**Rapport de cour de java: Les interfaces**

Une interface est un type de référence comme la classe c’est-à-dire qu’on peut créer des variables de type interface :

**Exemple**: I var **; avec I une interface**

En d’autre terme une terme une interface n’est pas directement instanciable.

Elle n’a pas de constructeurs et ne comporte que des méthodes abstraites et des attributs public static final.

**Exemple**: new I(<args) n’est pas autorise mais cependant on a le droit de mettre

I[] t=new I[3].

Une interface ne peut qu’être instancier indirectement par le biais de ses sous classes(ou classes filles ) concrètes.

Soit une classe A implémentant directement et uniquement une interface nomme I.

La syntaxe de définition de la classe A se définie comme suit :

Modificateur(s) classe A [**extends** B] **implements I**{  
 //corps de la classe

}

Remarque :

* Extends B pas obligatoire a moins que A hérite B
* A est une sous classe directe de l’interface I
* Une interface peut hériter de plusieurs interfaces

La syntaxe de définition d’une interface X héritant directement des interfaces I1,I2,I3…In se définie comme suit :

Modificateur(s) interface X **extends I1,I2,I3….In**{  
 //corps de l’interface

}

Une classe indirecte de l’interface I est une classe qui :

implémente une interface X héritant directement(ou indirectement) de l’interface I.

Représentation graphique d’une interface

Une interface X hérite **indirectement** d’une interface I si :

L’interface X hérite directement ou indirectement d’une interface Y héritant directement ou indirectement de l’interface I

|  |
| --- |
|  |
|  |

* Une sous classe indirecte d’une interface I est une classe qui soit hérite X directement ou indirectement d’une classe B qui elle implémente directement ou indirectement de l’interface I.

NB : B peut être abstraite ou concrète

Si une classe A hérite d’une classe B implémentant les interfaces I1,I2,I3….In sa syntaxe definition est la suivante :

Modificateur(s) classe A **extends** B **implements I1,I2,I3…In**{  
 //corps de la classe

}

* Différentes façon différences

Soit A et B deux classes :

A herite B : A et B concretes

A herite B avec A abstrait et B concret

A herite B avec A et B abstraite

A herite B avec B abstrait et A concret

NB : une classe n’herite jamais de plusieurs directement