficher le source html) pourra être ensuite lu par ce navigateur. (cette page est réactualisée toutes les 30secondes)

05/07/05-16:05	48.0
05/07/05-16:05	33.2
05/07/05-16:05	94.0
05/07/05-16:05	54.8
05/07/05-16:05	91.5
05/07/05-16:05	66.2
05/07/05-16:05	4.3
05/07/05-16:05	54.1
05/07/05-16:05	7.2
05/07/05-16:06	44.3

## Documentations annexes :

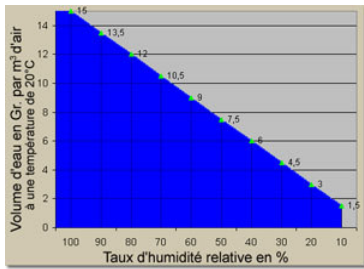
### Humidité Relative : définition extraite de [http://www.credo.fr/fr/monde\\_credo\\_world/humidite\\_relative.htm](http://www.credo.fr/fr/monde_credo_world/humidite_relative.htm)

L'Humidité Relative (HR) exprime le rapport entre la quantité effective de vapeur d'eau dans un volume donné d'air et la quantité maximale que ce volume peut contenir à la même température.

L'eau s'évapore dans l'atmosphère jusqu'à ce que soit atteinte une proportion maximale de vapeur d'eau dans l'air, dite humidité saturante.

Quand cette condition est atteinte, la moindre chute de température provoque la condensation de la vapeur et l'apparition de minuscules gouttes d'eau. Il s'agit du phénomène de rosée.

La saturation de l'air en vapeur d'eau (ou point de rosée) correspond donc à 100 % d'humidité relative.



La quantité de vapeur d'eau dans l'air à 100 % HR est d'environ 15 grammes d'eau par mètre cube d'air.

Quand la quantité de vapeur d'eau dans l'air est en deçà de la saturation, l'humidité relative est inférieure à 100 %.

A peu de choses près, une HR de 70 % correspond à une quantité d'eau par mètre cube d'air de 70 % de la quantité présente à saturation: environ 10,5 grammes d'eau par mètre cube d'air.

## Design Pattern et UML

extrait de <http://www.eli.sdsu.edu/courses/spring04/cs635/notes/chain/chain.html>

