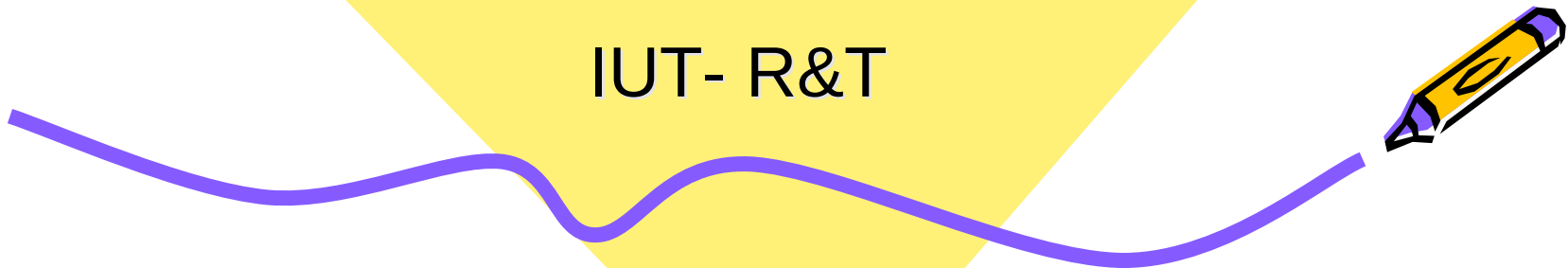




Javadoc

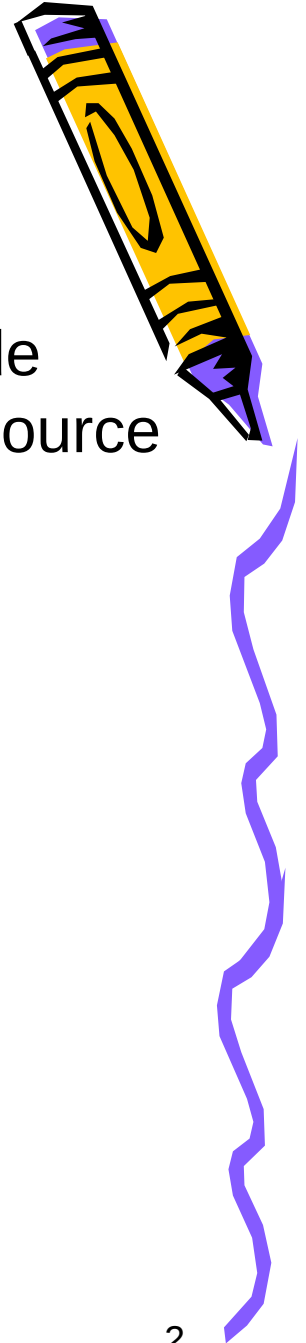
Karima Boudaoud
IUT- R&T



Généralités (1)

○ Objectif

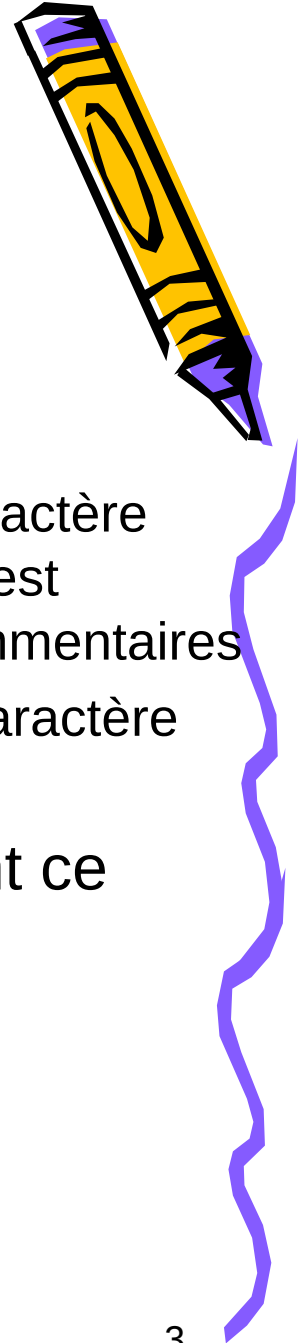
- javadoc produit de la documentation en partant de commentaires particuliers insérés dans le code source des classes (`/**...*/`)
- On peut ainsi documenter :
 - paquetages
 - classes ou interfaces
 - variables d'instances
 - méthodes



Généralités (2)

○ Format des commentaires

- Les commentaires peuvent contenir
 - du texte simple
 - des tags HTML de mise en forme de texte (italique, caractère gras, caractère à espacement fixe,..); un tag bien utile est `<code>` (et `</code>`) pour inclure du code dans les commentaires
 - des tags spéciaux à javadoc, qui commencent par le caractère `@`: `@author`, `@version`, `@param`,...
- Les commentaires doivent être placés juste avant ce qu'ils commentent



Utilisation de javadoc



○ Commande

- `javadoc -author -version -d <chemin> <fichiers java>`
- `<chemin>` : chemin où les fichiers html seront générés
- `< fichiers java >` : les fichiers java dont on veut générer de la documentation
- Exemple : `javadoc -author -version -d ~/Langage-Java/TP5/ *.java`



Commentaires de classe et d'interface

○@author nom

- Indique l'auteur (plusieurs fois le tag si plusieurs auteurs)

○@version texte

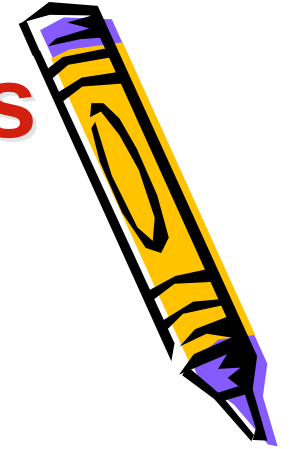
- Précise la version

○Remarque

- Ces 2 tags ne sont utilisés dans la documentation que si on donne les options `-author` et `-version` de la commande `javadoc`



Commentaires de méthodes



○ **@param** <paramètre> description

- Documente le paramètre de la méthode

○ **@return** description

- Documente ce que retourne la méthode

○ **@throws** <classe exception> description

- Documente une exception (on peut aussi utiliser **@exception**)

○ **Remarques**

- Tous les tags de même type doivent se suivre
- Les descriptions peuvent s'étaler sur plusieurs lignes.



Exemple de documentation de méthode

/**

*Returns the character at the specified index. An index ranges from `0` to `length() - 1`.

*

* **@param** index the index of the desired character.

* **@return** the desired character.

* **@exception** StringIndexOutOfBoundsException

* If the index is not in the range

* `0` to `length() - 1`.

*/

```
public char charAt(int index { ...  
}
```

