

République Algérienne Démocratique et populaire

Ministère de l'enseignement supérieur et la recherche scientifique



**UNIVERSITE AMMAR THLIDJI  
LAGHOUAT**

**THEME**

**UN OUTIL POUR ADMINISTRATION  
DISTANCE VIA SNMP**

**Réaliser par :**

1. Merad habiba .
2. Sherifi sara .

**Diriger par :**

– Galouma younes .

**2010 - 2011**

République Algérienne Démocratique et populaire

Ministère de l'enseignement supérieur et la recherche scientifique



**UNIVERSITE AMMAR THLIDJI  
LAGHOUAT**

**THEME**

**UN OUTIL POUR ADMINISTRATION  
DISTANCE VIA SNMP**

**Réaliser par :**

1. Merad habiba .
2. Sherifi sara .

**Diriger par :**

– Galouma younes .

**2010 - 2011**

# Dédicace

*Je dédie ce modeste travail:*

*À mes très chers parents Pour tout l'amour dont vous m'avez entouré, pour tout ce que vous avez fait pour moi. Je ferai de mon mieux pour rester un sujet de fierté à vos yeux avec l'espoir de ne jamais vous décevoir. Que ce modeste travail, soit l'exaucement de vos vœux tant formulés et de Vos prières quotidiennes.*

*À mon cher frère Sidali et mes sœurs Lamia et Zohra, Lina , aucune dédicace ne serait exprimer assez profondément ce que je ressens envers vous. Je vous dirais tout simplement, un grand merci, je vous aime.*

*À mon cher fiancé \*MOHAMED BADAOUTI\* ma chère belle mère Nadia et mon beau père Nasser et mes belles sœurs Nesrine et Rima Zohra , et mon beau frère Soufiane aucune dédicace ne serait exprimer assez profondément ce que je ressens envers vous. Je vous dédie ce travail en vous souhaitant un avenir radieux et plein de bonnes promesses Je vous dirais tout simplement, un grand merci, je vous aime.*

*À ma meilleure amie de l'enfance Massaouda Boubelaa , en témoignage de l'amitié sincère qui nous a liées et des bons moments passés ensemble. Je vous dédie ce travail en vous souhaitant un avenir radieux et plein de bonnes promesses.*

*À toutes mes amies d'études spécialement Benaya wafa ,Réche mazou ,Sone ihsane.*

*À ma grande mère, mes oncles, mes tantes spécialement (nacira et zohra ), mes cousins et mes cousines (amiana ,hadjer,zohra) Vous occupez ne place particulière dans mon cœur. Je vous dédie ce travail en vous souhaitant un avenir radieux, plein de bonheur et de succès.*

## **HABIBA MERAD**

### **Dédicace**

*Je dédie ce modeste travail:*

*À mes très chers parents Pour tout l'amour dont vous m'avez entouré, pour tout ce que vous avez fait pour moi. Je ferai de mon mieux pour rester un sujet de fierté à vos yeux avec l'espoir de ne jamais vous décevoir. Que ce modeste travail, soit l'exaucement de vos vœux tant formulés et de Vos prières quotidiennes.*

*À mon cher frère Sidali et mes sœurs Lamia et Zohra, Lina , aucune dédicace ne serait exprimer assez profondément ce que je ressens envers vous. Je vous dirais tout simplement, un grand merci, je vous aime.*

*À ma meilleure amie de l'enfance Massaouda Boubelaa , en témoignage de l'amitié sincère qui nos a liées et des bons moments passés ensemble. Je vous dédie ce travail en vous souhaitant un avenir radieux et plein de bonnes promesses.*

*À toutes mes amies d'études spécialement Benaya wafa ,Réche mazou ,Son ihsane*

## **HABIBA MERAD**

### **Dédicace**

*Je dédie ce modeste travail:*

*À mes très chers parents Pour tout l'amour dont vous m'avez entouré, pour tout ce que vous avez fait pour moi. Je ferai de mon mieux pour rester un sujet de fierté à vos yeux avec l'espoir de ne jamais vous décevoir. Que ce modeste travail, soit l'exaucement de vos vœux tant formulés et de Vos prières quotidiennes.*

*À mes chers frères mohamed et hamidou mes sœurs Rima et amina aucune dédicace ne serait exprimer assez profondément ce que je ressens envers vous. Je vous dirais tout simplement, un grand merci, je vous aime.*

*À toutes mes amies spécialement hafsa*

*Et kadidja*

A decorative border with repeating floral motifs surrounds the page. The top and bottom borders feature larger, more intricate floral designs at the corners. The left and right borders consist of smaller, repeating floral patterns.

***SARA SHERIFI***

# Table des matières --- ---

Remerciement

## **1 généralité sur réseaux informatique**

1.1 Introduction	-----03
1.2 L'évolution permanente du réseau	-----04
1.3 Gestion de réseau	-----05

## **2 le protocole SNMP**

2.1 Introduction	-----06
2.2 Les protocole de la gestion réseau	-----06
2.3 Comparaison des protocoles	-----08
2.4 Historique de snmp	-----08
2.5 Évolution de version snmp	-----08
2.6 Fonctionnement de snmp	-----12
2.7 Le fichier mib	-----14
2.8 Installation de SNMP sous Windows	-----17
2.9 Installation de SNMP sous linux	-----21

## **3 parties programmation**

3.1 Introduction	-----23
3.2 Description de l'outil	-----23
3.3 Environnement du développement	-----23
3.4 Langage de programmation	-----26
3.5 Exemple d'application	-----28

**Conclusion** -----30

**Références**

# Remerciements

Nous remercions notre bon *Dieu* de nous avoir accordé des connaissances de la science et de nous avoir aidé à réaliser ce modeste travail.

Nous tenent tout particulièrement à remercier le chef de département informatique *Mr DAHARI*.

Nous remerciant également notre encadreur *Mr .GALOUMA YOUNSE* pour nous avoir guidé et encouragé ainsi pour leurs soutiens, pour tous le temps qu'il nous a octroyés et pour tous les conseils qu'il nous a prodigués.

A tous les membres du jury.

A nos parents.

Nous remerciant également *Mr .BADAoui MOHAMED TAHER* pour nous avoir encouragés ainsi pour leurs soutiens.

*Merci à tous.*

# Remerciements

Nous remercions notre bon *Dieu* de nous avoir accordé des connaissances de la science et de nous avoir aidé à réaliser ce modeste travail.

Nous tenent tout particulièrement à remercier le chef de département informatique *Mr DAHARI*.

Nous remerciant également notre encadreur *Mr .GALOUMA YOUNSE* pour nous avoir guidé et encouragé ainsi pour leurs soutiens, pour tous le temps qu'il nous a octroyés et pour tous les conseils qu'il nous a prodigués.

A tous les membres du jury.

A nos parents.

*Merci à tous.*

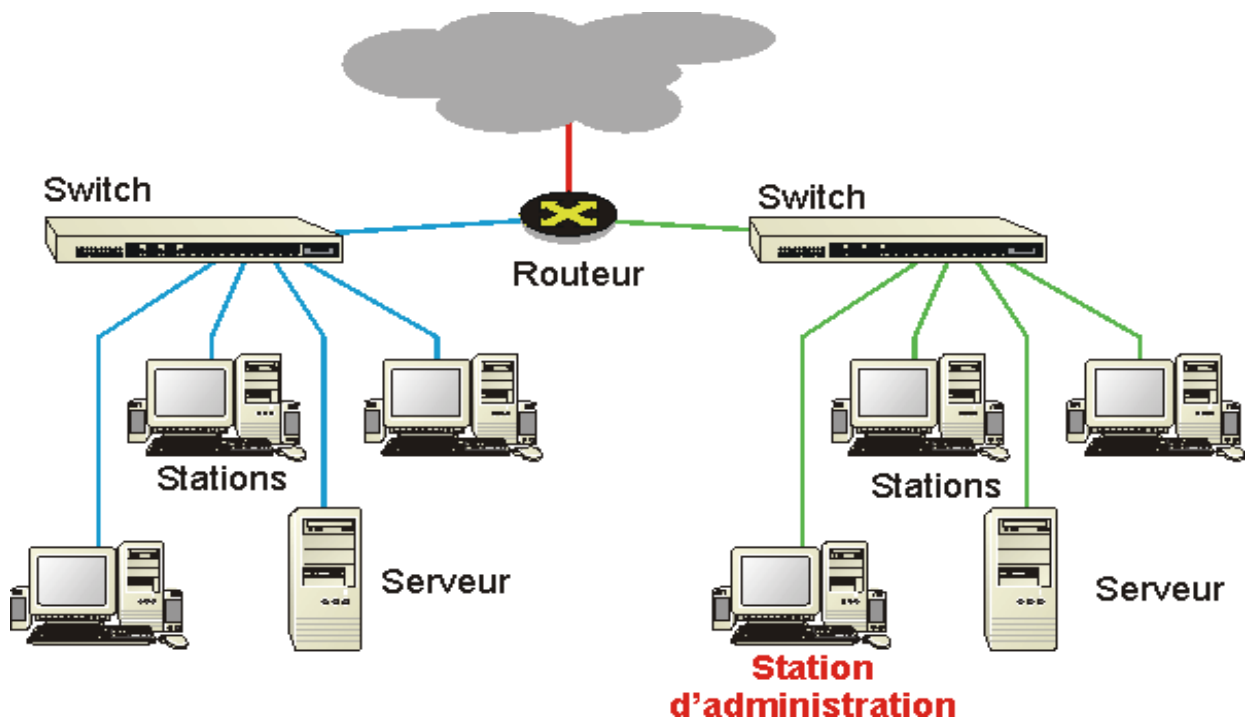
## 1-Généralité sur réseaux informatique :

### 1.1-Introduction :

Un réseau est un ensemble ordonné d'objets ou de personnes ayant tous une même tâche bien définie. On parle ainsi de réseaux d'espionnage, de réseaux ferrés, de réseaux informatiques, etc....

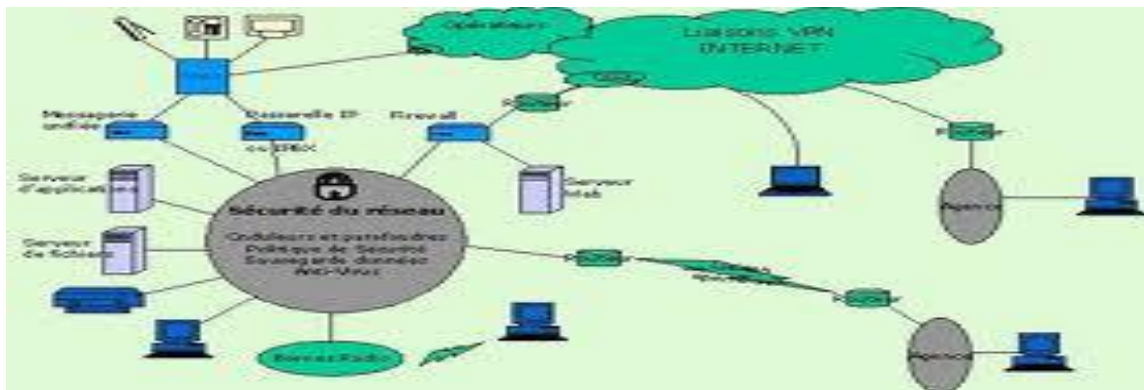
On le représente typiquement sous forme de graphes. Chaque objet ou personne constituant ce réseau est alors un nœud du graphe chaque nœud a pour tâche le traitement d'informations.

Un nœud peut parfois représenter un réseau en lui-même : on parle alors d'un sous-réseau de ce réseau.



## 1.2-L'évolution permanente des réseaux :

Depuis la création des premiers réseaux de communication informatisée dans les années 70 aux Etats-Unis ceux-ci ont énormément évolué. Premièrement, les matériels d'infrastructures réseaux (routeurs, concentrateurs, modems, etc..) ont des capacités de traitement de plus en plus importantes et offrent des fonctions de plus en plus complexes : qualité de service, réseaux virtuels, sécurisation des communications, débit accru, etc....



Par ailleurs, outre l'évolution technique, l'évolution la plus flagrante est l'importance des réseaux informatiques et le nombre de nœuds interconnectés. Il suffit de prendre l'exemple d'Internet : alors qu'il était réservé aux militaires dans un premier temps, ce réseau s'est ouvert aux universités dans les années 70 puis aux commerciaux dans les années 80.

Aujourd'hui, de plus en plus de produits ont un support de connexion à Internet, ordinateurs, téléphones, assistants personnels, tous ces outils sont aujourd'hui reliés en réseau et accroissent l'importance des réseaux de télécommunications mondiaux.

Alors qu'au début des années 90 Internet était constitué d'environ 400.000 nœuds, il y en aurait plus de 40.000.000 aujourd'hui.

## 1. 3-Gestion de réseau :

Pour les administrateurs et les programmeurs l'évolution des réseaux dont nous venons de parler pose un problème majeur : la manière dont ceux-ci sont gérés pratiquement. En effet, le nombre de nœuds ne cessant des étendre, il est important de pouvoir gérer tout ceux-ci le plus facilement possible .

Il est aujourd'hui nécessaire, étant donnée l'étendue des réseaux, de pouvoir le gérer à distance depuis son poste de travail et n'avoir à se déplacer qu'en dernier recours, lorsqu'une opération physique est nécessaire. Cela peut sembler exagéré, voire être perçu comme un luxe pour l'administrateur réseau, toutefois lorsque l'on a ne serait-ce qu'une centaine de nœuds à sa charge, il est important de pouvoir agir à distance dessus.



La centralisation de la gestion réseau permet également d'effectuer simultanément une même opération sur plusieurs matériels : si l'on souhaite appliquer une modification à un ensemble de matériels données, il suffit d'utiliser la liste de ce matériel dans notre programme et de l'exécuter depuis la machine de gestion. En l'absence de central, il aurait fallu se déplacer sur chaque matériel afin d'y appliquer ladite modification : la centralisation offre un gain de temps considérable.



## Environnement du développement :

L'outil doit s'intégrer facilement sous le Shell linux , pour cela on opte pour intégrer le programme dans des scriptes linux

### Scripte Shell :

Un Shell, quel qu'il soit, peut exécuter des commandes prises dans un fichier. Un fichier contenant des commandes pour le Shell est appelé un *script*. C'est en fait un *programme* écrit dans le langage du Shell. Ce langage comprend non seulement les commandes mais aussi des structures de contrôle (constructions conditionnelles et boucles).

Un script Shell est un fichier en mode texte. C'est-à-dire que ce n'est pas un fichier binaire, exécutable directement par la machine, mais il doit être interprété.

### L'interprétation d'un script

L'interprétation signifie que chaque commande contenue dans un script doit être lue par un programme, appelé *interpréteur* (et non interprète, bizarrement) ; l'interpréteur analyse chaque commande du script et la traduit « à la volée » en langage machine, ce qui permet l'exécution du script.

### L'édition d'un script

Un script étant un fichier en mode texte, il doit être créé avec un éditeur de texte. Un éditeur de texte est un programme dont la fonction est d'éditer du texte.

Mais quel éditeur choisir ?

Tout d'abord, certains il faut savoir que n'importe quel éditeur est capable d'ouvrir et d'écrire des scripts Shell, et vous pouvez tout à fait modifier avec n'importe quel éditeur de texte ce script Shell que vous avez écrit avec n'importe quel autre.

Mais il faut savoir aussi que éditeurs de texte sont plus appropriés que d'autres à l'écriture de. Par exemple, `nano` permet d'éditer des scripts comme tout autre éditeur, mais quand un script fait plus d'une dizaine de lignes, on commence à s'y perdre un peu. À l'inverse, `Emacs` et `Vim` offrent quelques fonctionnalités qui deviennent rapidement indispensables :

1. l'indentation ;
2. la coloration syntaxique.

### Rendre un script exécutable

Pour que le Shell sache comment l'interpréter, un script Shell doit commencer par la ligne:

```
#!/bin/sh
```

Il doit aussi avoir être exécutable (bit x). Le `#!/bin/sh` sur la première ligne indique que ce script doit être exécuté par le Shell `sh` dont on indique le chemin d'accès. Pour rendre un fichier exécutable, tapez :

```
Chaland ~ chmod u+x fichier
```

\*Shemat\*

## Langage de programmation:

Java est un langage de programmation informatique orienté objet créé par James Gosling et Patrick Naughton de Sun Microsystems. Mais c'est également un environnement d'exécution.

Java peut être séparée en deux parties. D'une part, votre programme écrit en langage Java et d'autre part, une machine virtuelle (JVM) qui va se charger de l'exécution de votre programme Java.

C'est cette plateforme qui garantit la portabilité de Java. Il suffit qu'un système ait une machine virtuelle Java pour que tout programme écrit en Java puisse fonctionner.

## Historique de java :

Les débuts de Java remontent à avril 1991, quand un petit groupe d'employés de Sun ont été déplacés dans un autre département pour travailler à un projet baptisé 'Green' voué à la réalisation d'application électronique commerciale.

L'équipe Green était libre de faire ce qu'elle voulait. Sa première mission fut de créer un appareil capable de

contrôler plusieurs équipements électroniques é la fois. Un langage fut créé à cette intention, le langage Oak, dont

les premiers pas remontent à aout 1991. Le nom Oak, trop proche d'une autre marque, fut rebaptisé Java en 1995.

L'annonce officielle de Java et HotJava date, quant à elle, du salon SunWorld 1995.

Le projet Green était

originellement orienté vers la création de systèmes d'exploitation pour Personal Digital Assistant, set-top boxes

etc.

Le projet initial fut dissout et la technologie recentre vers le CD-ROM, le multimédia en ligne et les environnements réseau.

Le maître d'œuvre de Java est James Gosling, qui commence par vouloir étendre le C++ avant de réaliser le

premier compilateur Java écrit en langage C. Arthur Van Hoff implanta le compilateur Java en Java lui-même à la

fin 1994.

Avec le langage Java, vous pouvez développer, des applications Desktop, développer des applets pour vos sites web, développer des sites en JSP, des applications pour téléphone mobile. La première chose à faire est bien évidemment d'apprendre à faire des applications standalones simples.

### Les caractéristiques du langage :

Java possède un certain nombre de de caractéristiques qui ont largement contribué à son énorme succès :

- java est interprétée :  
la source est compilée en pseudo code on byte code puis exécuté par un interpreteur  
java : Java Virtual Machine (JVM)
- java est potable :

### *Introduction :*

Le but de l'outil consiste à construire une interface confiante pour l'administrateur afin de piloter un appareil distant via le protocole snmp

Dans ce chapitre on veut montrer les différentes étapes de réalisation de notre outil depuis la définition des besoins jusqu'à l'implémentation et les tests finaux

### *Conception et développement :*

#### *Description de l'outil :*

En quoi s'agit notre outil ?

Notre logiciel est destiné à l'usage des administrateurs réseaux afin de rendre la manipulation des commandes snmp facile exécuté par un Shell . Le protocole SNMP est basé sur un fonctionnement asymétrique. Il est constitué d'un ensemble de requêtes, de réponses et d'un nombre limité d'alertes. Le manager envoie des requêtes à l'agent, lequel retourne des réponses. Lorsqu'un événement anormal surgit sur l'élément réseau, l'agent envoie une alerte (trap) au manager .Dans le but d'une plus grande efficacité dans la recherche avec une plus haute fiabilité et d'un gain de temps important ce trafic sera géré à partir d'une interface au lieu de la fenêtre Shell Bach .

#### *Description détaillé :*

### *Environnement du développement :*

L'outil doit s'intégrer facilement sous le Shell linux, pour cela on opte pour intégrer le programme dans des scripts linux.

### Scripte Shell :

Un Shell, quel qu'il soit, peut exécuter des commandes prises dans un fichier. Un fichier contenant des commandes pour le Shell est appelé un *script*. C'est en fait un *programme écrit dans le langage du Shell*. Ce langage comprend non seulement les commandes mais aussi des structures de contrôle (constructions conditionnelles et boucles).

Un script Shell est un fichier en mode texte. C'est-à-dire que ce n'est pas un fichier binaire, exécutable directement par la machine, mais il doit être interprété.

### L'interprétation d'un script :

L'interprétation signifie que chaque commande contenue dans un script doit être lue par un programme, appelé *interpréteur* (et non interprète, bizarrement) ; l'interpréteur analyse chaque commande du script et la traduit « à la volée » en langage machine, ce qui permet l'exécution du script.

### L'édition d'un script :

Un script étant un fichier en mode texte, il doit être créé avec un éditeur de texte. Un éditeur de texte est un programme dont la fonction est d'éditer du texte.

Mais quel éditeur choisir ?

Tout d'abord, certains il faut savoir que n'importe quel éditeur est capable d'ouvrir et d'écrire des scripts Shell, et vous pouvez tout à fait modifier avec n'importe quel éditeur de texte ce script Shell que vous avez écrit avec n'importe quel autre.

Mais il faut savoir aussi que éditeurs de texte sont plus appropriés que d'autres à l'écriture de. Par exemple, **nano** permet d'éditer des scripts comme tout autre éditeur, mais quand un script fait plus d'une dizaine de lignes, on commence à s'y perdre un peu. À l'inverse, **Emacs** et **Vim** offrent quelques fonctionnalités qui deviennent rapidement indispensables :

1. l'indentation ;
2. la coloration syntaxique.

### Rendre un script exécutable :

Pour que le Shell sache comment l'interpréter, un script Shell doit commencer par la ligne:

```
#!/bin/sh
```

Il doit aussi avoir être exécutable (bit x). Le #!/bin/sh sur la première ligne indique que ce script doit être exécuté par le Shell sh dont on indique le chemin d'accès. Pour rendre un fichier exécutable, tapez :

```
Chaland ~ chmod u+x fichi
```

On a utilisé le code suivant pour lire et écrire un fichier

```
import java.io.FileReader ;
import java.io.FileWriter ;
import java.io.BufferedReader ;
import java.io.IOException ;
import java.io.PrintWriter;

public class Fichier
{
public static void ecrire(String path, String text)
{
PrintWriter ecri ;
try
{
ecri = new PrintWriter(new FileWriter(path));
ecri.print(text);
ecri.flush();
ecri.close();
} //try
catch (NullPointerException a)
{
System.out.println("Erreur : pointeur null");
}
catch (IOException a)
```

```
{  
  
System.out.println("Problème d'IO");  
}  
}//ecrire  
  
public String lire (String path)  
{  
    BufferedReader lect ;  
    String tmp = "";  
    try  
    {  
        lect = new BufferedReader(new FileReader(path)) ;  
        while (lect.ready() == true)  
        {  
            tmp += lect.readLine() ;  
        }  
    }  
    }  
    catch (IOException a)  
    {  
        System.out.println("Erreur : pointeur null");  
    }  
    catch (NullPointerException a)  
    {  
        System.out.println("Problème d'IO");  
    }  
    return tmp;  
}  
}//lecture  
}  
}
```

### Langage de programmation:

Pourquoi Java? !

### Historique de java :

Les débuts de Java remontent à avril 1991, quand un petit groupe d'employés de Sun ont été déplacés dans un autre département pour travailler à un projet baptisé 'Green' voué à la réalisation d'application électronique commerciale.

L'équipe Green était libre de faire ce qu'elle voulait. Sa première mission fut de créer un appareil capable de contrôler plusieurs équipements électroniques é la fois. Un langage fut créé à cette intention, le langage Oak, dont les premiers pas remontent à aout 1991. Le nom Oak, trop proche d'une autre marque, fut rebaptisé Java en 1995.

L'annonce officielle de Java et HotJava date, quant à elle, du salon SunWorld 1995. Le projet Green était originellement orienté vers la création de systèmes d'exploitation pour Personal Digital Assistant, set-top boxes etc.

Le projet initial fut dissout et la technologie recentre vers le CD-ROM, le multimédia en ligne et les environnements réseau.

Le maître d'œuvre de Java est James Gosling, qui commence par vouloir étendre le C++ avant de réaliser le premier compilateur Java écrit en langage C. Arthur Van Hoff implanta le compilateur Java en Java lui-même à la fin 1994.

### Caractéristiques du langage :

#### □ □ **Distribué :**

Java possède une importante bibliothèque de routines permettant de gérer les protocoles TCP/IP tels que HTTP et FTP. Les applications Java peuvent charger et accéder à des sur Internet via des URL avec la même facilité qu'elles accèdent à un fichier local sur le système.

#### □ **Fiabilité :**

Java a été conçue pour que les programmes qui l'utilisent soient fiables sous différents aspects. Sa

Conception encourage le programmeur à traquer préventivement les éventuels problèmes, à lancer des

vérifications dynamiques en cours d'exécution et à éliminer les situations génératrices d'erreurs.

#### □ □ **Orienté objet :**

Pour rester simples, disons que la conception orientée objet est une technique de programmation qui se concentre sur les données (les objets) et sur les interfaces avec ces objets.

#### □ □ **Simple :**

Nous avoires voulu créer un système qui puisse être programmé simplement sans nécessiter un apprentissage ésotérique, et qui tire parti de l'expérience standard actuelle. En conséquence, même si nous pensions que C++ ne convenait pas, Java a été conçu de façon relativement proche de ce langage dans le dessein de faciliter la compréhension du système. De nombreuses fonctions compliquées, mal comprises, rarement utilisées de C++.

□ □ **Sécurité :**Java a été conçu pour être exploité dans des environnements serveur et distribués. Dans ce but, la sécurité n'a pas été négligée. Java permet le construction de systèmes inaltérables et sans virus.

#### □ □ **Architecture neutre :**

Le compilateur génère un format de fichier objet dont l'architecture est neutre – le code compilé est exécutable sur de nombreux processeurs, à partir du moment où le système d'exécution de Java est présent.

Pour ce faire, le compilateur Java génère des instructions en bytecode qui n'ont de lien avec aucune architecture particulière. Au contraire, ces instructions ont

été conçues pour être à la fois faciles à interpréter et faciles à traduire en code natif.

□ □ **Portable :**

*A la différence du C/C++, on ne trouve pas les aspects de dépendance de la mise ne œuvre dans la spécification. Les tailles des types de données primaires sont spécifiées, ainsi que le comportement arithmétique qui leur est applicable.*

□ □ **Interprété :**

*L'interpréteur Java peut exécuter les bytecode directement sur n'importe quelle machine sur laquelle il a été porté. Dans la mesure où la liaison est un processus plus incrémentiel et léger, le processus de développement peut se révéler plus rapide et exploratoire.*

□ □ **Performances élevées**

*En général, les performances des bytecodes interprétés sont tout à fait suffisantes, il existe toutefois des situations dans lesquelles des performances plus élevées sont nécessaires. Les bytecodes peuvent être traduits à la volée en code machine pour l'unité centrale destinée à accueillir l'application.*

□ □ **Multithread :**

*Les avantages du multithread sont une meilleure interactivité et un meilleur comportement en temps réel.*

## Référence:

[1]WWW .frameip.com

[2] site de Developpez

[3] site de sun.fr

[4] Livre programmer en java :CALUDE DELANNOY

[5] Wikipédia l'encyclopédie

[6] Commentsamarche.FR

[7]code source.com