

# Modélisation TLM en SystemC

## Consignes communes à tous les TPs

Pour tous les TP, un certain nombre de choses sont imposées. Une bonne idée est de se préparer une checklist des choses à ne pas oublier. Une autre est d'automatiser ces vérifications avec un script.

### 1 Style de codage imposé

Une part de la note sera donnée sur la présentation du code. On vous impose le style suivant (qui est en principe suivi sur le code fourni, si ce n'est pas le cas signalez-le comme un bug) :

- Indentation correcte, 3 ou 4 espaces, ou tabulation avec une taille de 3 ou 4 espaces. L'outil `indent` avec l'option `i-kr` va bien, mais marche moyen sur du C++, `clang-format` bien paramétré est une bonne solution aussi;
- Accolades ouvrantes sur la même ligne que la ligne de code qui lui correspond (par exemple, `if (toto) {` sans retour à la ligne entre `)` et `{`, sauf pour les déclarations de fonctions et de classes (toute règle a ses exceptions, sinon ce n'est pas drôle!);
- Placement des espaces autour des opérateurs, parenthèses, ... en suivant le style de codage de Linux; (<https://www.kernel.org/doc/Documentation/CodingStyle>, Chapitre 3.1, mais les autres chapitres sont intéressants à lire);
- Pas de ligne affreusement longue (i.e. de plus de 132 colonnes);
- Pas d'espace en fin de ligne (`:se list` vous les montre sous `vim`).

Voir aussi [http://ensiwiki.ensimag.fr/index.php/Ecrire\\_du\\_code\\_de\\_qualite](http://ensiwiki.ensimag.fr/index.php/Ecrire_du_code_de_qualite) pour plus d'information.

### 2 Fichiers à rendre

Une part de la note sera également accordée à la propreté du rendu : rendez tout ce qui est nécessaire à votre TP et seulement cela. Ne rendez pas de fichiers inutiles (par exemple, ne rendez pas les fichiers binaires ni les fichiers de sauvegardes de votre éditeur de texte).

### 3 Commentaires

Les commentaires doivent être pertinents : en particulier, supprimez les commentaires du squelette destinés à vous aider, ne laissez pas de traces de debug ...

En bref, tous les commentaires contenant les chaînes `TODO` ou `FIXME` doivent être supprimées. Faites également le ménage dans votre code (supprimez les commentaires « pense-bêtes » et les essais ratés par exemple).

## 4 Format du rendu

Le rendu doit impérativement être un pointeur vers un dépôt `git`.

Il n'est pas demandé de rapport détaillé, mais vous pouvez écrire quelques mots à l'intention du correcteur dans un fichier `README.md` ou `README.txt` ou `README.rst` (respectez le nom `README` scrupuleusement) si vous le souhaitez (si votre rendu a des spécificités, limitations connues ou extensions par exemple).

## 5 Fraude

Toute copie (sur une autre équipe, un projet d'une année passée—y compris votre ancien TP pour les redoublants—ou autre) sera sanctionnée par un 0/20 ou un passage en conseil de discipline. Lire la charte contre la fraude pour les détails : [https://intranet.ensimag.fr/teide/Charte\\_contre\\_la\\_fraude.php](https://intranet.ensimag.fr/teide/Charte_contre_la_fraude.php)