

Objectifs :

- Savoir récupérer des adresses IP des machines connectées dans le même sous réseaux, connaître l'adresse IP des serveurs dans le web.
- Savoir récupérer les différentes composantes d'un URL.
- Savoir établir une connexion HttpURLConnection. Et de modifier les propriétés de la connexion

Matériel nécessaire :

- Postes informatiques sous Windows dotés de cartes réseaux.

IP Adresse

Travail demandé :

- Ecrire un programme permettant de reconnaître si une adresse saisie est une adresse IP ou non.

```
package reseau;

import java.net.InetAddress;

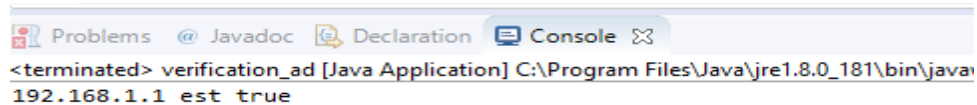
public class verification_ad {

    static boolean estCorrecte(String adr){
        String[] parties = adr.split("\\.");
        for(int i = 0 ; i < 4; i++){
            if(new Integer(parties[i])<0 || new Integer(parties[i])>255)
                return false;
        }
        return true;
    }

    public static void main(String[] args) {
        String adr= "192.168.1.1";

        System.out.println(adr+ " est "+estCorrecte(adr));
    }
}
```

⇒ Exécution :

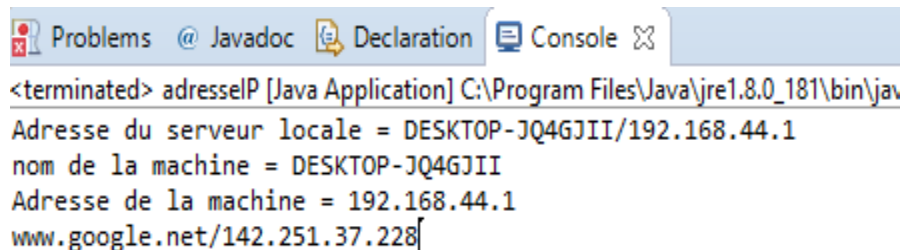


```
<terminated> verification_ad [Java Application] C:\Program Files\Java\jre1.8.0_181\bin\java
192.168.1.1 est true
```

- Ecrire un programme permettant de :
 1. Obtenir l'adresse du serveur local en appelant la méthode `getLocalHost()` de la classe `InetAddress`.
 2. Obtenir le nom de la machine en appelant la méthode `getHostName()`.
 3. Obtenir l'adresse IP en appelant la méthode `getHostAddress()`.
 4. Obtenir l'adresse IP du serveur Google donnée par son nom du domaine "www.google.net" en appelant la méthode `getByName()`

```
1 package reseau;
2
3 import java.io.IOException;
4
5
6
7
8
9 public class adresseIP {
10
11     public static void main(String[] args) throws IOException{
12
13
14         InetAddress ip= InetAddress.getLocalHost();
15         System.out.println("Adresse du serveur locale = "+ip );
16         System.out.println("nom de la machine = "+ip.getHostName());
17         System.out.println("Adresse de la machine = "+ip.getHostAddress() );
18         System.out.println(InetAddress.getByName("www.google.net"));
19     }
20 }
```

⇒ Exécution :



```
<terminated> adresseIP [Java Application] C:\Program Files\Java\jre1.8.0_181\bin\jav.exe
Adresse du serveur locale = DESKTOP-JQ4GJII/192.168.44.1
nom de la machine = DESKTOP-JQ4GJII
Adresse de la machine = 192.168.44.1
www.google.net/142.251.37.228
```

2.Ping

Travail demandé

1. Réaliser un programme qui fait un sort de ping avec la méthode `isReachable()` : tente un envoie ECHO REQUEST en ICMP, testez le programme sur les adresse : 127.0.0.1 , 133.192.31.42 , 145.154.42.58

```
1 package reseau;
2
3 import java.io.IOException;
4
5
6 public class ping {
7
8
9     public static void sendPingRequest(String ipAddress)throws Exception{
10         InetAddress adr = InetAddress.getByName(ipAddress);
11         System.out.println("Sending Ping Request to "+ipAddress);
12         if (adr.isReachable(12))
13             System.out.println("Host is reachable");
14         else
15             System.out.println("host unreachable");
16     }
17     public static void main(String[] args)throws Exception{
18         String ipAddress = "127.0.0.1";
19         sendPingRequest(ipAddress);
20         ipAddress = "133.192.31.42";
21         sendPingRequest(ipAddress);
22         ipAddress = "145.154.42.58";
23         sendPingRequest(ipAddress);
24     }
25 }
```

⇒ Exécution :

```
<terminated> ping [Java Application] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (Apr 20, 2022, 12:07:03 AM)
Sending Ping Request to 127.0.0.1
Host is reachable
Sending Ping Request to 133.192.31.42
host unreachable
Sending Ping Request to 145.154.42.58
host unreachable
<
```

3.URLS

Travail demandé

1. Ecrire un programmes permettant d'afficher tous les parties constituant l'URL suivant:

<https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.43.4903&rep=rep1&type=pdf>

```
1 package reseau;
2
3 import java.io.IOException;
4
5
6 public class Urls {
7
8     public static void main(String[] args) throws IOException {
9         URL url = new URL("https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.43.4903&rep=rep1&type=pdf");
10        System.out.println(url.toString());
11        System.out.println("getProtocol() : " + url.getProtocol());
12        System.out.println("getHost() : " + url.getHost());
13        System.out.println("getPort() : " + url.getPort());
14        System.out.println("getFile() : " + url.getFile());
15    }
16 }
17
18
19
```

⇒ Exécution :

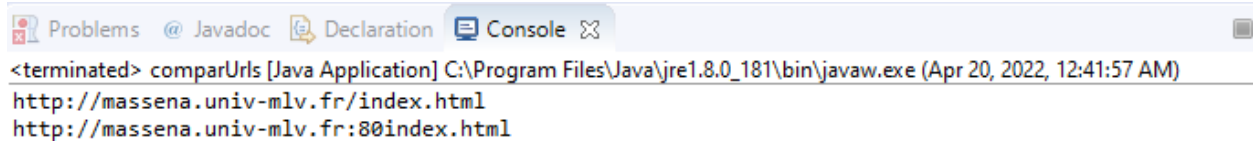
```
Problems @ Javadoc Declaration Console
<terminated> Urls [Java Application] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (Apr 20, 2022, 12:26:06 AM)
https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.43.4903&rep=rep1&type=pdf
getProtocol() : https
getHost() : citeseerx.ist.psu.edu
getPort() : -1
getFile() : /viewdoc/download?doi=10.1.1.43.4903&rep=rep1&type=pdf
```

- Grace à la méthode **getProtocol()** qui fait parti de la classe **URL** ,on peut retourner le protocole d'un URL spécifique
- La méthode **getHost()** qui fait parti de la classe **URL** et qui retourne le nom de la machine d'un URL spécifique
- La méthode **getPort()** qui retourne le numéro de port
- La méthode **getFile()** qui retourne le nom du fichier d'un URL spécifique

2.Exécuter le programme suivant puis commentez le résultat ?

```
1 package reseau;
2
3+ import java.net.MalformedURLException;
4
5
6 public class comparUrls {
7
8-     public static void main( String [] args) throws MalformedURLException {
9         URL u1= new URL("http://massena.univ-mlv.fr/index.html");
10        System.out.println(u1.toString());
11        URL u2= new URL("http","massena.univ-mlv.fr",80,"index.html");
12        System.out.println(u2.toString());
13    }
14 }
15
16
17
```

⇒ Exécution :



```
<terminated> comparUrls [Java Application] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (Apr 20, 2022, 12:41:57 AM)
http://massena.univ-mlv.fr/index.html
http://massena.univ-mlv.fr:80index.html
```

⇒ La seule différence entre url1 et url2 c'est que la deuxième url contient le port et même lorsque on tape le protocole http et on fait pas ' // ', il se fait automatiquement .

3. Afin d'établir une connexion HttpURLConnection, nous créons une classe permettant de valider, établir une connexion de type HttpURLConnection à partir d'un Url donnée, puis récupérer les informations du header du contenu de cet Url.

Exécuter le programme suivant puis commentez les résultats.

```

package reseau;

import java.io.IOException;

public class urlconec {

    public static void main( String args []) throws IOException{
        URL url = new URL("http://www.google.com");
        URLConnection urlc = url.openConnection();
        HttpURLConnection cnx = null ;
        if(urlc instanceof HttpURLConnection){
            cnx = (HttpURLConnection)urlc;
        }
        else{
            System.out.println("Please,entrez une Http URL valide");
            return;
        }
        for(int i = 1; i <= 9; i++){
            System.out.println(cnx.getHeaderFieldKey(i) + " = " + cnx.getHeaderField(i));
        }
    }
}

```

⇒ Exécution :

```

Problems | @ Javadoc | Declaration | Console
<terminated> urlconec [Java Application] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (Apr 20, 2022, 12:48:00 AM)
P3P = CP="This is not a P3P policy! See g.co/p3phelp for more info."
Server = gws
X-XSS-Protection = 0
X-Frame-Options = SAMEORIGIN
Set-Cookie = 1P_JAR=2022-04-19-23; expires=Thu, 19-May-2022 23:48:02 GMT; path=/; domain=.google.com; Secure

```